Oracle® Utilities Customer Cloud Service

User Guides



Release 26.10
G58676-01
October 2026

Oracle Utilities Customer Cloud Service User Guides, Release 26.10

G58676-01

Copyright © 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

Part I Oracle Utilities Customer Cloud Service User Guides

Part II Customer To Meter Business User Guide

Part III Application Framework Administrative User Guide

Part IV Customer to Meter Administrative User Guide

Part V Oracle Utilities Cloud Service Foundation

Copyright © 2026, Oracle and/or its affiliates.

Part I

Oracle Utilities Customer Cloud Service User Guides

Oracle Utilities Customer Cloud Service shares significant functionality with Oracle Utilities Customer to Meter. As such, both use the same user documentation. Oracle Utilities Customer Cloud Service User Guides comprise the following:

- Oracle Utilities Customer to Meter Business User Guide
- Oracle Utilities Application Framework Administrative User Guide
- Oracle Utilities Customer to Meter Administrative User Guide
- Oracle Utilities Cloud Service Foundation Administrative User Guide

Copyright © 2026, Oracle and/or its affiliates.

Part II

Customer To Meter Business User Guide

Welcome to the Oracle Utilities Customer To Meter Business User Guide. This document is used with the following products:

- Oracle Utilities Customer to Meter
- Oracle Utilities Customer Cloud Service
- Oracle Utilities Billing Cloud Service

> **Note**
>
> Refer to Oracle Utilities Billing Cloud Service for specific information related to Oracle Utilities Billing Cloud Service.

- Oracle Utilities Rate Cloud Service

> **Note**
>
> Refer to Oracle Utilities Rate Cloud Service for specific information related to Oracle Utilities Rate Cloud Service.

This guide describes business process-related features and functionality, including:

- Customer to Meter Overview
- User Interface Features
- Customer Information
- Asset Management
- Device Management
- Device Installations
- Measurements
- VEE
- Usage
- 360 Degree and Unified Search
- 360 Degree Views
- Service Orders
- Billing
- Payments
- Adjustments
- Credit & Collection

Copyright © 2026, Oracle and/or its affiliates.

- Financial Transactions
- Deposits
- Statements
- Sales & Marketing
- Rates
- Quotes
- Service Credits
- Loans
- Non-Billed Budgets
- Case Management
- Premise Management
- Umbrella Agreement Management
- Overdue Financial Obligations
- Rebate Claims
- Communications
- Integrations
- Aggregations
- Master Data Synchronization
- Outage Storm Mode
- Dashboards
- Totals and Trends
- Time Zone Support
- Service Issue Monitors
- Market Transaction Management Addendum
- Market Transaction Management - US Distribution
- Customer Program Management
- Oracle Utilities Rate Cloud Service
- Oracle Utilities Billing Cloud Service
- Settlement
- Data Access Restrictions
- To Do Processing
- Reports

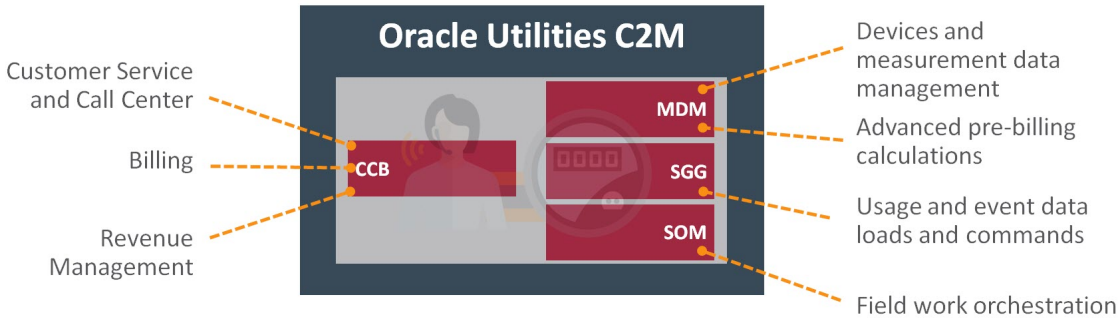
Customer to Meter Overview

Oracle Utilities Customer to Meter is a next generation customer service and billing application that incorporates a modern meter data management system. This system is designed to scale with mass smart meter volumes while handling the complexity associated with a utility's

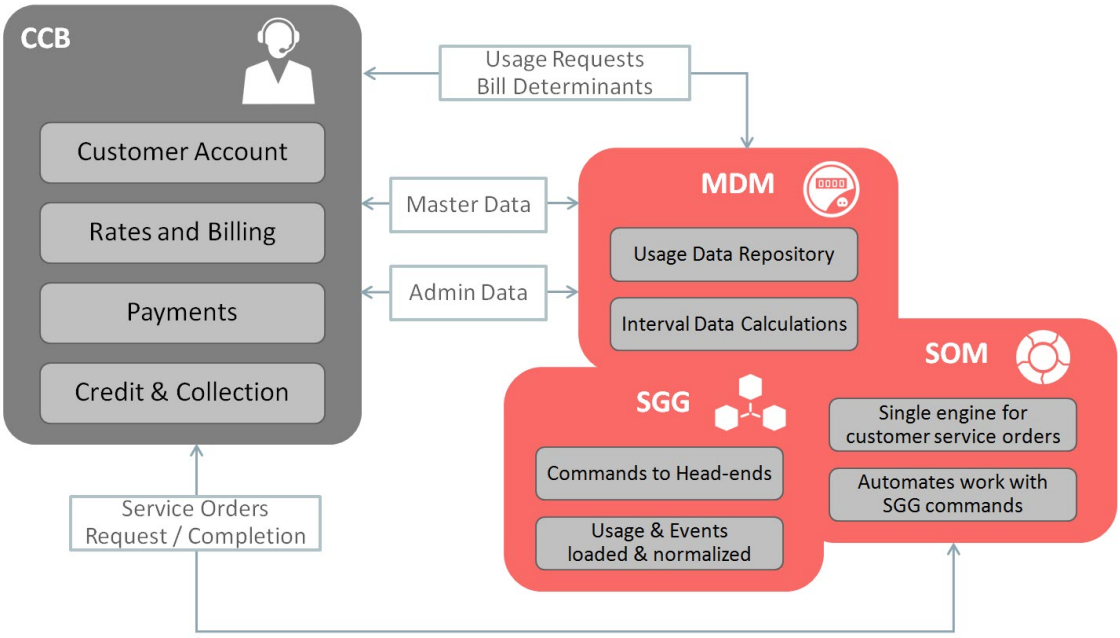
Copyright © 2026, Oracle and/or its affiliates.

processes across multiple types of service, operating divisions, and jurisdictions. The new application components allow for traditional scalar meter functionality that is equivalent to prior versions of Oracle Utilities Customer Care and Billing; so, it can accommodate the many types of utility environments, and evolve with the business to include more advanced digital requirements.

The roles for each application area in Oracle Utilities Customer to Meter are defined as follows:



The application components have internal integration for master, admin, and transactional data. The primary transactions between them are for bill determinant processing with MDM and service order processing with SOM. These transactional processes can be initiated by the CCB solution, or from other external systems.



In addition, Oracle Utilities Customer To Meter can also include asset management functionality via Oracle Utilities Operational Device Management which provides the ability to handle large volumes of assets and to manage the receipt, installation, maintenance, tracking and removal of those assets.

- Oracle Utilities Customer Care and Billing Overview
- Oracle Utilities Meter Data Management Overview
- Oracle Utilities Smart Grid Gateway Overview
- Oracle Utilities Service Order Management Overview
- Oracle Utilities Operational Device Management Overview

Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Market Settlements Management Overview

Oracle Utilities Customer Care and Billing Overview

Oracle Utilities Customer Care and Billing (CCB) manages customer information and processes associated with the customer lifecycle including:

- Sales and marketing, and program management.
- Customer information for residential, commercial, and industrial customers; including, complex premise management, umbrella agreements, and quotations proposals.
- Starting and stopping service; including, order entry for eligible products and services.
- Customer care interactions for inbound communications (such as phone calls), outbound communications (such as letters or notifications), and case management (configurable customer processes).
- Advanced rating rules.
- Financial management for billing, budget plans, deposits, loans, and payment processing (including cashiering portals).
- Credit and collection for overdue receivables management.

Oracle Utilities Customer Care and Billing acts as a detailed sub ledger for accounts receivables with the ability to send results to a general ledger. It also provides a Control Central portal that increases productivity by providing a comprehensive view of the customer and their related account information so that most questions can be quickly answered without having to navigate through the system.

Oracle Utilities Meter Data Management Overview

Oracle Utilities Meter Data Management (MDM) acts as a consolidated usage measurement data repository with advanced validation, editing, and estimation (VEE) rules. These rules ensure accuracy and measurement compliance that is associated with advanced metering infrastructure (AMI), traditional one-way meters, and a rich usage calculation rules engine.

The following lists the most crucial business processes that are enabled within this system:

- Defining devices; such as, meters at customer locations, device configurations for measurement collection and command execution, customer service points, and effective device installations.
- Loading scalar readings, interval data, and events data from a Head End system or other source.
- Automatic validating, editing, and estimating (VEE) measurement data
- Robust editing capabilities for scalar readings and interval measurement data
- Calculating and publishing bill determinants and other transactions from measurement data. This data is then used in external systems; such as, billing, and pricing. It can also be applied to aggregation for settlement, load research, forecasting, and revenue management.
- Service Issue Monitor configurable process to identify and correct devices that may have operating issues.

Copyright © 2026, Oracle and/or its affiliates.

The system has a 360 Degree View device portal that provides advanced navigation and operational analytics to improve operational efficiency. It is designed for handling large volumes of meter/device data to enable increased accuracy, flexibility, and scalability.

Oracle Utilities Smart Grid Gateway Overview

Oracle Utilities Smart Grid Gateway (SGG) is designed as a messaging broker to communicate with meters and other devices using a common interface. This allows the system to hide much of the background complexity that can occur when multiple head end or meter data collection systems are used.

SGG functionality covers the following types of standardized processing.

- Loading usage and event data records that are collected on a frequent basis.
- Providing normalized raw data to other applications that require information with the ability to filter based on requirements.
- Standardizing command and control use cases; such as, turning a meter device on or off, checking the current status of a device, and requesting a measurement on-demand via electronic messages to the corresponding head end system.
- Centralizing command and data exceptions with the ability to automate retry processing.

SGG delivers pre-built adapters that support integrations to leading metering technology vendors. It also provides an Adapter Development Kit (ADK) to easily integrate with other sources. The system can provide a single auditable link to devices across all systems, simplify integrating with other utilities applications, and allowing normalized protocols and data types. The platform can expand to allow other types of device and customer communications, beyond meters; or other types of distribution and network devices that are upstream from the metering technology.

Oracle Utilities Service Order Management Overview

Oracle Utilities Service Order Management (SOM) delivers a new approach to handling the most common customer service order use cases for smart metering technology and traditional metering. Rather than generating specific field activity tasks, Oracle Utilities CCB triggers field activity requests that are based on the classic customer scenarios: start service, stop service, disconnect for non-payment, reconnect service, etc.. Oracle Utilities SOM receives these requests and determines whether the installed device technology can support electronic commands (sending messages via Oracle Utilities SGG), or not (meaning that a field visit is required to complete necessary tasks). Some situations may also require both automated and manual activities to occur, and in a specific sequence; such as, first replacing an old meter with a smart meter through a field visit, then sending a series of Oracle Utilities SGG messages to the new meter.

Oracle Utilities SOM continually monitors and initiates activities to get the service point, device, and completion fully processed; this includes obtaining supporting data, such as measurements needed for billing, in a manner that is appropriate for the associated meter’s technology. As more smart devices are installed in the field, Oracle Utilities SOM provides increased savings by reducing field visits, and using more electronic messaging that take care of logic and error handling.

In addition to being initiated by the customer system, service order requests can also be initiated by other systems; for example,

Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Meter Data Management might detect service issue through its monitoring processes
- Oracle DataRaker's advanced analytics might detect patterns with the meter, device, or customer behavior.
- Oracle Utilities Operational Device Management (ODM) might trigger operational processes; such as, periodic device testing or configuration parameter updates.

It is critical for customer service business processes to see all types of requests that are initiated by other systems and their current execution status. SOM assumes that a mobile workforce system is in use. These systems provide several important functions: they remove the need for printed service orders, handle the assignment and scheduling of work to crews, and provide appointment booking services.

Oracle Utilities Operational Device Management Overview

Oracle Utilities Operational Device Management provides functionality to handle large volumes of assets and to manage the receipt, installation, maintenance, tracking and removal of those assets.

The system supports the following business processes:

- Capturing device attributes and supporting any type of device
- Supporting integration with other utility applications requiring device information and configuration
- Supporting detailed smart device attributes
- Supporting detailed location management and tracking of individual devices throughout their lifecycle
- Capturing device configurations and settings
- Managing configuration between devices, components and software
- Tracking firmware and firmware versions on smart devices

Oracle Utilities Market Settlements Management Overview

Oracle Utilities Market Settlements Management (MSM) provides the core functionality for calculating and settling energy changes, also referred known as load settlement,.

Market settlement is the process of reconciling the difference between a supplier's energy purchases and its customer's usage. Suppliers purchase energy via bilateral contracts and day-ahead market transactions. Any differences between these purchases and supplier customer usage must be determined by the market and charged or credited at real time prices. The supplier's customer usage is not directly determined by the market, but by numerous other market participants that track customer supplier enrollments, aggregate customer usage to the supplier, and submit supplier's usage schedule to the market.

Load settlement starts with the bottom up aggregation of actual load or estimated load for individual customer accounts or service points. Distribution and transmission losses are added in to calculate an unreconciled load amount. The market determines the total generation supplied to a zone and the difference with the unreconciled load is referred to as Unaccounted for Energy (UFE). The UFE is then allocated to suppliers on a load ratio share basis.

Copyright © 2026, Oracle and/or its affiliates.

The load settlement process is typically performed multiple times. The “preliminary” load settlement is done soon after the Operational Day and is often based on estimated load. The “final” load settlement is done around 60 days after the Operational Day, so that actual data has been received and customer bills issued.

The market participant responsible for load settlement can differ per market. For example, load settlement for a large portion of the Texas market is performed by an Independent System Operator (ISO) known as the Electric Reliability Council of Texas (ERCOT), whereas in the PJM market (encompassing Pennsylvania, New Jersey, and Maryland) load settlement is the responsibility of the Electric Distribution Company (EDC). While there are significant differences between markets in how load settlement is preformed, the high-level process steps are basically the same.

User Interface Features

This section describes basic system concepts, features, and standards of the user interface.

- User Interface Standard Features
- User Preferences
- Initiating Scripts
- Shortcut Key Summary

User Interface Standard Features

This section describes standard functionality realted to how the system looks and behaves.

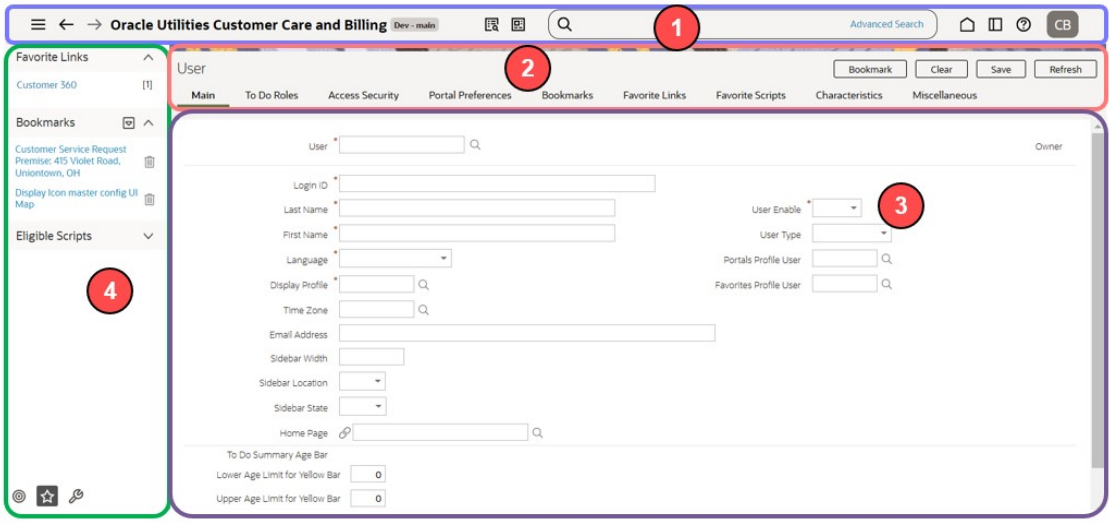
- Page Components
- Portal Pages
- Fixed Pages
- UI View Settings

Page Components

Oracle Utilities Application Framework screens are comprised of the following main areas:

1. The Application Toolbar
2. The Page Title Area
3. The Object Display Area
4. The Sidebar Area

Copyright © 2026, Oracle and/or its affiliates.



In addition there are two optional areas, not shown, that may be visible depending on conditions: the Assistant Area and the Script Area.

Keyboard Shortcuts	Description
`Alt+[`	Moves the cursor focus to the next main page component based on the standard browser keyboard navigation "order".
`Shift+Alt+[`	Moves the cursor focus to the previous main page component based on the standard browser keyboard navigation "order".

> **Note**
>
> The look and feel of the application may be modified after the product is installed. See Custom Look and Feel Options for customization information, including how to change colors, fonts, and other system features. The information provided in this document represents features and functionality available only in the delivered product.

The next topics provide more information about each component.

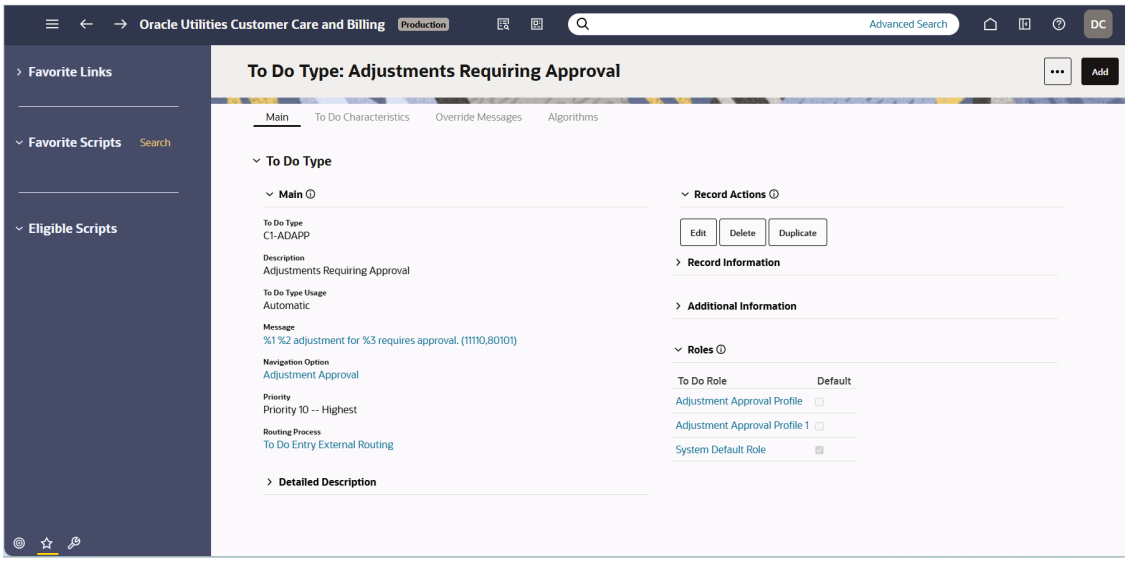
- Redwood User Experience
- The Application Toolbar
- Page Title Area
- Object Display Area
- Sidebar Area
- Assistant Area
- Script Area

Redwood User Experience

The Redwood user experience option may be enabled system wide via a feature configuration. Additionally, the user menu allows individual users to toggle between the Redwood user experience and the classic user experience.

When enabled, the following illustrates a typical portal page layout.

Copyright © 2026, Oracle and/or its affiliates.



The following points highlight differences in the behavior and appearance of pages in the application when enabling this experience:

- A colored theme is used for the application toolbar and the sidebar.
- Portal based pages follow the standard of the Label over the Field rather than before it.
- Some icons are different.
- The expand / collapse control for a zone or section is moved to the left (or right for a right-to-left language).
- All input dialogues for portal based pages are displayed as a Drawer.
- When there are more than two page action buttons, one button is visible directly and the rest are grouped in an 'overflow' button (with three dots). This change applies to both portal based pages and fixed pages
- The spacing of elements is adjusted.

The Application Toolbar

The application toolbar includes the product's name along with several features described in the subsequent sections.

> **Note**
>
> Your specific application may introduce additional icons or features to the toolbar.

> **Note**
>
> If a domain name is configured for your environment, it is listed after the product name.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The toolbar is responsive to window resizing or zooming in. When there is less room, the product name and the domain name are the initial elements to be reduced or removed altogether.

- Menu
- Back and Forward Arrows
- Search
- Home Icon
- Sidebar Toggle
- Help Menu
- User Menu

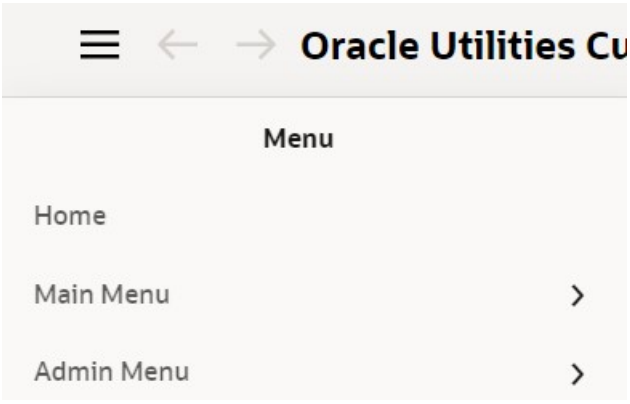
Menu

The Menu icon,  , is available in the application toolbar to help you navigate to the different pages of the system.

You only see the menu items that you have security access for. In turn if you do not have access to any menu items in a submenu, the whole submenu is suppressed.

Keyboard Shortcut
`Ctrl+Alt+M`

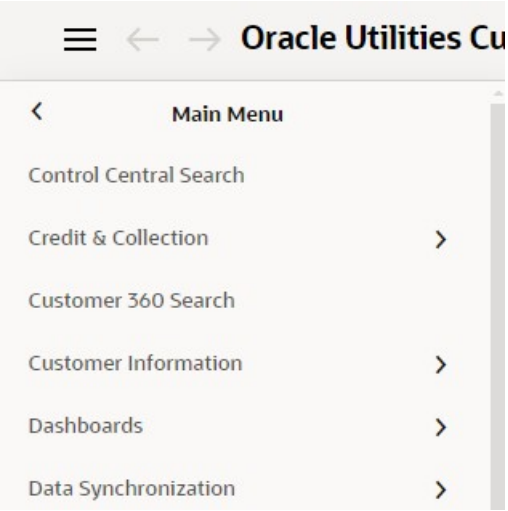
The initial view of the menu includes a link to the Home page, plus entries for the Main Menu and Admin Menu.



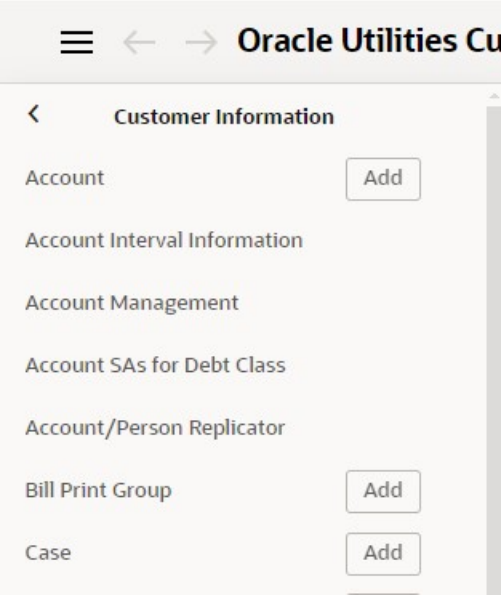
Main Menu

Clicking the Main Menu entry displays a submenu organized by functional area.

Copyright © 2026, Oracle and/or its affiliates.



Clicking any functional area in turn displays a submenu that contains pages within that area.



Some entries have an Add button adjacent. For these entries, clicking the menu item text opens the Search for that menu item. Click the Add button to add a new record for that search item. For entries without an Add option, select the menu item itself to launch that page.

How the Search and Add option behaves depends upon whether the maintenance page is fixed or portal-based.

Fixed:

- **Search:** Navigates to the page and displays a pop-up search window for the user to enter the search criteria and select the entity to display. Once the entity is selected, the maintenance page is displayed with the data populated in the fields.
- **Add:** Displays the maintenance page with empty fields so that the user can complete the information and save the entity.

Portal-Based:

Copyright © 2026, Oracle and/or its affiliates.

- **Search:** Displays a search portal where the user can enter the search criteria and select the desired entity. Once the entity is selected, the user is typically taken to a maintenance portal or the information is broadcasted to other zones within the same portal.
- **Add:** Either navigates to a page where the user can select the entity type or business object, or directly to the input map where the user can enter and save the entity.

Users may also opt to use Menu Item Search to find a page.

Admin Menu

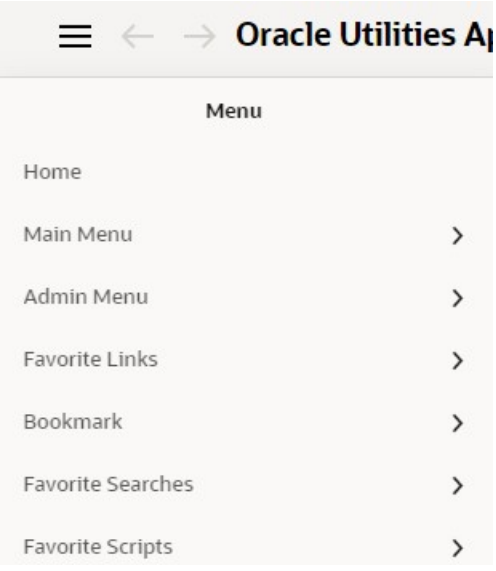
The admin menu functionality is analogous to the Main Menu functionality. One difference is that your implementation may opt to organize the admin menu alphabetically rather than by functional area.

> **Note**
>
> Refer to Installation Options for information about the admin menu configuration options.

> **Note**
>
> Menu navigation paths referenced in the administrative user guide provide the functional grouping. If your implementation uses the alphabetic grouping, the page can be found in the alphabetic letter that is the first letter of the page name.

Favorites, Bookmarks

If you have any Favorite Links, Favorite Scripts, Favorite Searches or Bookmarks, entries to open those options are also visible when initially opening the menu.



Clicking any of those menu entries displays you configured favorites or your saved bookmarks. The information is the same as what you see in the equivalent sidebar zone.

Copyright © 2026, Oracle and/or its affiliates.

Back and Forward Arrows

The back and forward arrows,  , in the application toolbar allow you to navigate between the most recently viewed pages.

Back

Click the **Back** arrow to open the previously visited page.

Keyboard Shortcut
`Alt+B`

Click and hold the **Back** arrow to open a list of pages previously visited that you can choose from.

Forward

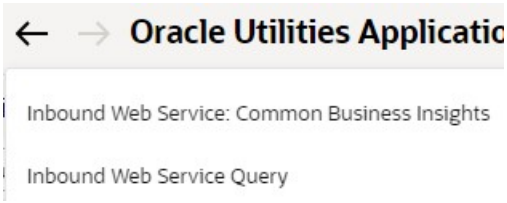
Click the **Forward** arrow to return to the page that was most recently dismissed.

Keyboard Shortcut
`Alt+G`

Click and hold the **Forward** arrow to open a list of pages that you had visited but went 'back' from that you can choose from.

The **back** and **forward** arrows are enabled in the toolbar only when page use warrants their appearance.

Note that if the page has been configured to display an information string in the page title area, that information will also be visible in the dropdown of previous pages listed.



> **Note**
>
> Under certain circumstances, such as revisiting a page that was left unsaved after being loaded with default data, a page may not appear as it did when you left it.

Search

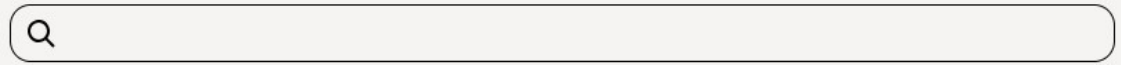
The toolbar Search field and search icon provide the ability to search for specific menu items or, depending on your product, certain business entities.

Copyright © 2026, Oracle and/or its affiliates.

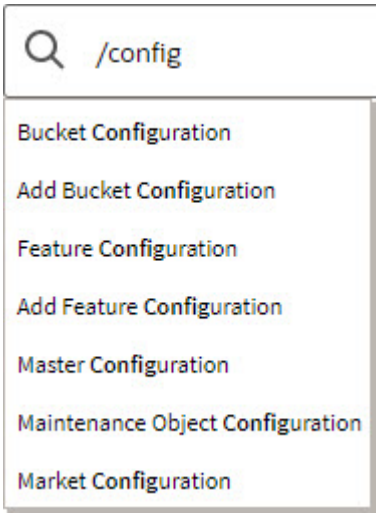
Menu Item Search

This search allows a user to enter the description of a menu item entry to navigate directly to the corresponding page or BPA script rather than using the menus to navigate to the desired page or script. If you start the search text with a slash "/", the search is confined to menu items. If you search for a menu item without a slash, the menu items are searched along with other unified search results.

Keyboard Shortcut
`Ctrl+Alt+F`



Typing text in the field preceded by a slash shows menu items whose descriptions include the typed text.



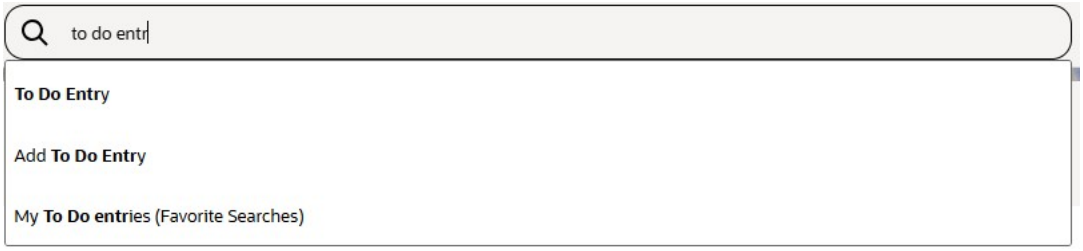
Note that for menu items within the Main Menu and Admin Menu, only menu entries that the user has security access for are included in the results. The text included in the search is taken from the menu item description. As described in Main Menu, menu lines often have options for search and Add. In most instances, the description of the "search" menu item is simply the menu entry text, for example **To Do Entry**. The description of the "Add" menu entry includes the word Add in front of the menu entry text, for example **Add To Do Entry**.

As mentioned in Menu Item Suppression, menu items will not appear if it is associated with a module that is suppressed.

> **Note**
>
> When operating in debug mode, the description of the parent menu is displayed in parentheses after the menu item description.

In addition, any of your Favorites, Bookmarks visible in the menu and in the respective sidebar zone are included in the search results. When the search result is related to one of these categories of configuration, the category name is displayed in parentheses in the results.

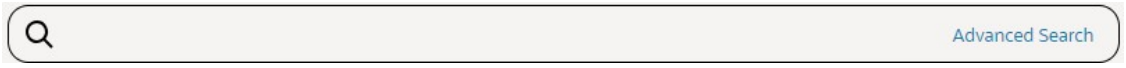
Copyright © 2026, Oracle and/or its affiliates.



> **Note**
>
> The toolbar is responsive to window resizing or zooming in. The search field may be removed when the toolbar area is reduced.

Unified Search

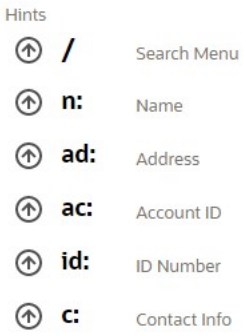
If your product has implemented unified search, you may use the toolbar search box to search for business entities. The unified search feature is a simplified version of your product's main search that allows you to lookup up records using free form text without leaving your current page. For more complex queries, use the link provided as part of the search box to navigate to your product's advanced search portal.



The type of business entities you may query and the filters you may use depend on the specific unified search implementation of your product. For example, if your product supports a search for customers based on their name or address, you may type in part of a name or an address to find matching results. If a unified search option is not configured for your product or you do not have security access to use it then the search box may only be used for searching for menu items.

To search for business entities, you may enter your search criteria in the following ways:

- Type in your search text in **free form**. For example, if your product's unified search option supports a search for customers by name or address, the text you enter may be considered as part of a name or an address.
- Depending on your product's unified search configuration, you may use specific **hints** to explicitly identify your filtering criteria. For example, an address filter may be associated with the hint name "ad:" allowing you to be more explicit and enter "ad: main" to find records by their address. Click the search icon in the search input area to view the hints that are configured for this search. You can also click the up arrow adjacent to the hint to automatically populate the hint text into the search.

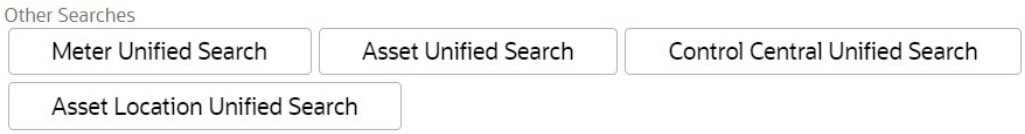


Copyright © 2026, Oracle and/or its affiliates.

- You may use free form text or hints but not both.

As you type in your text, the application attempts to find matching records and presents the top most results for your review. You may select a record from the results list to navigate to the corresponding portal.

If your product has configured more than one unified search that you are allowed to use, one of the searches is the default search. Click the search icon in the search input area to choose a different search option. After selecting a different option, the hints for the selected search are shown and the Advanced Search link will navigate to the query portal configured for the selected search.



Refer to your specific product documentation for more information about your product's unified search features.

Refer to Understanding Unified Search to better understand unified search configuration.

Home Icon

Click the **Home** icon,  , on the Application Toolbar to open your home page. Your home page is defined on the main tab in User Preferences.

Keyboard Shortcut	Alternate Shortcut
`Alt+O`	`Alt+Shift+O`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

> **Note**
>
> The toolbar is responsive to window resizing or zooming in. The home icon may be removed when the toolbar area is reduced.

Sidebar Toggle

Click the **Sidebar Toggle** icon on the application toolbar to open or close the Sidebar Area. The image displayed depends on your sidebar's configured location and whether it's visible or collapsed.

Copyright © 2026, Oracle and/or its affiliates.

Table Sidebar icon list

Location / State	Icon
Left / Visible	
Left / Collapsed	
Right / Visible	
Right / Collapsed	

Keyboard Shortcut
`Alt+J`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

Closing the sidebar suppresses the refresh of the zones within the sidebar based on updates to the context. Opening the sidebar bar restores the display of the sidebar portal. Maximizing the sidebar then refreshes the zones with the latest appropriate content.

Your user preferences record is updated with the current state of the sidebar.

Help Menu

The Help menu is available in the application toolbar to provide access to the following product support options. Open the menu by clicking the Help icon  .

Help

Select the **Help** entry in the menu to launch the Oracle Help Center in a new tab. The application will automatically navigate to the context-sensitive help information within the help center. Alternatively, use the following shortcut key to launch help directly.

Keyboard Shortcut
`Alt+F1`

Help Assistant

This entry is only relevant for cloud installations. The Implementation Assistant centralizes Oracle documentation, including configuration guides, setup instructions, data mappings, user guides, best practices, and troubleshooting material—into a natural-language chat experience. You may open this feature directly from the Oracle Help Center in the Utilities > Integrations page. In addition, if you have security access to the application service `F1-HELPASSISTANT`, you will see the Help Assistant option in the menu. Launching this opens the implementation assistant tool, targeting your specific product and version.

Copyright © 2026, Oracle and/or its affiliates.

Prepare Issue Details

Select the **Prepare Issue Details** entry in the menu to launch a script to capture details of an error that needs to be reported. Refer to Prepare Issue Details for more information on this option.

Release Notes

This entry is only visible for on premises installations. Select the **Release Notes** entry in the menu to launch in a new tab the Release Notes for your product and version.

What's New

This entry is only relevant for cloud installations. Select the **What's New** entry in the menu to navigate to the What's New Portal.

Enable | Disable Debug

If you have security access to the application service F10UDBUG, you will see an option in the menu related to debug mode depending on its current state. If debug mode is currently not enabled, the entry says Enable Debug, otherwise it says Disable Debug.

About

Select the **About** entry in the menu to display a window that describes the current release information for your version of your product.

The about window shows a variety of information including:

- The logged in user ID
- The list of products installed in the application along with the release ID. This information comes from the installation record. Note that in a Cloud installation, only the Cloud product name and release ID are displayed. The details of the products that make up the Cloud installation are not displayed (but are still visible on the installation record).

User Menu

The User menu is available in the application toolbar to provide access to user-specific options. Open the menu by clicking the User icon 

.

> **Note**
>
> **Configurable Menu.** Additional menu entries may be visible in your implementation as this menu configuration allows for additional options to be defined.

Preferences

Select the **Preferences** entry in the menu to view or modify your settings in a form that appears in the **Main** tab in the object display area. Alternatively, use the following shortcut key to access preferences directly.

Copyright © 2026, Oracle and/or its affiliates.

Keyboard Shortcut
`Ctrl+Alt+P`

Refer to User Preferences for more information.

Switch UI View

Select the **Switch UI View** entry in the menu to change the user experience from the Redwood User Experience to the classic user experience or from the classic to the Redwood user experience. This option is only available for users that have access to the **F1UIVIEW** application service.

Logout

Select the **Logout** entry in the menu to complete your current session of the application. Clicking this item logs the current user out and displays the Welcome page so that another user can login. Alternatively, use the following shortcut key to logout.

Keyboard Shortcut
`Ctrl+Alt+L`

Page Title Area

The Page Title Area displays the page title and the page actions toolbar.

> **Note**
>
> A portal-based page may be configured to suppress the page title altogether, including the page actions toolbar.

For certain portal pages in the system, the page title may have additional information.

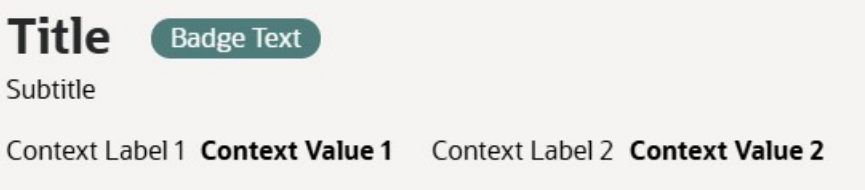
- If a portal-based page displays a single record, its information string may be displayed after the page title. This information is only visible if the portal has been configured to display the primary key's information string. Typically only "stand alone" maintenance portals are configured to show the information string, whereas "all-in-one" portals are not configured this way.

  > **NOT_SUPPORTED**
  >
  > Refer to Common Base Portals and Zones for more information about standard base product portals.

  > **Note**
  >
  > If an information string is long and would interfere with a page actions button, the information string will automatically be truncated and suffixed by an ellipsis, "...", indicating that more information is available. In this situation, the missing text can be viewed if the user hovers their mouse over the string.

Copyright © 2026, Oracle and/or its affiliates.

- A portal based page may be configured to show enhanced information, including a subtitle, information in 'badge' format and context name / value pairs, which may also allow you to click and drill into more information.



> **Note**
>
> Refer to Advanced Configuration for Portal Title / Tabs for details on how to configure this capability.

The Page Actions Toolbar displays standard control options, such as Save and Duplicate. More detail about the actions that may appear are described in the following section.

When applicable, banners may be presented to highlight important messages about the page content.

- Page Actions Toolbar
- Portal Banners

Page Actions Toolbar

In the Page Title Area, control functions appear in the Page Actions Toolbar; such as, Clear, Save, Duplicate, Delete and Refresh buttons.

> **Note**
>
> Buttons in the Page Title Area appear as the function is available to the user; for example, the Delete button will not appear if the user has made any changes to the page's content, or does not have the necessary security rights for this action.
>
> In addition, other functions that are specific to a portal-based page may appear in the toolbar, if configured. For example, the ability to navigate to the Search zone for a specific maintenance object may be visible on the maintenance portal. The portal specific actions may be displayed as individual actions, or grouped as an action menu.
>
> A portal-based page may be configured to suppress the page title altogether, including the page actions toolbar.

This section describes the standard features that are available in the Page Actions Toolbar.

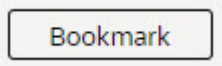
- Bookmark Button
- Next/Previous Item Buttons
- Duplicate Button
- Delete Button
- Clear Button
- Save Button

Copyright © 2026, Oracle and/or its affiliates.

- Refresh Button
- Portal Maintenance

Bookmark Button

Click the Bookmark button on the Page Actions Toolbar to save the current page and context to the bookmarks list.



Keyboard Shortcut
`Ctrl+Alt+B`

> **Note**
>
> **Secured.** This button is secured using the `F1-BOOKMARKS` application service. Users that do not have security for the application service will not see the Bookmark button (or the corresponding sidebar zone).

After clicking the Bookmark button, you are prompted to provide a name for the bookmark. After saving, it is visible in the Bookmarks Zone, as well as through the Menu, allowing you to return to this page with this context loaded with one click.

> **Note**
>
> Unsaved data will not be included in the bookmark. If you view a record and make changes to data without saving the changes and then bookmark the page, the unsaved changes are not included in the bookmark information.

You may change the name of the bookmark or change the sequence of the bookmarks by navigating to the User - Bookmarks in the user preferences.

> **Note**
>
> Depending on other actions available, this action may be grouped with other actions in a '**General Actions**' menu.

Next/Previous Item Buttons



The **Previous Item** and **Next Item** buttons appear in the Page Actions Toolbar under these conditions:

- You access a fixed page, meaning that it is not portal-based.
- You use a search page to display an item.
- There are multiple items in the search results list *below* or *above* the selected item.

Copyright © 2026, Oracle and/or its affiliates.

If, for example, you enter search criteria of "Smith" on the User Search page, you might see the following results:

- Smith, Albert
- Smith, Peter
- Smith, Zane

If you select **Smith, Peter**, Peter's user information is displayed on the User page. If you then click the **Next Item** button, **Smith, Zane** will be displayed. In addition, because **Smith, Zane** is the last entry in the search results, the **Next Item** button is no longer needed and will not appear in the Page Actions Toolbar.

If you then click the **Previous Item** button, **Smith, Peter** will be displayed; click again and the **Smith, Albert** record will appear, and, because **Smith, Albert** is the first entry in the search results, the **Previous Item** button is no longer needed and will not appear in the Page Actions Toolbar.

Keyboard Shortcut	Alternate Shortcut
`Alt+P`	
`Alt+N`	`Alt+Shift+N`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

Duplicate Button

Click the **Duplicate** button in the Page Actions Toolbar to create a duplicate of the currently selected object.



Before the new object is added to the database, a dialog appears in which you must enter the unique identifier for the duplicate object. When you click OK, the system creates the record by copying the original record. It may be helpful to think of the duplication function as the equivalent of a "save as" action.

Keyboard Shortcut
`Ctrl+Alt+D`

> **Note**
>
> Copying ancillary information for a record sometimes requires confirmation during a duplication request. If so, you may also be asked if you also want to duplicate ancillary information. If, for example, you duplicate a user group, besides supplying the ID of the new user group, you are asked if all users in the original user group should be linked to the new user group.

This button does not appear in the Page Actions Toolbar when:

Copyright © 2026, Oracle and/or its affiliates.

- Duplication is not supported for the current object.
- You are on a portal based page. (For these types of pages, if duplication is supported, the actions are defined in a zone on the portal.)
- You have made changes to the object without committing the changes to the database by clicking Save. (After making changes, you must either discard them using the **Refresh** button, or save them.)

Delete Button

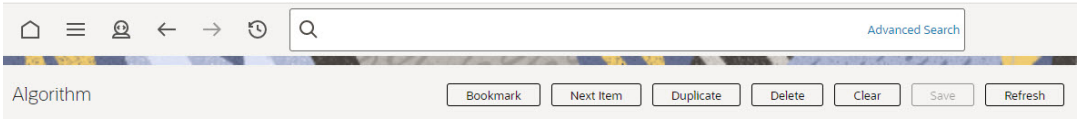
Not all transactions have the capability to be deleted. Deletion is related to the business impact of the action. In some transactions, Delete is present for some scenarios and unavailable for others; for example, you may not be able to delete a record at a certain status level.

> **Note**
>
> If you make a change to an object and do not save the changes to the database, you cannot delete the object until you either save the changes or refresh the object with the contents on the database.

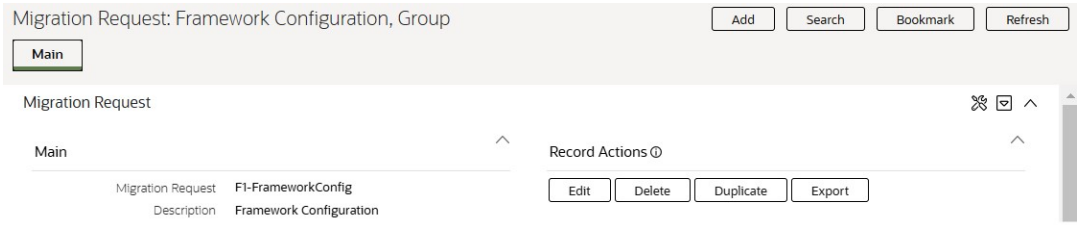
Before you delete an object, the system verifies that the object is not referenced anywhere in the system. If it is, you cannot delete it. If, for example, you attempt to delete a user, the system ensures this user has not been referenced in a user group or on a To Do.

Depending on the type of page that you are using, the option to delete may vary.

- Fixed pages display a Delete button in the page actions toolbar when the option is available.



- Portal-based pages display the Delete button in the object-related map.



- List style fixed pages commonly display the Delete button (garbage can icon) next to the entity in the list.

Copyright © 2026, Oracle and/or its affiliates.

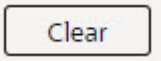


In all cases, before the object is deleted, a confirmation dialog is displayed.

> **⚠ Caution**
>
> A timeout situation could occur as the system attempts to verify the use of the deleted object if the object is referenced in many places *and* involves large volumes of production data. If this timeout situation occurs, you must with your database administration group to delete the offending object.

Clear Button

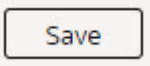
Click the **Clear** button to remove the values from the currently displayed object's fields. This action is typically used to clear input fields when adding a new object.



Keyboard Shortcut
`Ctrl+Alt+C`

Save Button

Clicking the **Save** button saves any changes you have made in the data on the current object.



Keyboard Shortcut	Alternate Shortcut
`Alt+S`	`Alt+Shift+S`

> **ⓘ Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

The button does not appear unless:

- You enter a page in add mode and default values are loaded.
- You make changes to an object after it is displayed (note that you must tab or click out of the field before the Save button appears).

Refresh Button

Click the Refresh button to reload the values on the page with those from the database.

Copyright © 2026, Oracle and/or its affiliates.



> **⚠ Caution**
>
> All unsaved changes are discarded.

Keyboard Shortcut
`Alt+R`

Portal Maintenance

The **Portal Maintenance** action appears in the Page Actions Toolbar under these conditions:

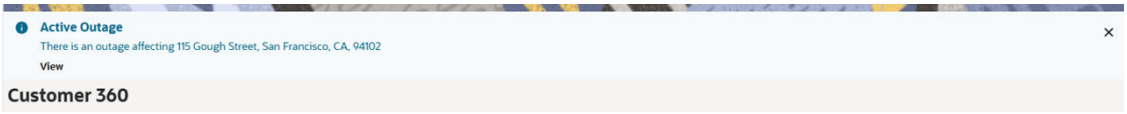
- You are on a Portal page.
- You have inquire access to the Portal application service.

Use this action to navigate to the portal definition for the portal page you are currently viewing.

> **ⓘ Note**
>
> Depending on other actions available, this action may be grouped with other actions in a '**General Actions**' menu.

Portal Banners

Important messages related to the context of a page may be displayed in the form of banners as part of the page title area.



The following describe important aspects of portal banner functionality:

- A banner message may optionally include actions that allow the user to navigate to another page or launch a drawer for more information.
- Up to three banners are displayed. When more exist, a 'View All' action is available to display all in a drawer.
- Banners can be temporarily closed while staying on the current page but navigating back to the page displays them again.

Object Display Area

The Object Display Area contains the details of the object(s) currently being viewed or maintained. For example, if you display a portal, its zones are displayed in this area.

Copyright © 2026, Oracle and/or its affiliates.

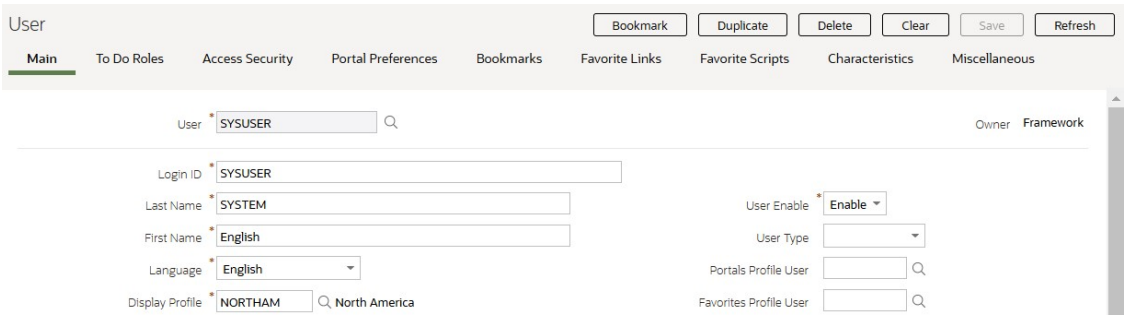
Because many of the page components follow Internet standards of operation, the behavior of these types of components are not explicitly described. This section focuses on special features that are unique to the application framework.

- Multiple Tabs
- Drawer
- Text Box
- Date and Time Fields
- Input Field with Search
- Context Menu Button
- Go To Button
- Broadcast Button
- Embedded Help Icon
- Knowledge Article Icon
- Work Lists
- Scroll Bars with Dynamic Height
- Sorting Functionality
- Get More Functionality
- Adding and Removing Rows Using Arrows
- Reorganizing Items Using the Up and Down Arrows
- Trees
- Required Field Indicator
- Accordion Windows

Multiple Tabs

When a page in the application includes a lot of information, the page may be implemented with vertical scrolling to allow the user to view all the information. Or alternatively, the page may be implemented with several tabs, allowing the user to navigate between tabs to view or update information. The following image is a section of the User page showing the multiple tabs.

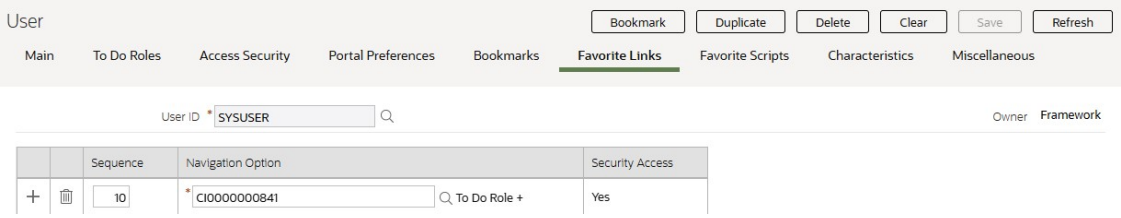
Figure Multiple Tabs Example



Copyright © 2026, Oracle and/or its affiliates.

The names of the tabs describe the type of information that will be displayed in the page when the tab is clicked. For example, if you click on the **Favorite Links** tab, the contents of the page show the favorite links configured for the user.

Figure Multiple Tabs Example (After Favorite Links Tab Is Clicked)



> **Note**
>
> **Hot key options.** Rather than clicking on a tab, you can click `Alt+F2` or `Alt+Shift+F2` to scroll through the tabs. `F2` moves you through the tabs from left to right. `Shift+F2` moves you through the tabs from right to left.

> **Note**
>
> **Hot key options.** Use `Alt+` a number to navigate to the tab page whose relative position corresponds with the number. For example, if you press `Alt+6` on the above page, the **Favorite Links** tab will be displayed. `Alt+0` may be used to navigate to the tenth tab. If there are more than 10 tabs on a page, you must use the mouse or `F2` / `Shift+F2` to display additional tabs. Note that the numeric keypad on your keyboard cannot be used to implement this function.

> **Note**
>
> **Returning to a page that contains multiple tabs.** If you return to a page (by clicking the Back arrow, the page will be positioned on the last tab that was opened.

> **Note**
>
> **Page versus Tab nomenclature.** The documentation frequently refers to the contents of a given tab as a page. However, the page actually comprises the entire collection of tabs.

Scoreboard Configuration

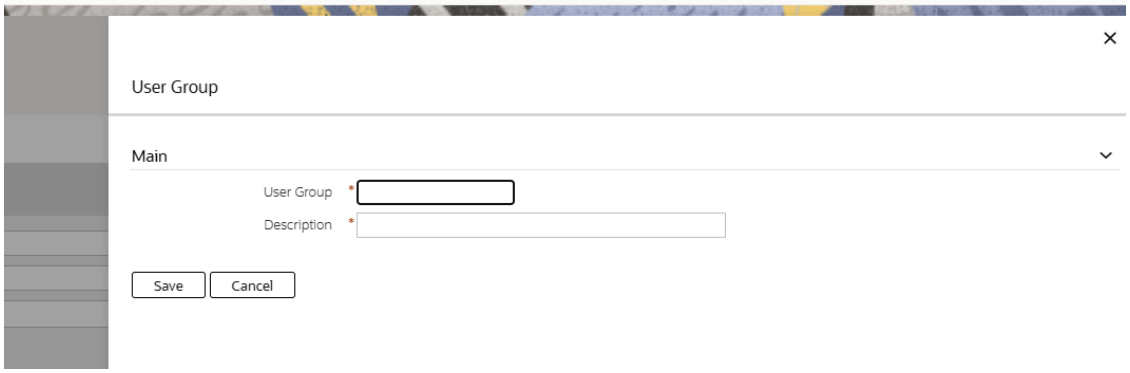
The product provides the ability for Dashboard portals to use the tab area to display a "scoreboard", where each tab includes its own specific information, called a "scorecard". A scorecard may contain any of the following information: additional text, a metric, trend information, a graph and information configured as a badge.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Advanced Configuration for Portal Title / Tabs for details on how to configure this capability.

Drawer

A drawer can be launched from the Object Display area to present related extended information. Drawers can be of small, medium and large sizes.



When applicable, a drawer may be launched from another drawer, forming a nested display of panels. You may navigate to the previous drawer using a 'back' action. Closing the top drawer closes all drawers.

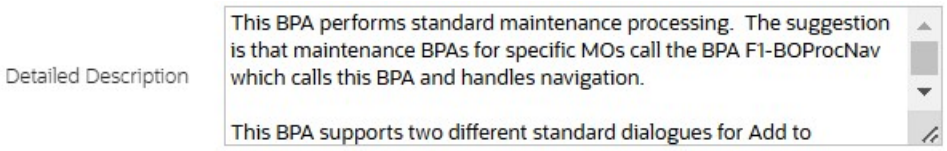
Input forms may also be displayed in a small, medium, or large drawer panel. A drawer may not be launched from an input form drawer.

Text Box

The following is an example of a basic text box:

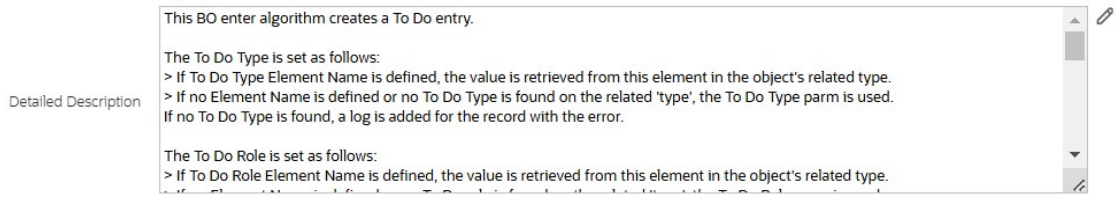


If the text box is a large field, the text box may be configured to enable scrolling. In this situation, most browsers offer a widget in the bottom outer corner that allows you to resize the editable window.

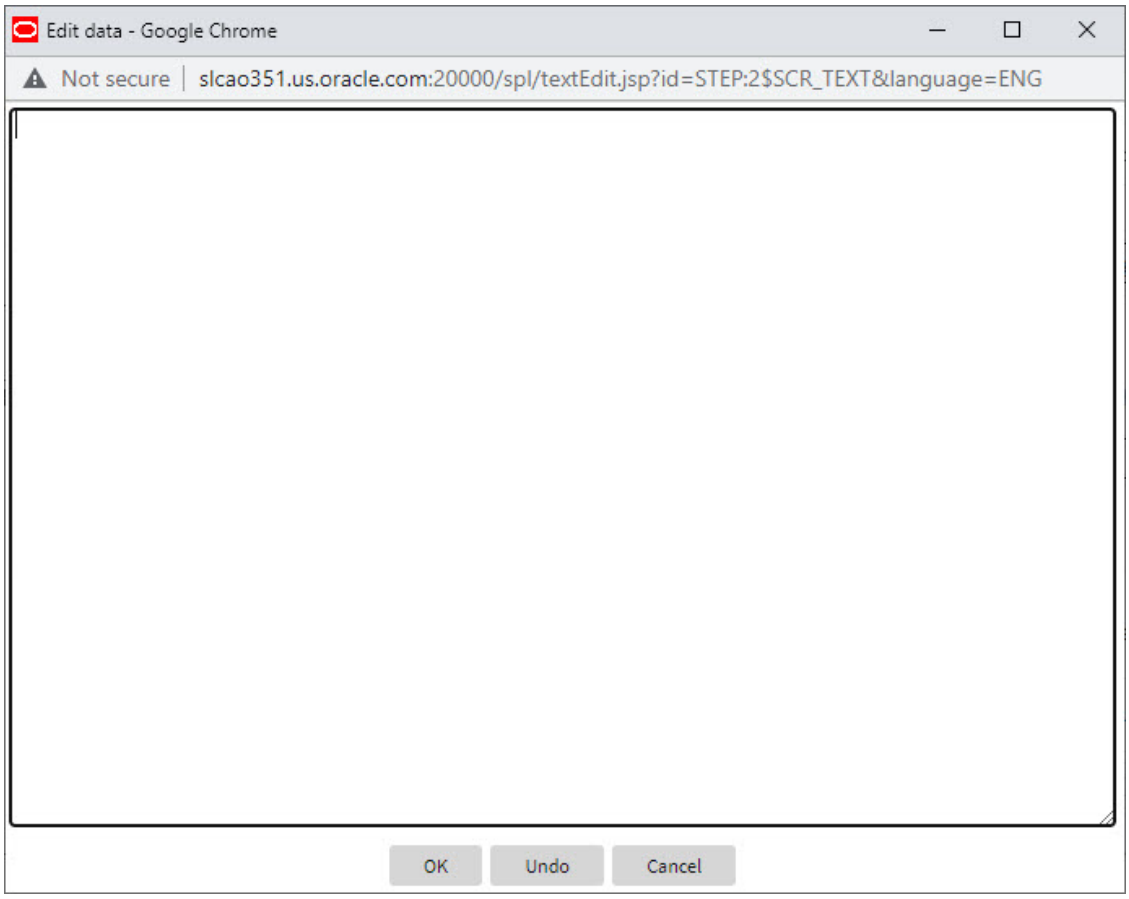


In our fixed pages, this type of text box also offers an expanded edit icon, (a pencil), that will appear adjacent to it.

Copyright © 2026, Oracle and/or its affiliates.



- If you click the expanded edit icon, an editable pop-up window appears. If the field is not protected as a result of business rules for the page, you may modify the text in this window.



> **Note**
>
> The following table highlights special characters that are converted to equivalent standard characters when entered in a text box.

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
Unicode U+201D	Straight double quote.
Curly left double quote.	

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
U+201C Curly right double quote.	Straight double quote.

Copyright © 2026, Oracle and/or its affiliates.

Unicode	Converted Character Description
Entered Character Description	
U+2019	Straight single quote.

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
Unicode U+2018 curly right single quote.	Straight single quote.

Copyright © 2026, Oracle and/or its affiliates.

Unicode Entered Character Description	Converted Character Description
U+2013 ndash.	Simple hyphen.
U+2014 mdash.	Simple hyphen.

Copyright © 2026, Oracle and/or its affiliates.

Entered Character	Unicode Description	Converted Character Description
…	U+2026	Ellipse	Three dots.

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Unicode	Converted Character Description
Middle dot (bullet/centered dot)	U+00B7	Asterisk.

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
Unicode fraction 1/4 (using 1 character).	1/4 (using three separate characters).

Copyright © 2026, Oracle and/or its affiliates.

Entered Character	Unicode Description	Converted Character Description

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Unicode	Converted Character Description
fraction 1/2 (using 1 character).	U+00BD	1/2 (using three separate character

Copyright © 2026, Oracle and/or its affiliates.

Entered Character	Unicode Description	Converted Character Description
		s) .

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
U+00BE	3/4 (using three separate character

Copyright © 2026, Oracle and/or its affiliates.

Entered Character Description	Converted Character Description
Unicode	s).

Date and Time Fields

When both the date and time are required, the Date and Time fields typically look as follows:



When only the date is required, the Date field looks as follows:

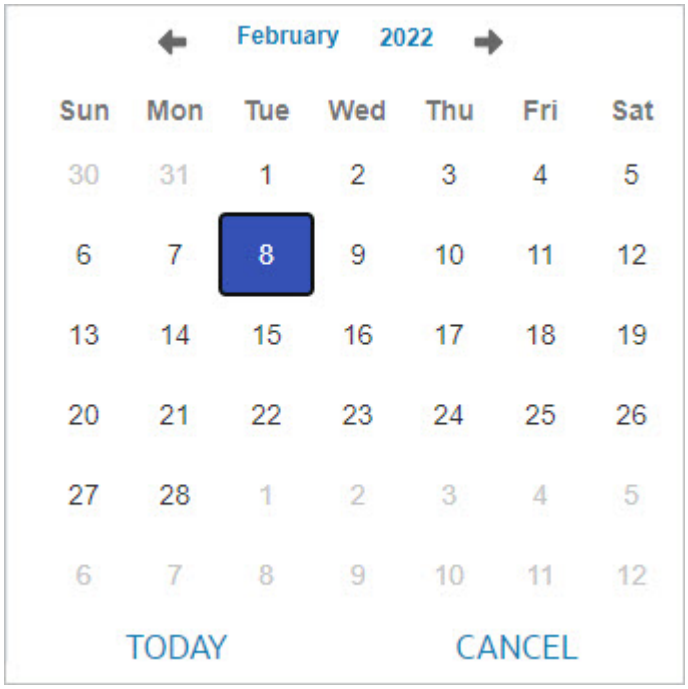


When only the time is required, the Time field looks as follows:



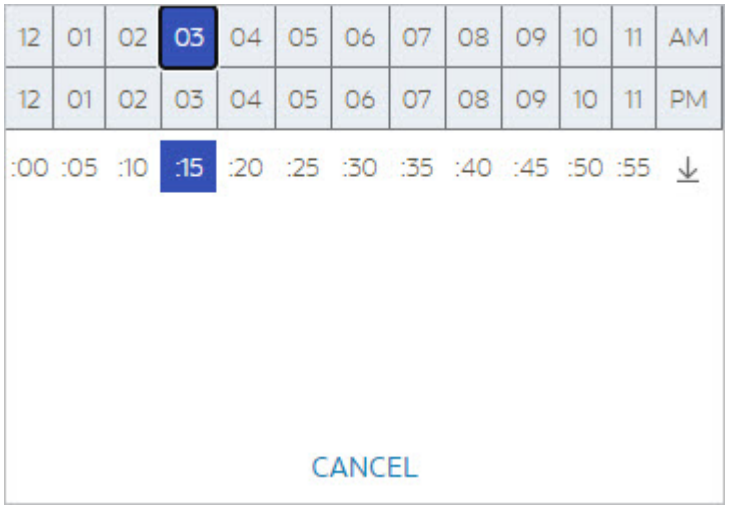
If you click on the calendar icon or press the Enter key while the cursor is positioned to the left of the icon in a Date or Date and Time field, the following window appears:

Copyright © 2026, Oracle and/or its affiliates.



Click or press enter to select the date and populate the input field. Use the left and right arrows to scroll back or forward a month. Click on the "Month" text to bring up a range of months to select from. Click on the year to bring up a range of years to select from.

If you click on the clock icon or press the Enter key while the cursor is positioned to the left of the icon in a Time or Date and Time field, the following window appears:



Select the hour then click or press enter on the minutes to select the time and populate the input field.

Input Field with Search

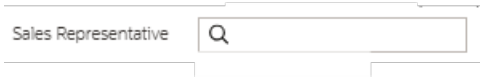
If an input field is related to an object in the system that uses a search to find a valid value, a search button appears, using the magnifying glass icon.

The position of the magnifying glass depends on the type of page you are on and configuration. If it is a fixed page, the search icon is adjacent to the input field. This is also where the icon appears in portal based input pages if the Auto-filtering option is turned off.

Copyright © 2026, Oracle and/or its affiliates.



On a portal based page, if the Auto-filtering option is turned on in your environment, the search icon is embedded in the beginning of the input field.



When the auto-filtering is turned on and the search is configured to enable auto-filtering functionality, the search also includes a down arrow.



> **Note**
>
> See below for more information about the auto-filtering functionality.

In all cases, you can have the system search for an item that matches the data you entered using any of the following methods:

- Enter a value in the search field and press **Enter** while the cursor is in the search field. The system assumes you want to search for the value entered and pre-populates it in the search window's search criteria. If a unique match can be found, the system automatically populates the field with details corresponding to the record found. If a unique match cannot be found (either no matches or multiple matches), a search page opens containing selectable items.
- Click the Search button without entering a criteria. This will typically cause a pop-up search window to appear so that you can further refine your search.
- Enter a value and tab out of the search field. The system looks for a value that exactly matches what you entered. If it cannot find such a value, the search dialog will not be presented.

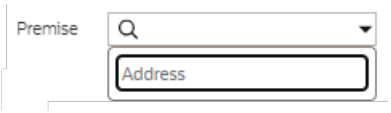
When an item is selected in the search page, the search page closes and the input field is populated with the value selected.

> **Note**
>
> **Sorting columns within the search grid.** You may click on the column heading of columns in a search grid to cause the contents of the grid to be sorted. Refer to Sorting Functionality for more information.

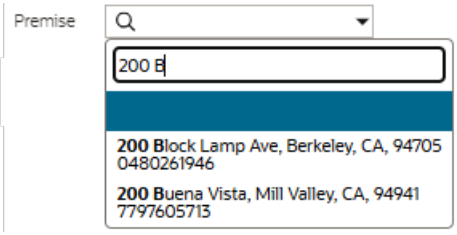
Auto-filtering

If your environment has turned on the Auto-filtering functionality, then if the search for an input field is configured for auto-filtering, the user can click the down arrow to open the filter area.

Copyright © 2026, Oracle and/or its affiliates.



Start typing in the filter area to see results that match the entered text. The search may be configured to display up to three rows of information for you to choose the correct record. The text that matches what you typed is highlighted in bold.



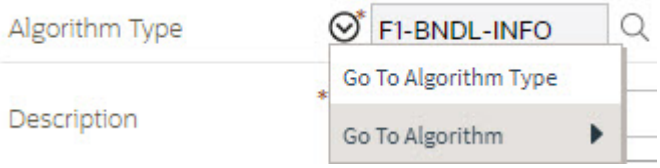
> **Note**
>
> Whether the text you type is matched only at the beginning of the string or at the middle of the string depends on how the filter is configured in the underlying search. Most search filters are configured to match text at the beginning of the string.

Context Menu Button

The Context Menu button is a small arrow that appears adjacent to an object's value or information string. An example is shown below:



Context menus exist to help you navigate to other pages and populating the page with the current context (by passing the ID of the related object). In addition, the menu items offered in a Context Menu differ depending on the entity that the menu is related to; for example, when you click the Algorithm Type context menu, a pop-up menu similar to this appears:

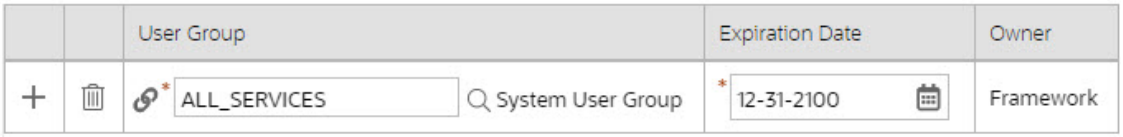


Different context menus exist for each of the major objects in the system; for example, in Customer Care and Billing, the context menu for Premise contains premise-oriented menu items, whereas the context menu for Account contains account-oriented menu items.

Go To Button

Clicking a Go To button, , opens the page for the related object - it functions like a hyperlink.

Copyright © 2026, Oracle and/or its affiliates.



Broadcast Button

Some zones have been configured to "broadcast" unique identifiers to other zones within the portal; if so, you'll see a broadcast icon, , in the first column. Clicking the icon causes specific zones to reload - these zones that have been configured to rebuild when the unique identifier is received. For example, you might have configured a zone to display the image of a bill when a bill ID is broadcast from an info zone.

Embedded Help Icon

The icon containing a small 'i' is an Embedded Help icon.

ⓘ

When an embedded help icon is clicked, a tool tip appears containing additional information.

Knowledge Article Icon

The Knowledge Article icon may appear in select contexts that are associated with a knowledge article.



Clicking the Knowledge Article icon presents the related article in a drawer panel. You may also search for and review other related articles in that panel.

Refer to Knowledge Article for configuration information.

Work Lists

The work list buttons can appear in a zone's column heading, .

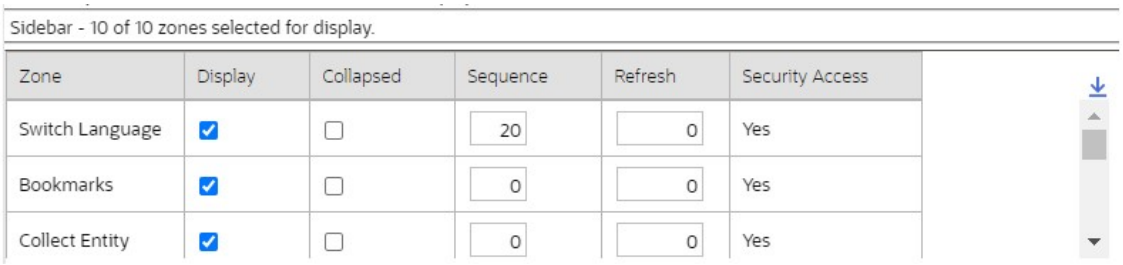
Clicking the icon results in populating the sidebar Work List Zone with the entire column. The user may use the Work List zone to easily process entries one at a time without returning back to the source query zone.

> **Note**
>
> The Work List zone is populated only upon user request.

Scroll Bars with Dynamic Height

Sometimes the height of a grid isn't sufficient to display a grid's rows. When this occurs, a vertical scroll bar appears and an "expand button" appears above this bar, . For example, the following grid is not high enough to show all of a the available sidebar zones:

Copyright © 2026, Oracle and/or its affiliates.



If you want to view all of the available zones without using the vertical scroll bar, you can you click the expand button. When clicked, the height of the grid expands to show all rows in the grid:



Notice, after the grid is expanded, the grid's vertical scroll bar disappears and the button in the upper corner becomes a Collapse Button, . If you click the collapse button, the grid will return to its original height.

> **Note**
>
> **Search grids.** Expand all functionality is not available in search grids.

Sorting Functionality

Grids have a heading row that contains labels for the columns. By clicking a column heading, you cause the rows to be sorted in the order of the selected column. By clicking on a column more than once, you cause the rows to be sorted in reverse order. Clicking the heading a third time removes the sorting function and returns the grid to its default display organization.

The following shows how information look after organizing the grid, clicking the column heading, by algorithm type (notice the icon in the column name).

Copyright © 2026, Oracle and/or its affiliates.

	Algorithm	Description	Algorithm Type
1	F1-OVRDT-ALG	Validates date in format YYYY-MM-DD	ADHV-DTD
2	F1-ACTIVEWS	Web Service - Verify Active Operations	F1-ACTIVEWS
3	F1-ATCHM-INF	Attachment Information	F1-ATCHM-INF
4	F1-ATSZCLVL	Attachment Size Class Validation	F1-ATSZCLVL
5	F1-BAT-LSDEF	Timed Batch - Service Level Based on 1.5 Timer Interval Factor	F1-BAT-LVSVC
6	F1-BFEMOLOG	Add Business Flag Error MO Log	F1-BFEMOLOG
7	F1-BFSTDNMVA	Business Flag Standard Name Validation	F1-BFSTDNMVA

> **Note**
>
> **Sorting on dropdown values.** If you attempt to sort a column that contains a dropdown value, the information is sorted in the order of the unique identifier associated with the dropdown value, not the description you see on the screen; for example, if you sort on a column containing unit of measure, the information will be sorted in the order of the unit of measure code, not the description that appears in the drop down. This means that the resultant sort will appear incorrect and therefore we discourage you from doing this.

Get More Functionality

Grids in fixed pages that can potentially contain a large number of rows have special processing that we refer to as Get More functionality. Rather than show every row, the system retrieves the first X records (where X varies from grid to grid, and from release to release) and appends a Get More button to the bottom of the grid:



When you click the **Get More** button, the system appends the next X rows to the bottom of the grid.

> **Note**
>
> **The absence of a "get more" button.** For pages configured to include Get More, if no button appears at the bottom of the grid, this means all of the rows are displayed.

> **Note**
>
> **Search grids.** The Get More function is not available in search grids.

Adding and Removing Rows Using Arrows

In some editable grids, a new row can be added to another editable grid (the grid on the left) by clicking the Add item button,  , of the desired row in the grid on the right:

Copyright © 2026, Oracle and/or its affiliates.

	Sequence	Description
	10	Winter kWh charge, first 500
	20	Winter kWh charge, remaining

	Sequence	Description
	5	Service Charge

When the Remove item button is clicked,  , the row is transferred from the grid on the right to the grid on the left. This change will not be committed to the database until you save the object being maintained

Reorganizing Items Using the Up and Down Arrows

Occasionally, rows can be repositioned in an editable grid by clicking up and down arrows. The following is an example of such a grid:

		Merge Type	Sequence	SQ Rule
		Original	10	DA Credits
		Original	20	Calc. Number of Days In Bill Period
		Original	30	Copy SQs from Master SA to sub SA

Notice the following about such a grid:

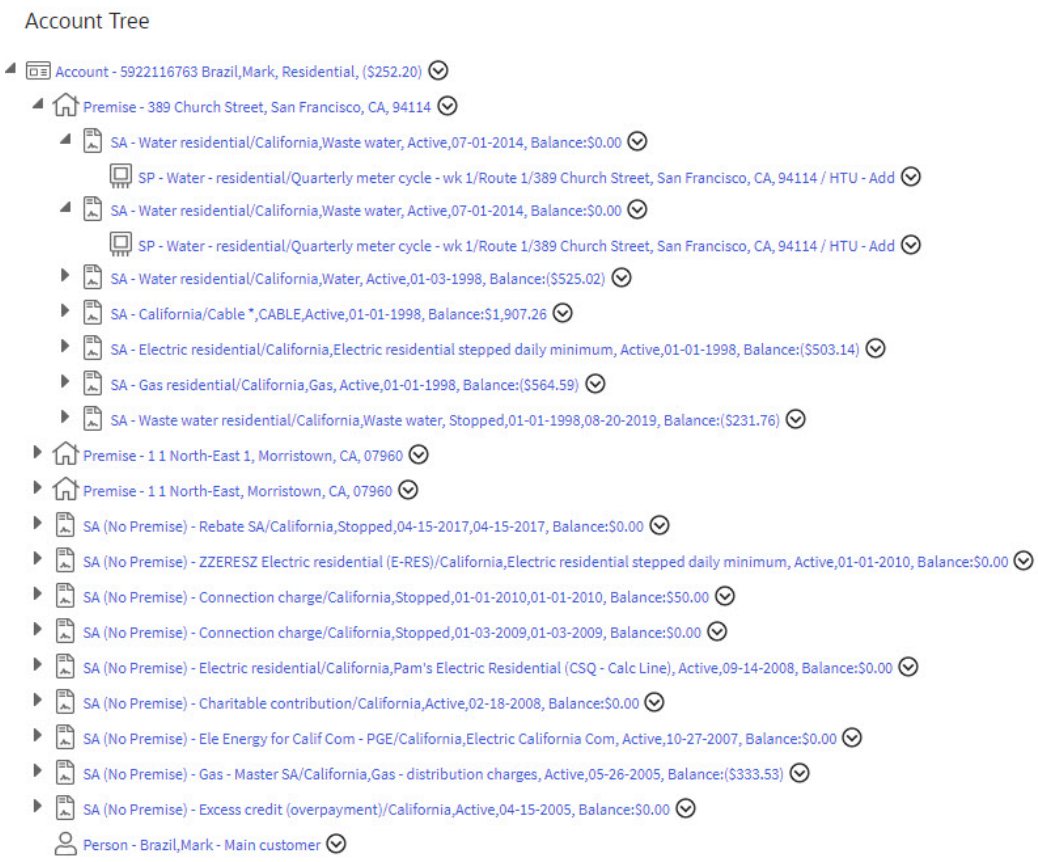
- The up arrow is dim in the first row. This is because it can't be moved further to the top.
- The down arrow is dim in the last row. This is because it can't be moved further down.
- The up and down arrows are usable in all other rows. Each time you click one of these arrows, the respective row is moved up or down one row.

> **Note**
>
> **Reminder.** When you move rows into a grid, the system does not apply the change to the database until you click the Save button (which commits the entire object to the database).

Trees

Trees are used to illustrate complex relationships between objects. The following is an example from Oracle Utilities Customer Care and Billing of a page with a tree that is expanded:

Copyright © 2026, Oracle and/or its affiliates.



The following points describe how to use trees:

You can expand and contract the nodes in a tree by clicking on the icon that prefixes the node (you can also click the arrows to expand and contract a node).

> **Note**
>
> **Expand all.** If you press `Ctrl` while clicking on an icon that prefixes a node, all child nodes (and their children) expand.

You can click the context menu button to cause a context menu to appear. You can navigate to any of the pages that are presented on this menu. Refer to Context Menu Button for more information.

You can click on the information that appears in an underlined node to drill into the page in which the object is maintained. For example, if you click on the account information, you will be transferred to the Account - Main page.

> **Note**
>
> **Important!** If you make changes after expanding a tree's nodes (e.g., by drilling down on a Premise and changing the premise's address), the tree will contract to the first node if you return to the page.

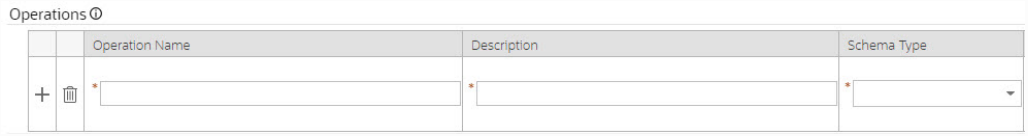
Nodes in trees fit into two broad categories:

Copyright © 2026, Oracle and/or its affiliates.

- **Information nodes.** Information nodes highlight the existence of other data. For example, on the Control Central - Bill Payment Tree, a node exists to highlight the existence of new financial transactions that were created since the last bill was completed.
- **Data nodes.** Data nodes show information about objects in the system. For example, data nodes exist on the Control Central - Bill Payment Tree that show bills and payments.

Required Field Indicator

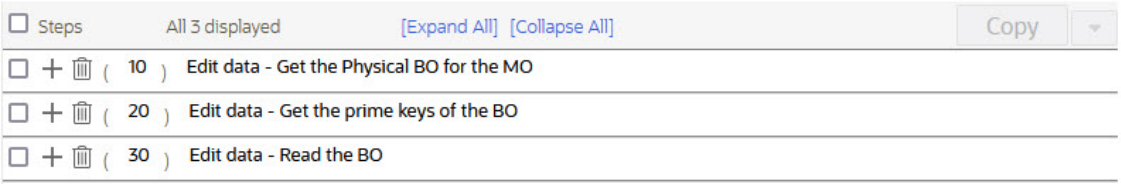
An asterisk (*) is used to indicate that a field is required. The system determines this based on the field's configuration. When a field becomes required because of an algorithm and not the field's configuration, however, the system will not add an asterisk.

> **Note**
>
> If a page includes a grid, fields within a row may be marked as required.
>
> 
>
>
> In this situation, the fields within the grid are **only** required if the row is being populated.

Accordion Windows

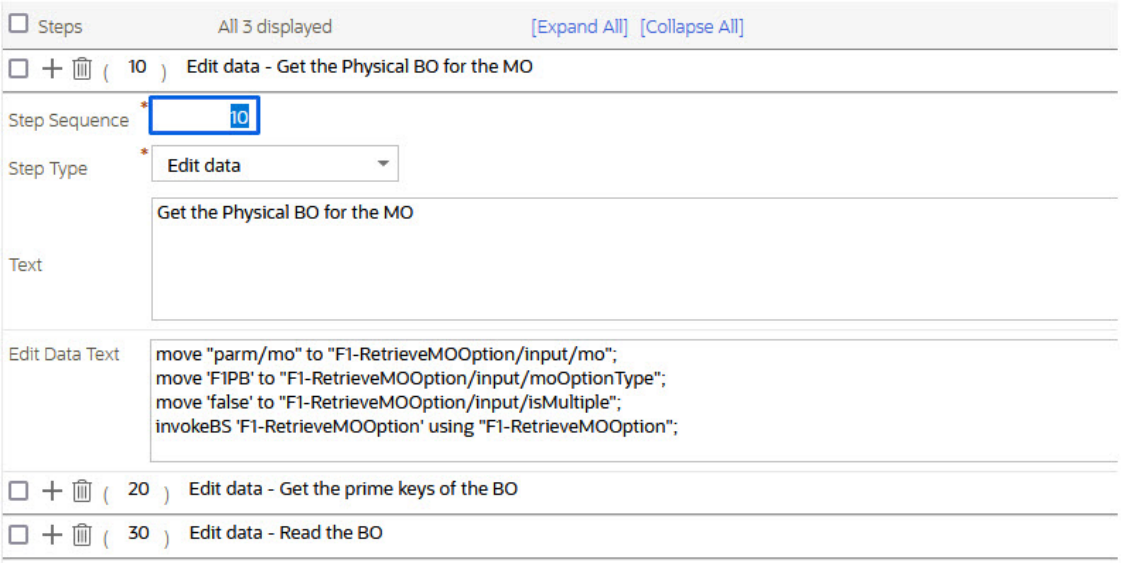
Accordion windows are used when the data held in an object's rows does not comfortably fit into a grid.

Each row in an accordion has a header that summarizes a row's details. When you click a header, the row expands to show the details. The following is an example of an accordion where all headers are collapsed (meaning that only summaries of the rows' details are shown):



When you expand a header, its details are exposed. For example, the following shows the same accordion after the third row is expanded:

Copyright © 2026, Oracle and/or its affiliates.



> **Note**
>
> **You must expand to update.** In order to update a row in an accordion, you must expand it.

Sidebar Area

The sidebar is a special area that is always available for display on the desktop. It displays the zones defined in the Sidebar Portal.

> **Note**
>
> **Open / close the sidebar**. You can open or close the sidebar using the Sidebar Toggle.
>
> **Minimize the sidebar when launching the system.** If desired, a URL parameter may be used to minimize the sidebar when launching the system, using the parameter `minSidebar=true`. Refer to Creating Application Links in External Applications for more information and examples.
>
> **Adjusting the sidebar width.** You can adjust the width of the sidebar by moving the slider that separates the sidebar from the object display area. This also updates your user preferences with the new width.

The sidebar area is displayed after the Object Display area by default. (On the right for a left-to-right language and on the left for a right-to-left language). You have control over its location, width and more as part of your user preferences.

Assistant Area

The assistant area is a special area that may appear for your product, depending on what is offered.

Copyright © 2026, Oracle and/or its affiliates.

The assistant area is located directly above the Sidebar Area and as such, is impacted by some of the appearance behavior that applies to the sidebar. If you collapse the sidebar area, the assistant area is also collapsed. If you adjust the sidebar width, the assistant area width is adjusted accordingly. However, the refresh events that impact the sidebar do not impact the assistant area.

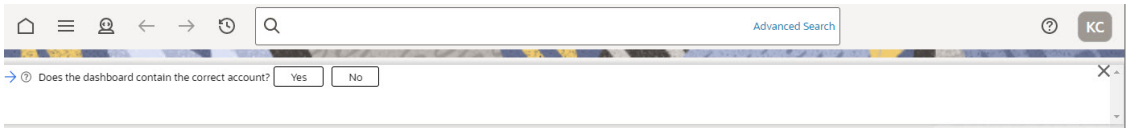
Refer to your product's specific documentation for more information about the functionality visible in this area.

Script Area

> ⚠ **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

When a script is initiated, the object display area shifts down to make room for the script area.

The information in the script area takes you through the steps in the selected business process (provided by your implementation team).



Portal Pages

This section describes using portals and zones to find and display information. This is the preferred technique used by the base product for building user interfaces.

Additionally, portals and zones may be configured by your implementation to supplement the base package provided functionality.

The contents of this section provide background information about portal page behavior.

- Portals Are Made Up Of Zones
- Users Must Be Granted Security Access To Each Zone
- Zones May Appear Collapsed When A Page Opens
- Some Portals Allow User Customization
- Sidebar Portal
- Zone Types and Characteristics
- Common Base Portals and Zones
- Process Flow

Portals Are Made Up Of Zones

The term "zone" is used to describe a section on a portal. Each zone has a heading and content.

Copyright © 2026, Oracle and/or its affiliates.

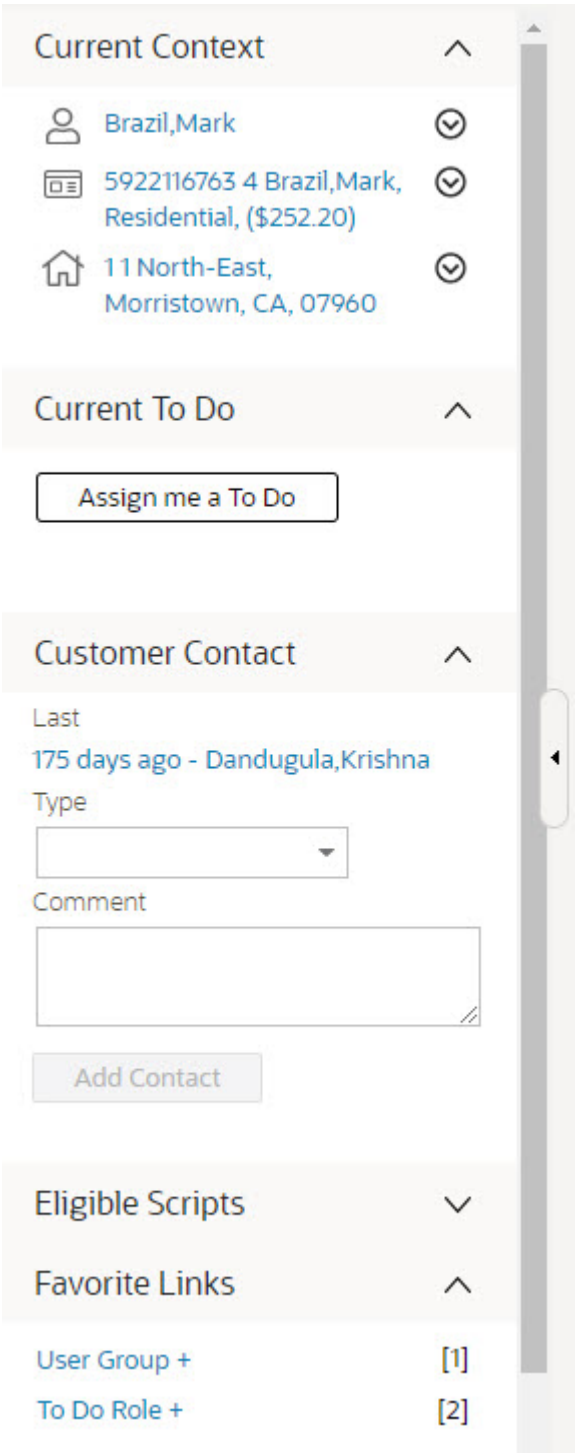
Keyboard Shortcuts	Description
`Alt+]`	Moves the cursor focus to the next zone on the page based on the standard browser keyboard navigation "order".
`Shift+Alt+]`	Moves the cursor focus to the previous zone on the page based on the standard browser keyboard navigation "order".

The following is a portal with seven zones:

- Current Context
- Current To Do
- Customer Contact
- Eligible Scripts
- Favorite Links

> **Note**
>
> This example may not be available in your application; it is used for illustration purposes only.

Copyright © 2026, Oracle and/or its affiliates.



Users Must Be Granted Security Access To Each Zone

A user must be security granted access rights in order to see a zone on a portal.

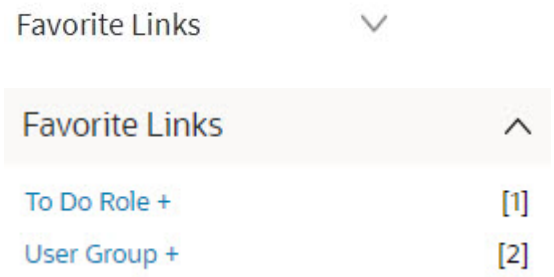
NOT_SUPPORTED

Refer to Granting Access to Zones for information .

Copyright © 2026, Oracle and/or its affiliates.

Zones May Appear Collapsed When A Page Opens

Zones may be designed to appear collapsed (or minimized) when the portal is initially displayed. To view the information in a collapsed zone, the user can press the zone's expansion button, , to either expand or collapse the zone:



A zone's data is only retrieved when it is expanded.

Some Portals Allow User Customization

There are some portals in each base product that are configured to allow user customization. In this case, the portal is visible on the user's portal preferences. For these portals each user has control over the following:

- Which zones appear on their portal
- The order in which the zones appear
- Whether the zones should be "collapsed" (i.e., minimized) when the portal opens.

Users define their preferences by clicking the User Preferences button. Pressing this button transfers the user to the User Preferences page on which their Portal Preferences can be defined.

Portals used for base product query and maintenance pages are typically configured to not allow user customization.

Sidebar Portal

The sidebar is a special portal displayed in the Sidebar Area. Its zones contain tools and data that are useful regardless of the object being displayed.

Functional Grouping. Depending on your configuration, content may be organized in functional areas with key information presented in the main focus area along with ancillary areas for favorites and tools. Each functional area is a sidebar portal enabled by the "Sidebar Configuration" extendable lookup. Functional grouping is disabled when the legacy sidebar portal "CI_DASHBOARD" is marked active and other functional sidebar portals are marked inactive in the sidebar configuration definition.

When your implementation is configured to be organized by functional area, you see icons at the bottom of the sidebar area. In this image, the 'target' icon is selected, indicating that you see the sidebar zones in the 'main' functional area. Click the 'star' icon to show the zones in the 'favorites' functional area. Click the 'wrench' icon to show zones in the 'tools' functional area.

Copyright © 2026, Oracle and/or its affiliates.



The subsequent topics describe the zones that are provided by the framework product for this portal. Please note the following additional information about the sidebar portal that may impact how it appears.

- **Additional zones may be available in other products.** The zones described below are available in every base product. Please refer to the product-specific documentation for additional sidebar zones available in your product.
- **Context Sensitive Zones may appear.** Some pages may define additional zones that should appear in the sidebar only if that page is displayed. These are referred to as Context Sensitive Zones.
- **Sidebar zones are configurable.** The sidebar portal is configured to allow a user to define preferences for it. Refer to Portal Preferences for more information.

> **Note**
>
> **Security.** As a general rule, sidebar zones each refer to a unique application service so that a security administrator may configure which users are allowed to have access to which zones. Note that the user portal preferences indicates which zones a user has security access to.

- Bookmarks Zone
- Collect Entity Zone
- Current To Do Zone
- Favorite Links Zone
- Favorite Scripts Zone
- Favorite Searches Zone
- Switch Language Zone
- Tags Zone
- To Do Summary Zone
- Work List Zone

Bookmarks Zone

After clicking the Bookmark button and saving a bookmark, it is visible in the **Bookmarks** zone. Clicking one of the links in the zone navigates to the page where the bookmark was saved and loads the information related to the record that was displayed when the bookmark was saved.

Click the trash can icon to delete the bookmark.

Note that a user may change the name of the bookmark or change the sequence of the bookmarks by navigating to the User - Bookmarks in the user preferences.

This information is also visible in the Favorites, Bookmarks section of the menu.

Copyright © 2026, Oracle and/or its affiliates.

Collect Entity Zone

The **Collect Entity** zone provides a quick way to add or remove the entity currently being displayed to and from the migration request in context. This is mainly useful when reviewing data to export and while doing so maintain the export list. The zone automatically appears once an **Entity List** migration request is visited and set in context.

Current To Do Zone

The **Current To Do** zone is designed to streamline the process of working on To Do entries from assignment to completion. It displays key information about the user's current To Do entry and provides functions that help minimize the number of steps it takes to resolve the entry and potentially other entries related to it.

> **Important**
>
> The information here assumes that you have configured the system to use the ability to have a substatus of Assigned, which includes the **In Progress** value. If you are not taking advantage of the substatus, then see below for information about how the functionality described here works differently.

The following sections provide more detail about the zone.

To Do Entry Visible

If you have a To Do Entry that is currently **In Progress**, its information is visible in this zone when you launch the system. As described in Working On A To Do, most of the actions in the system that cause a new To Do Entry to be pushed to this zone cause that To Do Entry to be updated to **Assigned / In Progress**. If you use those actions then the To Do Entry in this zone will always be your current **In Progress**. However, there are use cases that populate the To Do entry without assigning it to you. If you perform those actions, the To Do Entry that is pushed to this zone and remains until you perform a subsequent action that refreshes the information in this zone.

Next To Do Assignment

Users may have the option to click the **Assign me a To Do** button to request the next To Do entry they should work on. This functionality is optional and can be configured to follow your organization's business rules in determining the next task for a user. For example, if your organization practices work distribution "on demand", this feature can be configured to find the highest priority entry a user can work on and assign it to the user. Refer to work distribution for more information on this business practice.

> **Note**
>
> Refer to On Demand Assignment Configuration for more information on how to configure the system to take advantage of this function.

If the system is successful in determining the next entry the user should work on, this entry is assigned to the user and moved to In Progress and becomes the user's current To Do. Its information is displayed on the zone.

Copyright © 2026, Oracle and/or its affiliates.

To Do Information

The user's current To Do may be set using the Next Assignment algorithm, or by actions described in Working On A To Do. Once a current To Do has been set, the zone displays details related to the To Do.

The **Message** describes why the entry exists. The message description allows the user to drill into the page on which the entry's respective object is displayed. For example, if the entry is associated with an asset that needs review, the user is brought to the asset page to view the information and potentially take action. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched.

Info displays the standard To Do information. An icon appears if there are comments on the To Do and a Log Entry icon appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do. Clicking the description drills into the To Do entry page allowing the user to view more information about the entry or make changes to it as needed.

Depending on the status of the To Do entry and whether you are allowed to perform the actions, the following action buttons may be visible.

- You may use the **Complete** button in the zone to complete the current entry.
- If the current entry is **In Progress** and assigned to you, you may use the **Hold Progress** button to put the entry into the **On Hold** status. This is useful if your implementation wants an accurate picture of how long it takes to work on a To Do entry. You can put the entry on hold if you need to wait for some event to occur before continuing progress on the entry.
- You may see the **Start Progress** button if the current entry is assigned to you and not currently in progress. You may also see this if the current entry is Open and in a role that you are in. (This may happen if you drill into an open To Do from the To Do Search page.) Clicking this button puts the entry into **In Progress**.
- You may see the **Stop Work** button if the current entry is assigned to you and is not currently **In Progress** or **On Hold**. Clicking this button returns the entry to the **Assigned** state (resetting the assigned substatus).

Next / Previous Buttons

The **Previous** and **Next** buttons are enabled on this zone under the following conditions:

- The user selected an entry either on the To Do List page or the To Do Search page.
- At least one other entry exists *below* (for Next) or *above* the entry selected (for Previous).

> **Note**
>
> The system recommendation is to use the To Do Management or To Do Dashboard / To Do Details portals to manage work lists of To Do entries. Those pages use the worklist zone to manage the list and these Next and Previous buttons are not applicable.

Clicking **Next** or **Previous** will bring the user to the transaction indicated on the To Do Entry's message, where its respective object is displayed. It also makes that To Do the Current To Do displayed in the zone.

Copyright © 2026, Oracle and/or its affiliates.

Keyboard Shortcut	Comments	Alternate Shortcut
Alt+Y	Previous To Do	
Alt+Z	Next To Do	Alt+Shift+Z

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

> **Note**
>
> When a user drills down on a To Do entry from the To Do List page, the system automatically changes the status of the entry to **Assigned** and **In Progress**. If **Next To Do** or **Previous To Do** is subsequently used to scroll through this list, the system does *not* change the To Do entry's status. You may click the **Start Progress** button in this zone.

Related To Do Entries

If the To Do entry has non-complete related entries a summary **Related To Do** description appears. The description states the total number of related entries, how many are still in **Open**, how many are assigned to current user and how many assigned to others. This hyperlink navigates to the To Do Search query allowing you to review the related entries. At this point you may decide to assign entries that can be positively identified as being caused by the same problem as the current To Do also to you.

If at least one related To Do entry is **Open** or also assigned to you, the **Complete All** button is visible. If you determines that the related To Do entries have been resolved with the appropriate action taken, the button may be used to complete in one step the current entry and all related entries that are **Open** or assigned to you.

If the related entries are not resolved with one action, you may opt to use the **Complete** button to complete the current entry alone and work on related entries one at a time. Using the related To Do information hyperlink you may navigate to the To Do search page and drill into the first related To Do to work on using the message hyperlink. This entry now becomes you current To Do and the **Next** and **Previous** action buttons are set to scroll through the related entries list. It's not automatically assigned to you. Use the **Start Progress** button in this zone to assign it to you and start progress.

Implementations Not Using Substatus

If your implementation is not taking advantage of the substatus as described in The Lifecycle Of A To Do Entry, some functionality described above works differently.

When launching the system, no To Do Entry is populated in this zone by default. To Do entries only get populated in this zone when you take an explicit action.

Any action that automatically assigns you a To Do Entry only performs that step and does not update it to **In Progress**.

The buttons **Start Progress**, **Hold Progress** and **Stop Work** are never visible.

If you are using To Do Search and drill in on an entry before assigning it to you, you'd need to go back to To Do Search page or drill into the To Do Entry page to assign it to you. Similarly, if

Copyright © 2026, Oracle and/or its affiliates.

you use the Next / Previous buttons to work through a list from either To Do Search or To Do List where the entries are not already assigned to you, the recommendation is to return to the respective pages to assign them to you.

Favorite Links Zone

Each line in the **Favorite Links Zone** corresponds with one of the user's favorite links (e.g., transactions). Clicking on a line invokes the transaction. The first nine entries can be invoked by pressing an accelerator key (`Ctrl` + the number of the relative position of the link in the list). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Users define their favorite links by clicking the User Preferences button. Clicking this button transfers the user to the User Preferences page on which their Favorite Links can be defined.

This information is also visible in the Favorites, Bookmarks section of the menu.

Favorite Scripts Zone

The Favorite Scripts zone lists the business process assistant (BPA) scripts that you have defined on your user preferences. You can click on a script to execute it. You can click the **Search** button in the zone to search for other scripts.

> **Note**
>
> If you do not see the Favorite Scripts zone in the sidebar, navigate to User Menu and click the **Preferences** menu item to display the User - Portal Preferences page. Under the Sidebar Portal row, select the **Display** option for the Favorite Scripts zone.

This information is also visible in the Favorites, Bookmarks section of the menu.

Favorite Searches Zone

If you marked a saved search as a favorite using Saved Searches functionality, the search is visible in the **Favorite Searches** zone. Clicking one of the links in the zone navigates to the page where the search was saved and populates the filter values and executes the search.

This information is also visible in the Favorites, Bookmarks section of the menu.

Switch Language Zone

The **Switch Language** zone is used to switch the application's language. Only languages that are supported by your implementation will appear.

To switch the application's language, choose the desired language and press the **Refresh** button.

> **Note**
>
> This zone only appears if more than one language is enabled in your implementation. You may further limit which users see this zone by not granting security rights to the zone's application service.

Copyright © 2026, Oracle and/or its affiliates.

Tags Zone

The **Tags** zone provides a quick way to associate the entity currently being displayed with an entity tag or remove any of the tags it currently has. Tagging an entity may be useful to highlight various contexts in which it can be used, for example for testing or reporting purposes. Refer to Defining Entity Tags for more information.

Note that the zone is visible only when there is an entity in context, i.e. there is an entity to tag, and the user has application security access to this zone.

To Do Summary Zone

The **To Do Summary Zone** may appear in the sidebar area for backward compatibility reasons. The zone has several purposes:

- It presents a summary of the To Do entries that have been assigned to you.
- The colored "age bars" highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is user-configurable. Note, you can hover the cursor over an "age bar" to see the number of To Do entries.
- If you click on the To Do Type description, you'll be transferred to the To Do Management portal where these To Do entries will be displayed. From this page, you can drill down to the specific entries.

> **Note**
>
> **Time saver!** Use the worklist button when working on an entry in the To Do Management portal. All the individual To Do entries will be visible in the Worklist sidebar zone, allowing you to work through additional To Do entries without returning to the To Do Management portal.

Work List Zone

The **Work List** zone is populated when a user click the Work Lists icon in an info or query zone. When you click the link of the work list entry, the system does whatever it would have done if you had clicked on the hyperlink in the zone. The system marks the current entry with a star. It marks any previously worked entry with a checkmark.

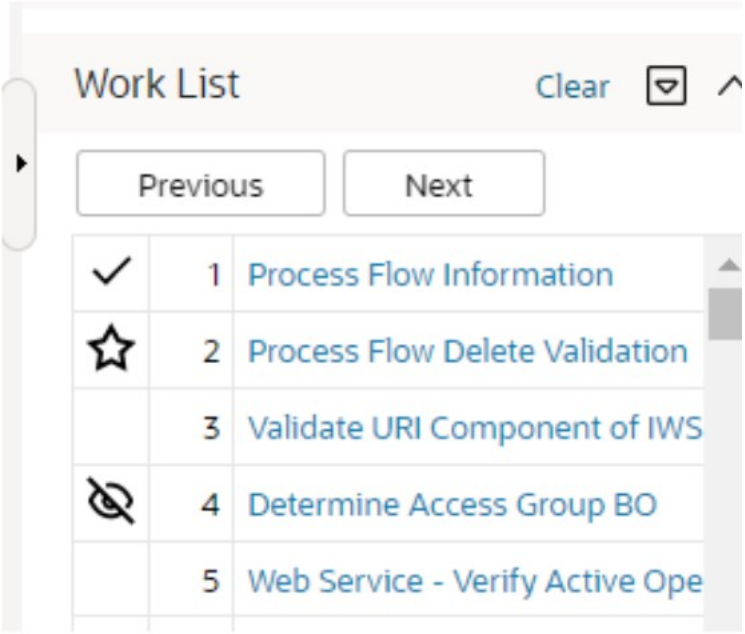
Previous and Next buttons are available in the zone allowing you to go to the next or previous entry without having to locate it yourself and click the link.

You may actively mark an entry to be ignored by the Previous or Next button by clicking the icon column for that entry. An icon of an eye with a slash through it is visible. You can click the icon to re-include that entry in the previous / next action.

The work list stays populated until you either click the Clear link in the zone's header or you overlay the work list by clicking a different column's work list icon.

The image shows a work list zone illustrating that the first entry was "visited", the second entry is the current one being viewed and the fourth entry is marked to skip or ignore.

Copyright © 2026, Oracle and/or its affiliates.



Zone Types and Characteristics

The topics in this section describe common zone types, function, and common characteristics.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Portals and Zones for a description of general portal and zone functionality.

- Common Characteristics of Zones
- Explorer Zones
- Map Zones
- Tabbed Display Zones
- Timeline Zone

Common Characteristics of Zones

The following topics describe common characteristics of all zone types supported by the base package.

- Zone Visibility

Zone Visibility

A zone may be configured to dynamically be visible or completely hidden depending on the information being displayed. When hidden, not even the header is displayed.

For configuration information, refer to Zone Visibility Service Script.

Copyright © 2026, Oracle and/or its affiliates.

Explorer Zones

The product provide several zone types that are used to find or display a set of information. These are called explorer zones. There are two types of explorer zones:

- **Info zones.** An info zone displays a list of information about an object. These types of zones often display information without requiring any input from the user.
- **Query zones.** A query zone allows users to find objects matching a given search criteria.

Virtually every aspect of an info or query zone is controlled by its zone configuration. However, most share basic common components described in the subsequent sections.

- Search Criteria / Filter Area
- Description Bar
- Results Area
- Title Bar
- Configurable Columns
- Multi-Query Zones

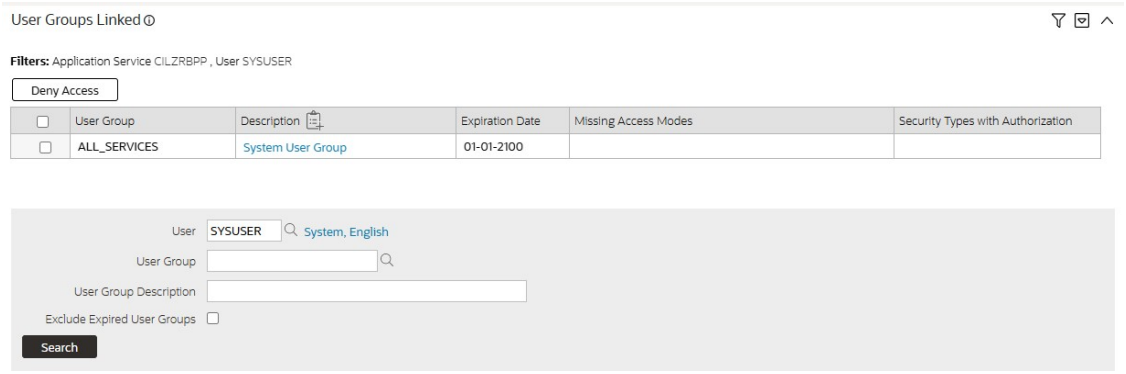
Search Criteria / Filter Area

For query zones, the Search Criteria Area (also called Filter Area) contains the criteria used to find objects. The design of the zone controls the number and type of filters that appear in this area. After search criteria is entered, clicking the **Search** button causes the system to look for records that match the criteria. The results are shown in the Results Area.

The system automatically collapses the filter area to allow more space for displaying the results. At that time the **Expand Filters** button may be used to open the filter area to enter different criteria. When the filter area is expanded the **Hide Filters** button may be used to collapse the filter area again without performing a new search.

For info zones, the zone's configuration indicates whether or not a Filter area is applicable and if so, whether or not it is open or closed by default. If the filter area is configured, the Show Filter icon is visible in the zone header. Click this icon to expose the Filter Area, if it is not visible and click the icon to hide the Filter Area if desired. Use this area to filter the information that appears in the zone by different values. Click the **Search** button to rebuild the Results Area using the specified filters.

For example, the following shows an info zone listing user groups that have access to a given Application Service where a filter area has been configured to allow users to refine the list by various criteria.



Copyright © 2026, Oracle and/or its affiliates.

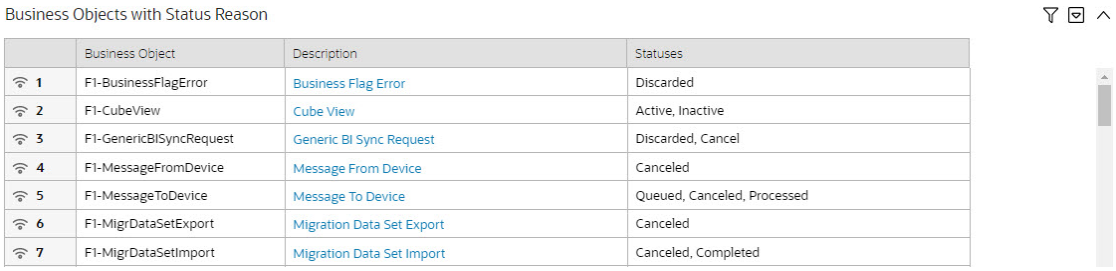
The info zone includes configuration to indicate whether the filter area, when exposed, should be shown at the top of the zone (above the results) or at the bottom of the zone (below the results). For query zones, the search criteria is always displayed at the top.

Description Bar

The Description Bar describes the search criteria that were used to build the Search Results Area.

Results Area

The Results Area displays the appropriate list of objects. For an info zone, this is the list of appropriate objects with any input filter criteria considered. The following is an example of the results for an info zone. For a query zone, this is the list that matches the search criteria defined in the Filter Area. The following is an example of the results set of an Info Zone.


Business Objects with Status Reason

	Business Object	Description	Statuses
1	F1-BusinessFlagError	Business Flag Error	Discarded
2	F1-CubeView	Cube View	Active, Inactive
3	F1-GenericBISyncRequest	Generic BI Sync Request	Discarded, Cancel
4	F1-MessageFromDevice	Message From Device	Canceled
5	F1-MessageToDevice	Message To Device	Queued, Canceled, Processed
6	F1-MigrDataSetExport	Migration Data Set Export	Canceled
7	F1-MigrDataSetImport	Migration Data Set Import	Canceled, Completed

For a query zone, the columns that appear may differ depending on the search criteria. For example, if you search for a customer by their driver's license number, the resulting columns may be different than the ones displayed when you search by customer name.

The following points highlight details related to viewing and working with the result set:

- **Broadcast.** If the results set is configured to display a broadcast icon,  , it means that at least one other zone on the portal is configured to display data related to the record selected in the results list. Clicking this icon will cause the other zone or zones to be built or refreshed to display related data.
- **Hypertext.** One or more columns in the zone may be configured as hypertext. Clicking the hypertext will result in navigating to another page to display data related to that column based on the configuration for that column.
- **Maximum height.** The maximum height of the report area is controlled by the zone's configuration. A vertical scroll will appear when more rows exist than fit into the height.
- **Maximum rows.** The maximum number of rows that can appear is controlled by the zone's configuration. Excess rows are truncated before display. Please note, you can configure the zone to export more rows than appear in the Results Area.
- **Control column sorting.** By default zones are designed to allow a user to sort the results list by clicking any column heading. Clicking once sorts ascending. Clicking again sorts descending. However, a zone may be configured to not allow this type of sorting.
- **Work list.** Columns may be configured to enable the work list icon,  . Clicking this pushes all the data into the Work List zone.
- **Rearranging columns.** A zone may be configured to allow a user to rearrange columns and add or remove columns visible in the Result Area. Refer to Column Configuration for the details.

Copyright © 2026, Oracle and/or its affiliates.

The type and variety of columns that appear in the Results Area is controlled by the zone's configuration.

Title Bar

The Title Bar contains the zone's description along with other icons and functions described in this section.

In the corner of the Title Bar, you will see an indicator that tells you whether the zone is collapsed or expanded. Each time you click the indicator, the zone toggles between expanded and collapsed. The following image shows a collapsed zone.



> **Note**
>
> Recommendation: For zones that are configured to show on portal preferences, set up Portal Preferences to collapse zones that aren't needed every time a portal is displayed. Rationale: The system doesn't build collapsed zones when a portal is displayed. Instead, collapsed zones are only built when a user expands them. Thus, indicating a zone is collapsed improves response times.

If the zone is configured with help text, a help icon appears to the right of the zone title. Click this icon to view the help text.

The following sections describe additional icons and tools visible on the title bar. Note that in addition to the information described in this section, additional actions that are specific to a zone may appear based on configuration. The actions may appear in the form of hypertext, an icon or a button.

- Explorer Zone Menu
- Show Filters
- Show Column Configuration Area
- The Open Zone Maintenance Icon

Explorer Zone Menu

Click the Explorer Zone Menu icon  to list available tools. The following topics describe each tool.

- Clear Filters
- Export Data
- Save Search Options

Clear Filters

Click the **Clear Filters** link in the Explorer Zone menu to clear the filter values and zone data.

Export Data

Click the **Export** link in the Explorer Zone menu to download the zone's data in csv format. This link is hidden if the zone has been configured to not download rows.

Please note the following with respect to this feature:

- Clicking export performs the search because it is possible to configure a number of records for download that differs from the number of records configured to display.

Copyright © 2026, Oracle and/or its affiliates.

- Because the search is performed, you can request an export even without first searching on the zone. The filters populated (or not populated) at time of exporting are the ones used for the export. If the user did not search at all or did a search and then changed the filters and exported before searching again, the results in the export may not match what is visible on the zone.
- It is possible through zone configuration to mark columns that are visible to the user interface as suppressed in the export. Conversely, it's also possibly to mark columns as suppressed on the user interface but included in the export. As such, the columns visible in the export may not always match what is visible in the user interface. This applies when for zones that support configurable columns. Columns removed from view through this technique are still included in the export if not marked to suppress.
- If the column results include any HTML, the HTML is stripped out before exporting.

Save Search Options

Query zones and info zones that include filter areas, allow a user to capture and save filter criteria so that the same search can be more quickly executed in the future. Users may save many different search options for the same zone. Users also have the option to nominate one search as the default so that this search is automatically loaded upon entering the zone. In addition, users may choose to set one or more searches as a Favorite, which causes the search to appear in the Favorite Searches zone in the sidebar, as well as through the Menu. This allows a user to navigate to the portal for that zone and load the saved search in one click.

The Explorer Menu includes various options for supporting Saved Searches that are described below.

Note the following additional points about this functionality:

- **Click Search before Saving.** When a user wants to save a search, the search criteria should be entered and the **Search** button should be clicked prior to saving the search.
- **Criteria is saved, not results.** Although a user must click **Search** prior to saving the search, the system saves the criteria, but not the results. When returning to the query or info zone in the future and opening a given saved search, the results displayed will be the current records in the system that match the saved search criteria.
- **Multi-Query Zones.** When a user is on a multi-query zone, the searches are saved at the multi-query zone level so that all saved searches for the various search dropdowns for a given multi-query zone are visible in the Open Saved Searches menu option.
- **Functionality is Secured.** If for any reason an implementation wishes to restrict which users are allowed to save searches, the functionality is securable. Only users with **Inquire** access to the `F1_USRFAVSCH` application service will see these options in the explorer menu. This is the same application service that is used to provide access to the Favorite Searches sidebar zone.
- Save As
- Open Saved Search
- Save
- Set As Default
- Remove As Default
- Add To Favorites
- Remove From Favorites
- Delete

Copyright © 2026, Oracle and/or its affiliates.

Save As

The **Save As...** option in the Explorer Zone menu allows a user to save the current entered search criteria. This should be clicked after executing the search.

The user is prompted to provide a name for the search. This search is now available from the Open Saved Searches option. If desired, a user can also choose to make this search the Default or may choose to Add to Favorites.

This option is visible whether or not an existing saved search is loaded. When a saved search is loaded, this option allows a user to make changes to the search criteria and save a new search with that criteria.

Open Saved Search

The **Open Saved Search...** option in the Explorer Zone menu is visible if there are any saved searches for the zone that is currently visible. Select this menu entry to view the current list of saved searches for the zone. For multi-query zones, any saved search for any of the query zones that are part of the multi-query zone are visible. Choose the desired saved search. The criteria is updated with this information and the search is performed.

Save

The **Save** option in the Explorer Zone menu is visible if a Saved Search is already loaded. This option allows a user to save changes to the criteria of an existing saved search (overwriting the previously saved criteria).

Set As Default

The **Set As Default** option in the Explorer Zone menu is only visible if a saved search is loaded and it is not currently the default. Click this option to nominate this search to be the one that is loaded when navigating to this zone from the menu.

Remove As Default

The **Remove As Default** option in the Explorer Zone menu is only visible if a saved search is loaded and it is currently the default. Click this option to no longer load this search automatically when navigating to this zone from the menu.

Add To Favorites

The **Add To Favorites** option in the Explorer Zone menu is only visible if a saved search is loaded and it is not currently included in the favorite searches zone. Click this option to include this search in the Favorite Searches zone. Once a search is in the Favorite Searches zone, you can use the link to the search in the sidebar to navigate to the portal for this zone and load the saved search in one click.

> **Note**
>
> **Not available for dependent zones** There are some portals that include a zone that is not the main query or info zone on the portal, but is rather a zone providing additional information for a record that is displayed. If a zone like this has filter options or configurable columns, a user is able to save searches and mark one as a default, but will not be able to add the saved search to favorites because the zone is dependent on other data on the page that is not captured for the individual zone.

> **Note**
>
> **Sidebar Zone Preferences.** Be sure that your portal preferences are configured to include the Favorite Searches sidebar zone.

Copyright © 2026, Oracle and/or its affiliates.

Remove From Favorites

The **Remove From Favorites** option in the Explorer Zone menu is only visible if a saved search is loaded and it is currently included in the Favorite Searches zone. Click this option when you want to keep the saved search but you no longer want it visible in your Favorite Searches zone. (Deleting a saved search that is also marked as a favorite will automatically remove the search from the favorite searches zone.)

Delete

The **Delete** option in the Explorer Zone menu is only visible if a saved search is loaded. Click this option to remove the currently opened saved search from the list of saved searches.

Note that if the saved search had been configured as favorite, deleting the saved search remove it from the favorite searches zone.

Show Filters

Click the **Show Filters** icon  to show the Filter Area. This button only appears if the zone has been configured to have filters.

> **Note**
>
> If the zone is configured to have filters and the drag and drop (column configuration) area, only one or the other is shown at any given time. It means that a user can either change the filter criteria and refresh the search results or change the column configuration and refresh the results.

Show Column Configuration Area

Click the **Column Configuration** icon  to show the Column Configuration Area. This button only appears if the zone has been configured to allow columns to be dragged and dropped into the Results Area.

> **Note**
>
> If the zone is configured to have filters and the drag and drop (column configuration) area, only one or the other is shown at any given time. It means that a user can either change the filter criteria and refresh the search results or change the column configuration and refresh the results.

The Open Zone Maintenance Icon

Click the **Go To Zone** icon  to switch to the zone's zone maintenance page. On this page, you can change the parameters that control the zone's behavior.

> **Note**
>
> **Debug mode only.** This icon only appears when the system operates in Debug mode.

Configurable Columns

Info zones and query zones may be configured to allow users to rearrange columns and to remove or add columns to the output. This is also referred to as "drag and drop" of columns.

Copyright © 2026, Oracle and/or its affiliates.

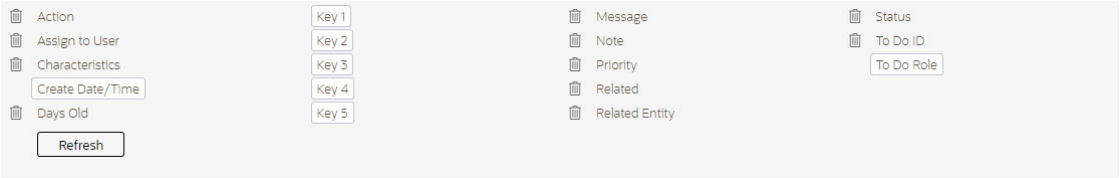
When a zones is configured to support this functionality, the Show Column Configuration Area icon is visible in the zone title area. Click this icon to expose the Column Configuration Area, if it is not visible and click the icon to hide the Column Configuration Area if desired.

If configurable columns are allowed, a user may rearrange the order of columns currently displayed in the Results Area by clicking the column heading and "dragging" it to the desired location.

In addition, the column configuration area may display columns that are not currently in the Results Area. These are indicated by a color cream background. A user can drag one of these columns to the results area and then click **Refresh** to rebuild the zone.

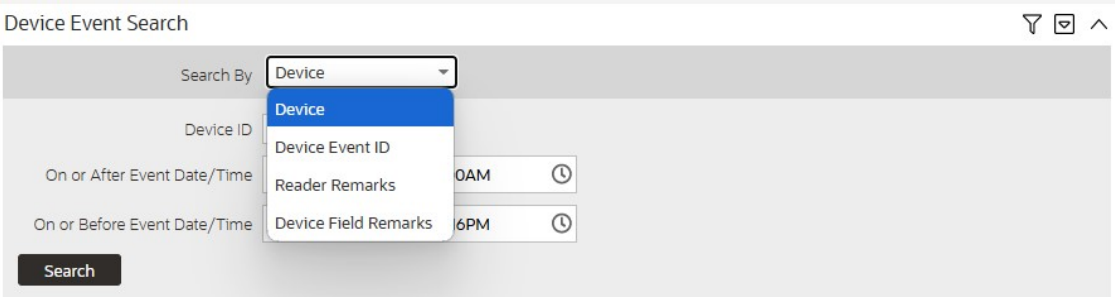
To remove a column that is currently in the results area, click the trash can icon next to the column name in the column configuration area and then click **Refresh** to rebuild the zone.

The following is an example of how the Column Configuration Area looks:



Multi-Query Zones

This type of zone combines individual Query Zones into a single query zone. A **Search By** dropdown, built from the zone descriptions of the individual Query Zones, allows you to select the desired query to execute. Selecting a specific Query Zone display the filters of that query in the filter area. At this point the zone simply executes the selected Query Zone, as if it was executed on a standalone zone. By default, the first query zone option is selected.

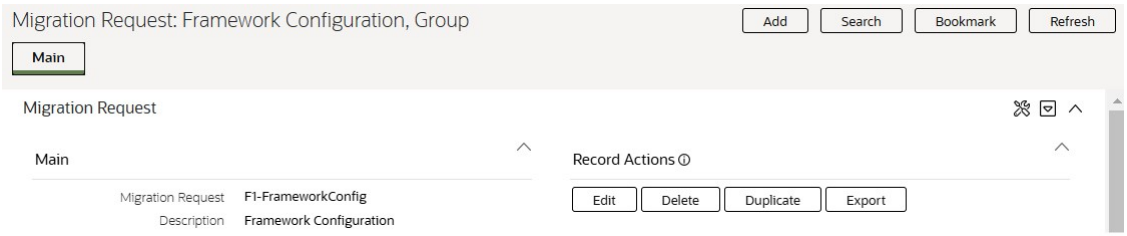


Map Zones

A map zone displays information related to one or more objects. For objects that are maintained using portal-based pages, the standard display of the object is using a map zone. Map zones may be configured to display an explicit map or may dynamically display the map based on the configuration of the record being displayed.

Because the information displayed in a map zone is configured using HTML, virtually any format supported by HTML is possible. The following is an example of a typical map zone used to display information about an object.

Copyright © 2026, Oracle and/or its affiliates.



A tool tip can be used to display additional help information for the details shown in a map zone. The tool tip may appear adjacent to the section header or adjacent to individual elements on a map.

> **Note**
>
> You can customize the help icon or disable map help using the Custom Look and Feel feature configuration.

- Open Map Maintenance Icon

Open Map Maintenance Icon

Click the **Go To Map** icon to switch to the UI map maintenance page. On this page, you can view the HTML configuration for the zone. Note that for map zones that display a business object configured without a map, the elements displayed are automatically rendered. In this case, no icon is visible.

> **Note**
>
> **Debug mode only.** This icon only appears when the system operates in Debug mode.

Tabbed Display Zones

This type of zone combines individual map zones into a single zone, allowing for complex of information to be organized and displayed in different tabs.

Tab labels are built from the zone descriptions of the individual zones and become the zone's title. Selecting a specific tab displays the information managed by the underlying zone. By default, the first tab is selected.

Timeline Zone

A timeline zone is used to show events over time for multiple types of data. This type of zone is helpful for providing a consolidated view of separate events occurring in time that may in fact impact each other. For example, in the billing system, a timeline can show bills, payments and collection processes for the same time period.

Copyright © 2026, Oracle and/or its affiliates.

The topics in this section describe the functionality available in timeline zones.

Timelines Zones Are Configured By Your Implementation Team

The product provides support for creating timeline zones. However, whether timeline zones are configured for your implementation and on which portals they may be visible will depend on your specific configuration. Refer to Configuring Timeline Zones for general information about configuring timeline zones. Refer to your product documentation for details about what type of timeline information is provided out of the box.

You Can Move Through Time

You can click the controls at the top of a timeline zone to change the date-range of the zone's information. To reposition the timeline to a specific date, selected the desired month and year at the top of the zone and click the search arrow.

Controls at the top right of the zone allow you to navigate through time.

- To go back one year, click the double-left arrow.
- To go back one month, click the single-left arrow.
- To go to today, click the middle dot.
- To go forward one month, click the single-right arrow.
- To go forward one year, click the double-right arrow.

Timelines Can Have Many Lines

Depending on how a timeline zone is configured, it will typically have one or more "lines" that show when significant events have occurred. Each line represents a specific type of data. The data may be related to a specific maintenance object, for example, payments may be displayed in one line and bills in another. It may also be possible to create multiple lines for different types of records for the same maintenance object. For example, if your product provides an algorithm to display cases, it may support defining a different entry for each case type.

Each Line Shows Events

Each line on a timeline may contain zero or more events where each event shows the date when the event occurs. For example, the payment line in a timeline has a separate event for every payment received from the customer. Each line's description contains the number of events on the line.

> ⚠ **Caution**
>
> If a line has more events than can fit onto a timeline, the line will show the first "chunk" of events and a message will appear in the "more info area" explaining that some events have been truncated. If this happens and the truncated events are in a later period, you can reposition the timeline's base period to show the truncated events.

Hovering Over An Event Shows the Event Information

When you hover over an event's date icon, a pop-up widow appears showing details of the event. The following information may appear:

- The event date.

Copyright © 2026, Oracle and/or its affiliates.

- The event "hover text". Typically the standard information string is shown for each event.
- The event's common "information string" appears. This information is hypertext to allow for easy access to the transaction on which the object is maintained.
- Additional information may appear. This is dependent on the particular timeline algorithm. Refer to each timeline algorithm's description to find out if additional information appears in the pop-up area under any conditions.
- If the algorithm has configured BPA scripts that can be executed to perform business processes on the object, the BPA script description appears prefixed with a "wizard's hat" icon.

Common Base Portals and Zones

The following sections describe common standards followed by the product for base query and maintenance dialogue.

- All-In-One Portal
- Separate Query and Maintenance Portals

All-In-One Portal

The all-in-one portal is used for low volume objects, where a small number of records are expected.

For example, many control tables use this style of portal. In this portal, rather than separate query and maintenance portals, a single portal allows a user to find a record and maintain the record. A zone listing all existing records is displayed when first entering the portal. A user can perform actions on existing records or add new records from there. More details about the list zone are provided below.

Refer to Maintenance Portal for information about zones commonly found on maintenance portals.

The all-in-one portal will typically provide the ability to add a new record. This may be a button in the Page Action area or a hyperlink defined directly on the list zone's header area.

- List Zone

List Zone

This info zone appears when first entering the portal and displays a row for each record.

The following functions are commonly available in a list zone:

- A broadcast icon is included in each row. Clicking this icon opens other zones that contain more information about the adjacent record.
- An **Edit** icon may be included in each row. Clicking this icon launches the maintenance BPA script to modify the adjacent record.
- A **Duplicate** icon may be included in each row allowing the user to duplicate the adjacent record without having to first display the record.
- A **Delete** icon may be included in each row allowing the user to delete the adjacent record without having to first display the record.
- If the record has a simple BO lifecycle of **Active** and **Inactive**, the list zone may include a button to **Activate** or **Deactivate** the adjacent record.

Copyright © 2026, Oracle and/or its affiliates.

Separate Query and Maintenance Portals

For high volume objects where more sophisticated search criteria is needed to find a particular record, separate portals are provided to search for the record and to maintain it.

The query portal contains a zone to search of the object that you wish to maintain. Additional information about the query portal's search zone is provided below. Clicking an appropriate record in the search results navigates to the object's stand-alone maintenance portal.

The query portal will typically provide the ability to add a new record. This may be a button in the Page Action area or a hyperlink defined directly on the list zone's header area.

- Query Portal Search Zone
- Maintenance Portal

Query Portal Search Zone

The query portal contains one zone used to search for the object.

The zone may be a standard query zone or may be a multi-query zone providing one or more ways to search for your record.

Once the search results are displayed, click on the hypertext for the desired record's information string to navigate to the stand-alone maintenance portal.

Maintenance Portal

A maintenance portal refers to a page that is used to view and maintain a specific record. These types of records are governed by a business object, which defines the user interface layout for both the display of the record and the add or update of the record.

An all-in-one portal is a type of maintenance portal where the maintenance zone is displayed once a record has been broadcast from the list zone. For the separate query and maintenance portal dialogue, the maintenance portal includes only zones relevant for the specific record selected.

Maintenance portals may have multiple tabs to display information relevant for the object. Often, if the object has a Log, the log is on a separate tab. (However, there may be some maintenance portals that include the log zone on the Main tab.)

The stand-alone maintenance portal is typically configured with the ability to navigate back to the query portal using either a **Search** button in the Page Actions area or using a **Go To Search** hyperlink in the main zone's header area. The stand-alone maintenance portal may also include the ability to add a new record using an Add button in the Page Actions area. (If not, a new record can be added by navigating through the menu or by navigating to the query portal where an Add hyperlink may be found.)

The following sections describe zones that are common to portals used to maintain a record. Note that additional zones may exist for specific all-in-one portals or stand-alone maintenance portals based on the particular functionality. Refer to the documentation for each portal to understand the specific zones that are provided.

- Object Display Zone
- Log Zone

Object Display Zone

This is a zone that displays the object being maintained.

Copyright © 2026, Oracle and/or its affiliates.

An object being displayed may or may not allow for actions or updates to be made. Typically there is a Record Actions section in the display area that presents the valid actions. Depending on the functionality, the actions may prompt the users to input values required for that action.

Log Zone

If the maintenance object for the record includes a status, it typically has a log that records important events in its lifecycle.

This log is shown in an info zone on the portal, either on the Main tab or on a separate Log tab. A link appears in the zone's title bar to allow the user to add a new manual log entry.

Process Flow

This portal is used to guide a user through the steps needed to complete a process flow. If the process flow type is configured to keep a record of a completed process flow then this portal may also be used to display the details of such record.

Refer to Understanding Process Flows for more information.

Each type of process flow is assumed to be configured with designated custom menu options that allows you to launch a new process flow of that type as well as review saved off and completed records. When reviewing existing process flow records, you are brought to a query portal with options for searching for a specific record. You may **Resume** a saved off but not yet finalized process flow or review a complete one.

The **Process Flow** zone on the portal's **Main** tab page assists the user in completing the process flow as well as displays the information captured on a completed process flow.

Fixed Pages

There are some pages in the system that are built using a fixed page metaphor. All user interfaces implemented in this way function in a similar manner. This section describes functionality to these types of pages.

> **Note**
>
> Not all pages in the application use this metaphor. The preferred metaphor for pages in the system is to use portal-based pages. For more information see Common Base Portals and Zones.

- Maintenance Pages
- Query Pages

Maintenance Pages

Maintenance pages are used to maintain (add, change, etc.) the information in the system using a fixed page metaphor. This section focuses on functionality common to all maintenance pages using this metaphor.

- The Unique Identifier Area
- Confirmation & Error Messages
- Warning If Unsaved Changes Exist

Copyright © 2026, Oracle and/or its affiliates.

The Unique Identifier Area

Every object in the system must have a unique identifier. For some objects, typically administrative or control table data, the unique identifier is defined by the user creating the record. Other objects, typically master or transaction entities, the unique identifier is assigned by the system after the record is saved.

Where the identifier is displayed depends upon whether the ID is user-defined or system generated.

Pages displaying objects with a user defined identifier (such as To Do Type) typically display the identifier on the left along with a search button. For master and transaction tables, (such as To Do Entry) the unique identifier is typically found in the upper right corner of the screen along with a search button and is labeled as the entity's ID. In this case, if there is an information string associated with the object, it is displayed on the left.

If the object being maintained has an associated context menu, a Context Menu Button is displayed appear.

Confirmation & Error Messages

Whenever a database modification action is requested, the system first validates the data to be put on the database.

First, the system validates the information. If there are errors, the first error encountered is displayed in a window. In addition to the error message, the error window also contains the error number and the identities of the programs that detected the error. If you don't understand an error and you have to call product support, please supply them with this information.

If there are no errors, the system is modified.

Warning If Unsaved Changes Exist

If you make any changes to an object and you don't commit these changes to the database, the following warning appears: "You have unsaved changes that will be lost if you continue. Press OK to continue anyway."

If you want to commit the changes, click **Cancel**. You will then return to the page and can make further changes or press Save. If you want to discard your changes and transfer to the new page, click **OK**.

> ⚠ **Caution**
>
> This warning is not displayed if you are working on a *new* object on a fixed page and you navigate to a different page. Information about new objects is preserved in memory in anticipation of your return to the page. For example, if you are in the middle of adding a new user and you want to quickly reference information on a different page, you will not receive the above warning because your information will be saved. If you use the back button to return to the user page, all of the information you were adding will be present and you can enter the rest of the new user's data.

Copyright © 2026, Oracle and/or its affiliates.

Query Pages

Query pages are used to display record sets in the system. Most query pages function in a similar manner. This section focuses on functionality common to all of the system's query pages.

- The Search Area

The Search Area

The upper right corner of the screen is almost always the identifier of the object associated with the search. In Oracle Utilities Customer Care and Billing for example, the unique identifier on the Account Financial History page is the Account ID.

To the right of the unique identifier is a Search button. This button is used to search for a different object than the one whose data is being displayed in the page.

If the object being displayed has an associated context menu, a Context Menu Button is displayed.

UI View Settings

The system is delivered with a standard view of the user interface. A "compact" view of the user interface is also available. This option reduces the amount of white space for the rendered pages. This option is available system-wide or an individual user can override the system-wide setting.

- You can set change the UI view system-wide for all users via the Custom Look and Feel Options.
- Users can set their preferred value in the main tab of the user preferences.

User Preferences

The Preferences (or My Preferences) transaction and the User Maintenance transactions are the same except that with My Preferences, only the following items can be modified:

- Main Tab
  - Language
  - Display Profile
  - Time Zone
  - Email Address
  - UI View
  - Sidebar Width
  - Sidebar Location
  - Sidebar State
  - Home Page
  - To Do Summary Age Bar Parameters
- All the information on the following tabs:
  - Bookmarks

Copyright © 2026, Oracle and/or its affiliates.

  - Portal Preferences
  - Favorite Links
  - Favorite Scripts

- User - Main
- User - To Do Roles
- User - Access Security
- User - Bookmarks
- User - Portal Preferences
- User - Favorite Links
- User - Favorite Scripts
- User - Characteristics
- User - Miscellaneous

User - Main

This page lets you view or define user information and access settings.

To open the page, use one of the following navigation options:

- The **Admin**, and then **Security**, and then **User**, and then **Search** menu item.
- The **Menu**, and then **My Preferences** menu item.
- The **User Menu**, and then **Preferences** menu item in the application toolbar.

Note that some fields are only modifiable when the page is opened from the from the **Admin** menu.

Description of Page

Basic user identification information is viewed or set in the **First Name**, **Last Name**, **User ID**, and **Login ID** fields, all of which are modifiable only when the page is opened from the **Admin** menu. User and logon IDs must be unique. The Login ID is the ID that the user uses to log in to the system. The User ID is the ID that is used throughout the application when users are assigned to tasks or when the user ID is stamped on records when logging add or update actions.

Language defines the language in which the user's screen prompts and messages appear.

Set **User Enable** to allow the user to access the system.

> **Note**
>
> This field, available only when the page is accessed through the **Admin** menu, provides special field-level security. A user can change the field only if the **Enable/Disable** access mode is set.

Display Profile defines the display options for the user.

The following fields are related to defining a template user along with some configuration available for using a template user. This fields are modifiable only when the page is accessed through the **Admin** menu. Refer to template user for more information about these fields.

Copyright © 2026, Oracle and/or its affiliates.

- Set **User Type** to **Template User** if the user is going to serve as a template user.
- If a user's portal preferences are to be inherited from a "template user", use the **Portals Profile User** to define the user ID of the template user.
- If a user's favorite links are to be inherited from a "template user", use the **Favorites Profile User** to define the user ID of the template user.

The **Time Zone** associated with the user may be defined. Implementations may use this information for features and functions where the user's time zone impacts display of data. Refer to Designing Time Zones for more information about the use of time zones.

Email Address captures the user's email address.

UI View allows the user to set a preference for the UI View Settings that is different from the system wide definition.

Sidebar Width defines the number of pixels to use for the user's sidebar area. A setting of 250 is recommended as a minimum. Depending on the width of a user's monitor, a higher setting (such as 250) will display more information without compromising the space on the main area of the application. Setting the value to 0 will suppress the sidebar. The width setting is updated in background to retain the value based on dynamic sizing.

Sidebar Location governs whether the sidebar appears before or after the main page area. If no value is specified, the default position of the sidebar is after the main page area.

Sidebar State indicates whether the sidebar is open or closed. The state is updated in background to retain the value based on dynamic opening or closing of the sidebar.

Home Page defines the page that appears when the user starts a session or clicks the Home link. Access to the home page is also available from the menu.

The To Do entry fields control how To Do entries are aged in respect of the "age bars" that appear on To Do Summary, To Do Supervisor Summary and the To Do Summary Sidebar Zone.

- Use **Lower Age Limit for Yellow Bar** to define the number of days old a To Do entry must be to be considered "yellow" in the age bar. To Do entries below this limit are considered "green" in the age bar.
- Use **Upper Age Limit for Yellow Bar** is the high limit for the "yellow" bar. To Do entries older than this are considered "red" in the age bar.

The **User Group** collection defines the group(s) with which the user is associated and the date (if any) when the user's association expires. This feature is modifiable only when the page is accessed through the **Admin** menu. For more information, see The Big Picture Of Application Security. Assignments of users in a user group can also be established or maintained through the User Group page.

> **Note**
>
> The ability to maintain the user groups on this page requires special field-level security. A user can change the field only if the **System Administration** access mode is set.

If your implementation has configured Object Erasure for the User maintenance object, the **Object Erasure Schedule** sidebar zone is visible displaying the erasure status for this user record, if applicable. Refer to Erasing User Information by Obfuscation for more information on object erasure for User records.

Copyright © 2026, Oracle and/or its affiliates.

User - To Do Roles

Open the User page and then navigate to the **To Do Roles** tab to define the To Do roles assigned to the user.

The information on this page cannot be modified if accessed through **My Preferences**.

> **NOT_SUPPORTED**
>
> A user's To Do roles control the types of **To Do** entries the user can access. For additional information on To Do roles, see To Do Entries Reference A Role.

Description of Page

The grid contains the user's To Do roles. To modify a To Do role linked to the user, simply move to a field and change its value. To remove a To Do role, click the **-** (minus) button. To add a new To Do role, click the **+** (plus) button and enter the role.

Where Used

Every **To Do** entry references a role. A To Do role has one or more users (and a user may belong to many To Do roles). Users who are part of the To Do role assigned to a To Do entry may work on the To Do entry. For more information on this feature, see The Big Picture Of To Do Lists.

User - Access Security

Open the User page and then navigate to the **Access Security** tab to define a user's security rights. You cannot modify this page if you accessed it through **My Preferences**.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Application Security for more information about data access roles and access groups.

Description of Page

Use the **Default Access Group** to define the access group that is defaulted on new records added by this user that are subject to row security.

The scroll area contains the **Data Access Roles** to which this user belongs. A user's data access roles play a part in determining the accounts whose data they can access.

To add additional data access roles to this user, click the **+** button and specify the following:

Enter the **Data Access Role**. Keep in mind that when you add a **Data Access Role** to a **User**, you are granting this user access to all of the accounts linked to the data access role's access groups.

Use **Expiration Date** to define when the user's membership in this data access role expires.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can also use Data Access Role - Main to maintain a data access role's users.

The tree highlights the **Access Groups** to which the above **Data Access Roles** provide access.

User - Bookmarks

This page is used to manage a user's bookmarks that are created when a user clicks the Bookmark button and saves a bookmark. From this page the user can delete a bookmark, change the name of the bookmark or change the sort sequence.

Open the User page and then navigate to the **Bookmarks** tab to modify or delete a bookmark.

Description of Page

Sequence is the sort sequence associated with the bookmark. Bookmarks with the same sequence are then sorted alphabetically.

Name is the user-defined name assigned to the bookmark. This is the text that is visible in the Bookmarks sidebar zone.

> **Note**
>
> **Favorite Links vs. Bookmarks.** Favorite links are configured through the user page and define pages that a user can navigate to. They do not define context (although some pages may use information currently in global context when navigating). Some users may not be allowed to configure their own favorite links and would see the favorite links associated with a profile user. Bookmarks are created using a bookmark button available on each page. The bookmark saves navigation information, context information and also captures some information about the state of a page. Bookmarks are not governed by any user profile configuration.

User - Portal Preferences

The base product contains several portals that allow users to customize them via portal preferences. Portal preferences allow users to control:

- Which zones appear on the portal
- The order in which the zones appear
- Whether the zones should be "collapsed" (i.e., minimized) when the portal opens.

> **Note**
>
> **You may not be able to change your portal preferences.** If a note appears immediately before the list of portals, a system administrator has configured your user ID to reference a **Portals Profile User** (this is defined on the **Main** tab). Preferences set in this way cannot be modified. System administrators do this in order to enforce a common look-and-feel throughout the user community.

Copyright © 2026, Oracle and/or its affiliates.

Open the User page and then navigate to the **Portal Preferences** tab to modify a user's preferences.

Description of Page

The accordion contains a row for every portal configured to show on user preferences and to which you have access. To change how a portal's zones appear, expand the respective row and change the elements accordingly. The remainder of this section describes how you can configure how a portal's zones appear.

Zone lists all of the zones in the portal.

Place a check under **Display** if the zone should be displayed on the portal.

Place a check under **Initially Collapsed** if the zone should be minimized when the portal is initially opened. Setting zones as initially collapsed is a good idea to save space and to help pages to load more quickly. Refer to Zones May Appear Collapsed When A Page Opens for more information.

> ⚠ **Caution**
>
> Recommendation. It is recommended that you set your preferences to collapse zones that you don't use often. This is because the system doesn't perform the processing necessary to build collapsed zones until you ask to expand a zone. Opening a page with zones collapsed accelerates the response times of portal.

Use **Sequence** to control the position of the zone in respect of the other zones. This field is disabled if **Display** is not checked. Duplicate **Sequence** numbers are allowed.

For zones displaying data that is changing often, use **Refresh Seconds** to define in seconds how often the zone is refreshed. The minimum valid value is 15. The maximum valid value is 3600 (1 hour). A value of 0 indicates no automatic refresh.

Security Access indicates whether the user has access rights to a zone. Refer to Granting Access To Zones for more information.

User - Favorite Links

This page is used to define a user's favorite transactions and URLs. Each such link appears as an entry in the Favorite Links Zone in the Sidebar Portal, as well as through the Menu.

> ⓘ **Note**
>
> **A hot key can be used to execute a favorite link.** Rather than clicking a button in the favorite links zone, you can invoke your first nine favorite links by pressing `Ctrl` + a number (where the number corresponds with the relative position of the transaction in your list of favorites). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Open the User page and then navigate to the **Favorite Links** tab to modify a user's favorite links.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You may not be able to change your favorite links.** If a **Note** appears beneath your **User ID**, your user id has been setup to reference a specific **Favorites Profile User** as defined on the **Main** tab. If a system administrator has setup your user ID in this way, your preferences may not be changed. System administrators may apply this feature in order to enforce a common look-and-feel throughout the user community.

Each row references a Navigation Option. The navigation options supplied with the base package correspond with the system's menu items. This means you can setup a favorite link to open any transaction in the system (in either **add** or **update** mode).

> **Note**
>
> **Favorite links can point to other places.** The navigation options supplied with the base package always open a transaction's **Main** tab. If you find yourself continually accessing a different tab page, you can have your implementation team create a new navigation option that references the desired tab page. Once this navigation option is set up, you can reference it as one of your favorite links. In addition, it's also possible for your implementation team to setup navigation options that open a URL outside of the system. For example, your implementation team could setup a navigation option to a credit rating company's website. Once this navigation option is set up, you can reference it as one of your favorite links.

Use a **Sequence** to control the order in which the navigation option appears in the Favorite Links Zone.

Security Access indicates if you have been given security rights to the respective transaction. Refer to The Big Picture Of Application Security for more information.

> **Note**
>
> **Favorite Links vs. Bookmarks.** Favorite links are configured through the user page and define pages that a user can navigate to. They do not define context (although some pages may use information currently in global context when navigating). Some users may not be allowed to configure their own favorite links and would see the favorite links associated with a profile user. Bookmarks are created using a bookmark button available on each page. The bookmark saves navigation information, context information and also captures some information about the state of a page. Bookmarks are not governed by any user profile configuration.

User - Favorite Scripts

This page is used to define a user's favorite scripts. Each script is displayed in the Favorite Scripts Zone in the Sidebar Portal, as well as through the Menu. Clicking the entry in the favorites can be used to initiate a script.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **A hot key can be used to execute a favorite script.** Rather than selecting a script from the business process assistant search, you can invoke your first nine favorite scripts by pressing `Ctrl + Shift +` a number (where the number corresponds with the relative position of the script in your list of favorites). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Open the User page and then navigate to the **Favorite Scripts** tab to modify a user's Favorite Scripts.

Description of Page

> **Note**
>
> **You may not be able to change your favorite scripts.** If a **Note** is displayed on top of the grid, your user id has been setup to reference a specific **Favorites Profile User** as defined on the **Main** tab. If a system administrator has setup your user ID in this way, your preferences may not be changed. System administrators may apply this feature in order to enforce a common look-and-feel throughout the user community.

Each row in the grid contains one of your favorite **Scripts**.

Use the **Sort Sequence** to control the order in which the scripts appear in the zone.

Security Access indicates whether you have access to the script's application service. Refer to The Big Picture Of Application Security for more information.

User - Characteristics

Select **Admin**, and then **Security**, and then **User**, and then **Search** and navigate to the **Characteristics** tab to define characteristic values for the user. You cannot modify this page if you accessed it through My Preferences.

Description of Page

Use the characteristics grid to capture additional information about the user.

You can only choose **Characteristic Types** defined as permissible for the user record. Refer to Setting Up Characteristic Types & Their Values for more information.

Enter a valid **Characteristic Value** for the selected characteristic type. You may enter more than one characteristic row for the same characteristic type, each associated with a unique **Sequence** number. If not specified, the system defaults it to the next sequence number for the characteristic type.

User - Miscellaneous

Select **Admin**, and then **Security**, and then **User**, and then **Search** and navigate to the **Miscellaneous** tab to view additional information associated with a user.

User Id, First Name and Last Name are displayed. Depending on your implementation, additional zones may appear on the User Portal

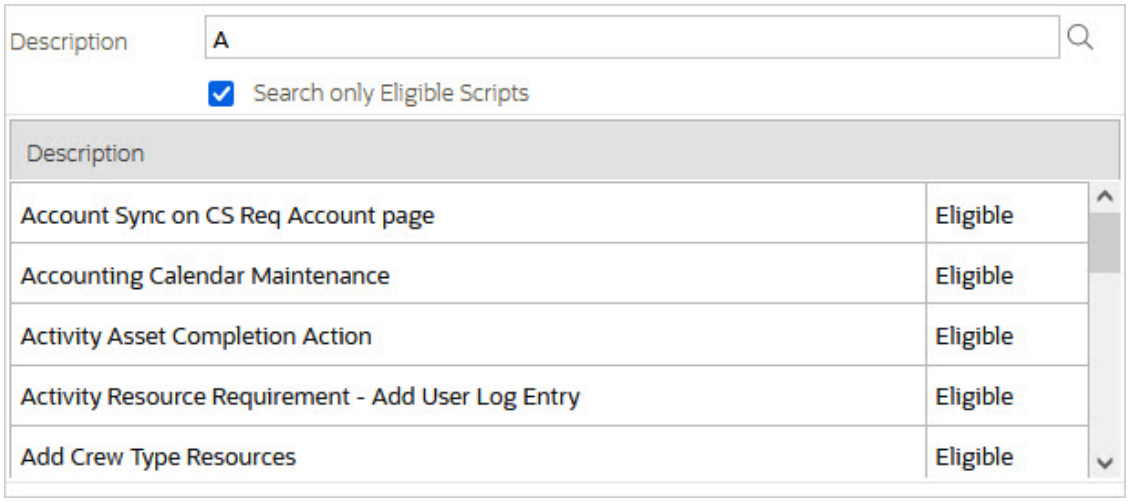
Copyright © 2026, Oracle and/or its affiliates.

If additional zones appear, use the embedded help for information about the fields managed on the zone.

Initiating Scripts

Your implementation team can set up business process assistant (BPA) scripts to walk you through business processes. Once scripts have been set up, you can initiate a script using any of the following methods:

- You can define a list of your "favorite scripts" on your user preferences. You can then invoke any of the scripts from the Favorite Scripts zone on the sidebar. You can also invoke any of your first 9 favorites by pressing `Ctrl + Shift +` a number (where the number corresponds with the relative position of the script in your list of favorites). For example, if you press `Ctrl+Shift+2` and you've setup your "favorite scripts" to be **Start Autopay** and **Stop sending marketing information**; the **Stop receiving marketing information** script is invoked. Note that the numeric keypad on your keyboard cannot be used to implement this function.
- You can configure a menu item to launch a script by referencing a navigation option that is a **script** navigation option type.
- The system creates To Do entries to highlight tasks that require attention (e.g., bills in errors, accounts without bill cycles, etc.). Users can complete many of these tasks without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill. Refer to Executing A Script When A To Do Is Launched for more information.
- You can indicate that a script should be launched upon entering the system. This functionality enables an external system to launch the system and immediately launch a script.
- The following hot keys cause an interactive menu to be displayed:
  - `Ctrl+Alt+Q`
  - `Ctrl+Shift+S`



The menu allows you to search for a script using any word in its description. Note, if you've turned on **Search only Eligible Scripts**, your search is limited to eligible scripts. Note, the system displays an indication if each script in the grid is **Eligible** or **Ineligible** in this section.

Copyright © 2026, Oracle and/or its affiliates.

You can initiate a script from this area by clicking on it.

When a script is initiated, this menu closes and the object display area shifts down to make room for the script area.

> **Note**
>
> **Securing scripts.** Only scripts the user is allowed to execute are displayed. Refer to Securing Script Execution for more information.

> **Note**
>
> **Customize the script area.** Using the Height step type, an individual script may be configured to use more or less of the script area than the standard script area. It is also possible to configure a script to hide the script area altogether.

The information in the script area takes you through the steps in the selected business process (note, your implementation team controls the information that appears in the script area).

The following points highlight useful information common to all scripts:

- **Return focus to script area.** Often during a script the focus moves from the script area to another section of the page. For example, you might be required to fill in data for a new record being added. When you are ready to return to the script area, rather than using the mouse to return to the area, you can press `Ctrl+Shift+F`. The focus returns you to either the first available input field or to a "default" button if no input field is available.
- **The Continue button.** Many steps in a script ask you to click the **Continue** button when you're ready to restart a script. For example, a step may ask you to confirm a customer's automatic payment information and then click **Continue** when you're ready for the script to restart. Rather than using the mouse to click this button, you can press `Ctrl+Shift+C`.
- **Selected button changes color.** When you click a button in the script area, the button selected changes color so that you can easily see which option you selected.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Scripts for more information about scripts.

> **Note**
>
> You can abort a script at any time by clicking the red X button in the upper right corner of the script area (or by pressing `Ctrl+Shift+X`).

Shortcut Key Summary

The following table summarizes the keyboard shortcuts (hot keys) available in the system:

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> It is possible that your specific product supports additional shortcut keys. Refer to your product documentation for more information.

> **Note**
>
> It is possible that some of the following system accelerator keys may duplicate shortcuts, or hot keys, that are predefined within your browser. If this is the case, the browser's functionality commonly takes precedence.

> **Note**
>
> It is possible that your implementation uses one of the accelerator keys listed below for a different purpose. For example, in some languages, a shortcut key may be used for a special character in the alphabet not included on all keyboards. It is possible to turn off any of the Alt+letter shortcuts and Ctrl+Alt+letter shortcuts listed below using a properties file entry. The property is `ouaf.shortcut.ignore.altKeys=` where one or more letters may be entered separated by a comma. When setting this property, the system automatically enables a 'potential alternate key' for certain functions (see the table below). Refer to the *Server Administration Guide* for more information about the property.

Accelerator Key	Potential Alternate Key (see Note above)	Function
`Alt+B`		Returns to the previous page on which you were working. This is the equivalent of clicking the Back arrow.
`Ctrl+Alt+B`		Launches the Bookmark dialog. This is the equivalent of clicking the Bookmark button.
`Alt+C,Ctrl+Alt+C, Alt+Shift+C`		Clears the currently displayed object from the page. This is the equivalent of clicking the Clear button.
`Ctrl+Shift+C`		If the current step in the Script area has a **Continue** button, this shortcut is equivalent to clicking **Continue**.
`Ctrl+Alt+D, Alt+D`		Creates a duplicate of the object being displayed. Equivalent to clicking the Duplicate button.
`Ctrl+Alt+F`		Positions the user's cursor to the Search input field.
`Ctrl+Shift+F`		If the Script area is open and you are working on a different area of the page, this key returns focus to the script area. Focus returns to either the first available input field or to a "default" button if no input field is available.
`Alt+G`		Returns to the most recent page that was displayed when the Back arrow was clicked. Equivalent to clicking the Forward arrow.
`Alt+J`		Minimizes the sidebar. It also expands the sidebar, if it was minimized.

Copyright © 2026, Oracle and/or its affiliates.

Accelerator Key	Potential Alternate Key (see Note above)	Function
Alt+K		Transfers you to To Do Entry - Main for the current item in the To Do list. Pressing Alt+K while on this page takes you to the first To Do entry on the list.
Ctrl+Alt+L		Logs the user out of the system. This is equivalent to clicking the Logout menu item.
Ctrl+Alt+M		Opens the Menu.
Alt+N	Alt+Shift+N	Displays the next entry in the search results. Equivalent to clicking Next Item.
Alt+O	Alt+Shift+O	Navigates to the user's home page. Equivalent to clicking the Home icon.
Ctrl+Alt+P		Opens your User Preferences page. Equivalent to clicking the Preferences menu item.
Alt+P		Displays the previous entry in the search results. Equivalent to clicking Previous Item.
Ctrl+Alt+Q		Opens the business process assistant, where you can select a script that walks you through a business process.
Alt+RCtrl+Alt+R		Refreshes the page with the last saved version of the object being displayed in the page. Equivalent to clicking the Refresh button.
Alt+S	Alt+Shift+S	Saves any changes that you've made on the database. Equivalent to clicking the Save button.
Ctrl+Shift+S		Opens the business process assistant, where you can select a script that walks you through a business process.
Alt+X	Alt+Shift+X	Opens the To Do Summary.
Ctrl+Shift+X, Ctrl+Shift+Z		If the Script area is open, equivalent to clicking the **Close** button in the script area. If a pop-up window is open, this closes the window.
Alt+Y		Displays the previous item in the To Do list. Equivalent to clicking the **Previous** button in the Current To Do zone.
Alt+Z	Alt+Shift+Z	Displays the next item in the To Do list. Equivalent to clicking the **Next** button in the Current To Do zone.
Alt+Insert		Inserts a new row into an editable grid if the insertion point is in the grid. Equivalent to clicking the **Insert** button.
Alt+Delete		Removes a row from an editable grid if the insertion point is in the grid. Equivalent to clicking the **Delete** button.
Alt+[		Navigates to the next main page component area of the application. Refer to Page Components for the list of the main page components. Using Shift+Alt+[reverses the sequence of navigation.

Copyright © 2026, Oracle and/or its affiliates.

Accelerator Key	Potential Alternate Key (see Note above)	Function
`Alt+]`		Navigates to the next zone. When you are on a portal with zones, this key brings you to the next zone in the portal. If you are on the last zone in the portal, the key brings you to the first zone on the sidebar, if visible. Keying through all the zones on the sidebar brings you to the first zone on the portal displayed in the main area. If the page displayed is a fixed page rather than a portal, this key will only navigate within the sidebar. Using Shift+Alt+] brings you to the previous zone.
`Alt+F1`		Invokes online Help for the application. Equivalent to clicking the Help entry in the Help menu.
`Alt+1` ...`Alt+9`, `Alt+0`		These keys are only applicable when a page / portal with multiple tabs is open. The shortcuts open the tab page whose relative position corresponds with the shortcut number. For example, if you press Alt+3 on a page with multiple tabs, the third tab is displayed. Alt+0 navigates to the 10th tab. If there are more than ten tabs on a page, you must use the mouse or F2/Shift+F2 to display additional tabs. Note that you cannot use the numeric keypad for these shortcuts.
`Ctrl+1` ...`Ctrl+9`		Invokes the favorite link whose relative position corresponds with the shortcut number. For example, if you press `Ctrl+3`, the transaction for your third "favorite link" is invoked. Note that you cannot use the numeric keypad for these shortcuts.
`Ctrl+Shift+1` .. `Ctrl+Shift+9`		Invokes the favorite script whose relative position corresponds with the shortcut number. For example, pressing Ctrl+Shift+2 selects the second of your "favorite scripts". The numeric keypad cannot be used for these shortcuts.
`Alt+left arrow`		Populates a scroll area with the previous entry. Equivalent to clicking the scroll control .
`Alt+right arrow`		Populates a scroll area with the next entry. Equivalent to clicking the scroll control .
`Enter` when in a field with an adjacent Search button		Invokes the search. Equivalent to clicking the Search Button.
`Enter` when a row in a search result grid is highlighted		Selects the item and closes the search. Note that after selecting an item from the search results, you can press Alt+N or Alt+P to scroll up or down through other items in the result grid without returning to the search results.
`Enter` when a menu item is highlighted		Selects the menu item.

Copyright © 2026, Oracle and/or its affiliates.

Accelerator Key	Potential Alternate Key (see Note above)	Function
`Enter` when the insertion point is within a tree node		Equivalent to clicking on the item in the tree: • If the insertion point is on a node image, the node is either expanded or collapsed, dependant on its current state. • If the insertion point is on a context menu, the context menu is opened and normal menu processing applies. • If the insertion point is on a "link" element, normal navigation processing occurs.
`Tab` when a row in a search result is highlighted		Highlights the next row.
`Shift + Tab` when a row in a search result is highlighted		Highlights the previous row.
`Space` when the insertion point is on a button		Equivalent to pressing the button.
`Space` when the insertion point is in a date/time field		Opens the date/time selection window.
`Space` when the insertion point is in a check box		Toggles the check box between checked and unchecked.
`Page Up` while in a list grid		Scrolls the contents of the list grid up approximately one page.
`Page Down` while in a list grid		Scrolls the contents of the list grid down approximately one page.
`Up or Down Arrow` when in a dropdown list		Moves to the next or previous value in the list. You can also press a letter key to locate the next value starting with that letter.
`Up or Down Arrow` keys when in search results		Highlights the next/previous row in search results.
`Up, Down, Left or Right Arrow` when using a menu		Highlights other menu items.
`Esc` when using a menu		Closes the menu.
`Alt+F2`		Opens the next tab page when viewing a page / portal with multiple tabs.
`Shift+Alt+F2`		Opens the previous tab page when viewing a page / portal with multiple tabs.

Copyright © 2026, Oracle and/or its affiliates.

Customer Information

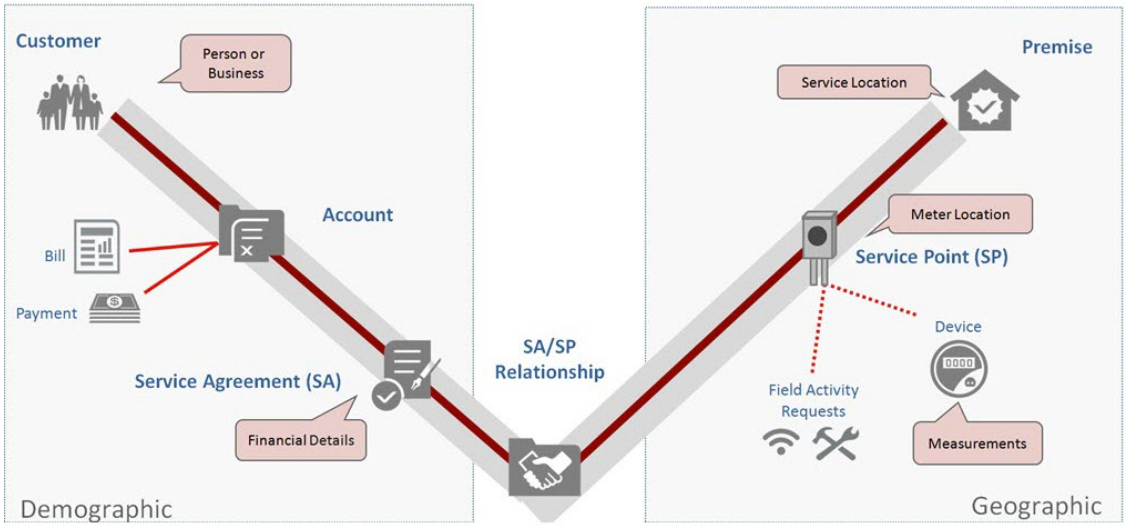
We use the term **customer information** to reference the demographic, geographic, and financial objects that form the core of your CIS system. In this section, we describe how to maintain these objects.

- Understanding The V
- Navigating The V
- Maintaining The V
- Sidebar Portal
- Setting Up Bill Print Groups
- Landlord Reversion
- Using The Premise Replicator
- Start/Stop
- Customer Contacts
- Printing Letters
- Maintaining Declarations
- Service Tasks
- Registration Points
- Consumer Contracts
- Customer Service Requests
- Customer Profiles

Understanding The V

The "V" is the shape of a diagram we use to illustrate the objects that form the core of the system: **Person**, **Account**, **Premise**, **Service Agreement**, and **Service Point**. These objects hold demographic, geographic, and financial information about your company's customers and properties. The following diagram illustrates the objects in the "V", as well as important objects related to the "V" objects (**bill**, **payment**, **meter**, **field activity**, **meter read**).

Copyright © 2026, Oracle and/or its affiliates.



You must intuitively understand the concepts embodied in the "V" before you can perform the business processes in the system. In this section, we provide an overview of these objects. In later sections, we describe how you maintain this information.

- Persons
- Accounts
- Service Agreements (SAs)
- Premises
- Service Points (SPs)
- Putting It All Together
- Sample V For A Residential Customer
- Sample V For A Commercial Customer With Multiple Premises
- How To Set Up Customer Hierarchies

Persons

A person exists for every individual or business with which your company has contact. Besides customers, persons exist for landlords, contractors, accountants at corporate customers, third party guarantors, energy distributors, collection agencies, etc.

On a person is maintained demographic information like Name, Mailing Address, Phone Numbers, Email Address, and Life Support Details.

Most persons are linked to at least one account because, without an account, the person cannot receive a bill. There are rare examples of persons without accounts; for example, collection agencies that don't receive bills, and parent companies that receive statements (as opposed to bills).

NOT_SUPPORTED

Refer to Maintaining Persons for more information about persons.

Copyright © 2026, Oracle and/or its affiliates.

Accounts

Accounts are the entities for which bills are created. You must create at least one account for every customer. The account contains information that controls when bills are created and how they are formatted.

Every account must reference at least one person because the person contains the customer's demographic information (e.g., names, phone numbers, forms of ID). We refer to this individual as the "main" person linked to the account. In addition to the "main" person, an account may reference other types of persons, e.g., the billing contact, the third party guarantor of the account's debt, persons who receive copies of bills, etc.

Most accounts are linked to at least one service agreement because, without a service agreement, there is nothing on which a bill can be based. An account without a service agreement may exist; you just won't be able to do much in the system with such an account.

NOT_SUPPORTED

Refer to Maintaining Accounts for more information about accounts.

Service Agreements (SAs)

Think of a service agreement as a contract between your company and a customer. The service agreement contains the terms and conditions controlling how the system calculates charges for the specific service supplied to the customer.

Every account should have at least one service agreement (otherwise, the account has no financial obligations with your company). There is no limit to the number of service agreements that may be linked to an account. However, most residential customers have X service agreements where X is the number of services sold by your company. For example, if your company sells both electricity and gas, most accounts will have two service agreements.

A separate service agreement is required for every service supplied to every account. Why? Because a variety of fields may differ for each service agreement:

- Obviously, the price of the service is different.
- The credit and collection criteria may be different.
- The general ledger account to which the resultant revenue and/or receivable are linked is different.
- The sales tax could be different for each good / service / location.
- The service point measuring consumption will frequently be different.
- The list goes on...

Service agreements that bill for premise-based services are linked to one or more premises via their service points. When you link a service point to a service agreement, you are telling the system that the service agreement's account will pay for the service point's service.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Maintaining Service Agreements for more information.

Premises

A premise is created for every location to which your company supplies service. On a premise is maintained geographic information like:

- The address.
- Characteristics that determine tax jurisdictions.
- Descriptions of unusual situations associated with a property.
- The list goes on...

Most premises have at least one service point because, without a service point, there are no services supplied to the premise. A premise without service points may exist; you just won't be able to do much in the system with such a premise.

NOT_SUPPORTED

Refer to Maintaining Premises for more information about premises.

Service Points (SPs)

Think of a service point as a specific location at a premise where your company supplies service. A service point contains information describing the type of service and how it is measured.

A premise should have at least one service point (otherwise, there are no services supplied to the premise). However, it is permissible for a new premise to exist without service points until you are ready to physically start service at the premise.

A premise may have many service points. However, most residential premises have X service points where X is the number of services sold by your company. For example, if your company sells both electricity and gas, most of your premises will have two service points.

Separate service points are required for every service supplied to a premise. Why? Because a variety of fields may differ for each service point:

- Basic information about the Service Point, including time zone, market, parent Service Point (if applicable), and status.
- Information related to field work that is performed at the current Service Point, including applicable warnings or instructions.
- The measurement cycle, route, and route sequence for the Service Point.
- Basic characteristics about the service; for example, pipe size, take point, and amperes.

Copyright © 2026, Oracle and/or its affiliates.

When you link a service point to a service agreement, you are telling the system that the service agreement's account will pay for the service point's service.

The following topics describe the relationship between a service point and the physical devices that may be installed at the service point.

- Metered versus Item-Based versus Non-Badged Service Points
- Service Points With Multiple Items

Metered versus Item-Based versus Non-Badged Service Points

There are three major categories of service points:

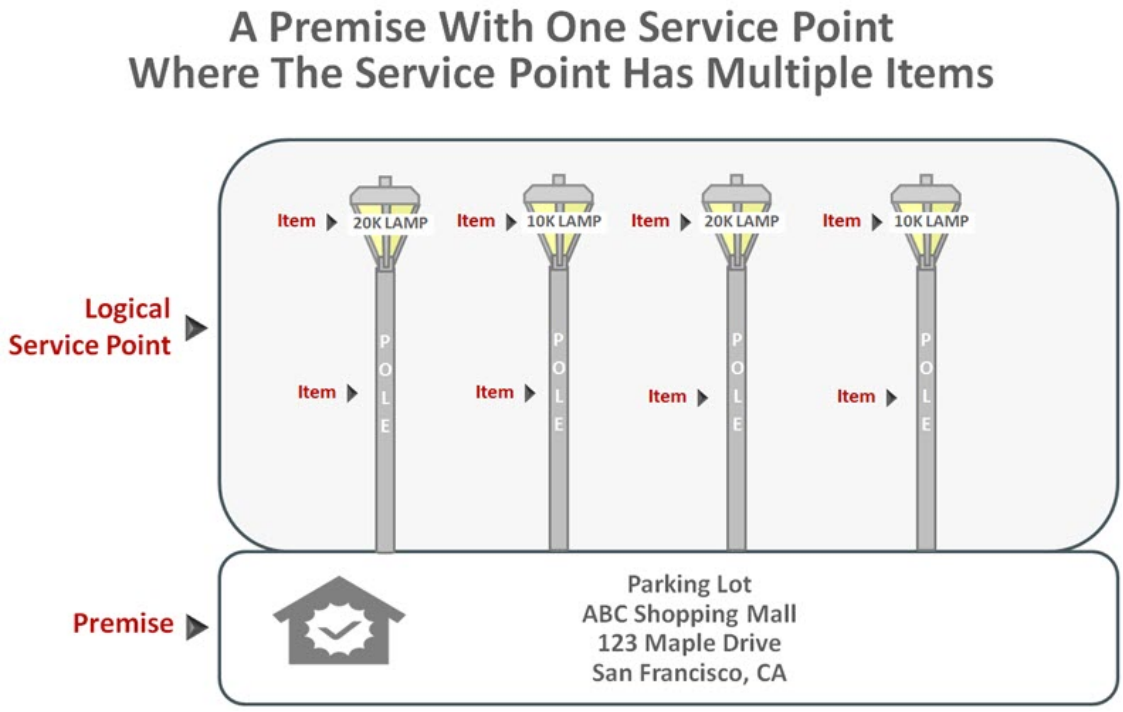
- Those where consumption is measured (e.g., electricity, gas, water) by a meter. We refer to these types of service points as **metered**. A metered service point may have zero or one meter installed at any instant in time. Over time, a metered service point may have many meters installed and removed.
- Those where consumption is not measured, but where there exists some type of badged (i.e., uniquely identified) item that impacts billing and dispatching (e.g., streetlights, a security camera). We refer to these types of service points as **item** -*based*. An item-based service point may have zero or one badged item installed at any instant in time. Over time, an item-based service point may have many badged items installed and removed.
- Those where one or more non-badged items (i.e., items without a unique identity) that have some impact on billing or dispatching exist. We refer to these types of service points as **unbadged**. At a single instant in time, a non-badged service point may have many unbadged items installed at it. Refer to Service Points With Multiple Items for an illustration of such a service point.

> **Note**
>
> **Multiple unbadged items can also be defined on metered and item-based service points.** Under some billing scenarios, you may define non-badged items on a metered or item-based service point. For example, assume you have an item-based service point that exists to support lamp service. Also assume that the rate for lamp service is based on the type of lamp and the number of feet of electric wire between the pole and the power source. To define the number of feet of wire, you could create a non-badged item type called WIREFEET and indicate the number of feet on the item-based service point. On this service point, you can also keep track of the specific lamp that's installed.

Service Points With Multiple Items

An example will help explain a single service point with multiple non-badged items. Assume you have a parking lot with 4 lights, none of which are badged.

Copyright © 2026, Oracle and/or its affiliates.



You can set up the above example with one premise with a single non-badged service point. On the service point, you create one item for each type of item in the parking lot:

- 20K lumen lamp
- 10K lumen lamp
- pole

For each type of item, you indicate the quantity that exist in the parking lot:

- 20K lumen lamp - 2
- 10K lumen lamp - 2
- pole - 4

Putting It All Together

To help you understand the significance of the "V", consider a few scenarios:

- When a new residential customer starts service, you create both person and account objects for the new customer. In addition, you also create service agreements and link them to the appropriate service points. You won't have to create a new premise and service points because the customer is probably moving into an existing house.
- If the residential customer moves within your service territory, you don't have to set up new person or account objects. You also don't have to set up new premise and service point objects if the customer is moving into an existing house. You only have to create new service agreements and link them to the appropriate service points (and the system does most of the service agreement set up processes for you).
- A business with a single location uses the same "V" objects as does a residential customer. If the business expands and starts using services at other locations, you simply

Copyright © 2026, Oracle and/or its affiliates.

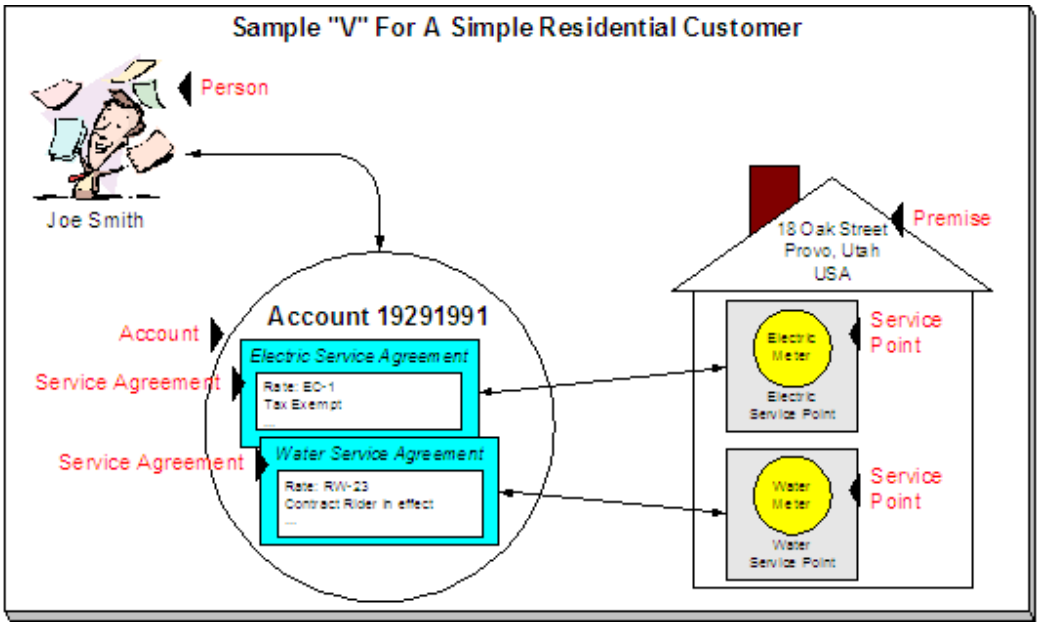
link new service agreements to the existing account. You don't have to redefine person or account information.

> **Note**
>
> **Bottom line.** A person and an account are needed for every customer. After these exist, future service requests simply involve stopping existing service agreements and starting new ones. You typically won't have to add or modify premise or service point information unless new locations are built.

> **NOT_SUPPORTED**
>
> For more information, refer to Maintaining The "V", Navigating The "V" Using Control Central, The Big Picture Of Starting Service, and The Big Picture Of Stopping Service.

Sample V For A Residential Customer

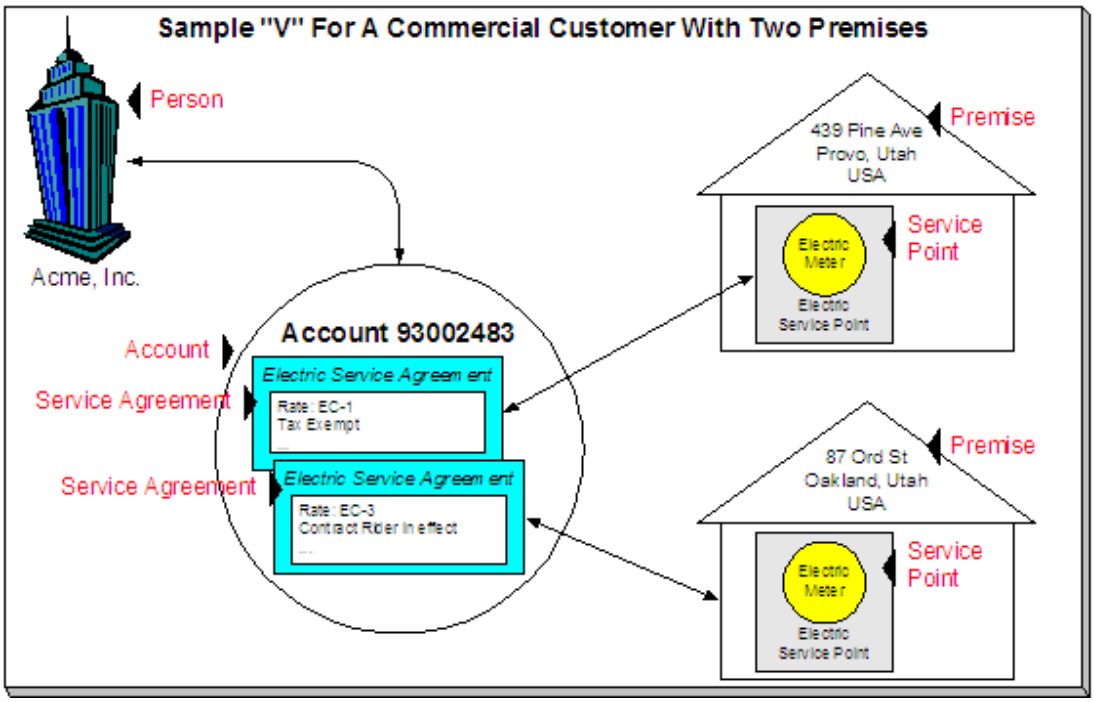
The following illustration shows a simple residential customer's "V" objects.



Sample V For A Commercial Customer With Multiple Premises

The following illustration shows the "V" objects for a commercial customer who pays for service at two premises.

Copyright © 2026, Oracle and/or its affiliates.



> **Note**
>
> The above example shows a separate service agreement per premise. If your rates allow, you could link both premises to a single service agreement.

How To Set Up Customer Hierarchies

Consider the following customer hierarchy:



The above hierarchy shows a parent company (Michelin Inc) with two subsidiaries. Each subsidiary, in turn, could have their subsidiaries, and these subsidiaries can have their own subsidiaries ...

You must set up a person for every individual and business in a hierarchy. After the basic demographic information is defined, you define each person's children using the Person to Person page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **A script can be used to set up a hierarchy.** The demonstration database contains a script that guides users through the process of setting up a hierarchy. Please speak to your implementation staff if you'd like to import this script for use in your implementation.

After a hierarchy is set up, the system displays its members in the Person Tree Zone (this is a zone on the Customer Information Portal). In addition, if the persons in a hierarchy have accounts, summary information about these accounts is displayed in the Account Summary Zone.

While hierarchies would typically be set up to define parent companies and subsidiaries, you can use this functionality for many other purposes. For example, a customer relationship manager could set up a hierarchy containing their "important" customers. To do this, they'd need to create a "dummy" person for the parent and then link the "important" customers to it. After this hierarchy is set up, the manager could see a summary of these customers in the Account Summary Zone whenever they display the "dummy" person.

Navigating The V

The product provides two options for Navigating the V:

- Control Central
- Customer 360

Control Central is the name given to the functionality that helps you:

- Find a person.
- Find an account.
- Find a premise.
- View information about a person / account / premise.
- View alerts and messages about a person / account / premise.
- Navigate to pages that contain information related to a person / account / premise.

Customer 360 is an updated and more configurable alternative to Control Central. The Customer 360 functionality works in conjunction with Unified Search, which provides for a more sophisticated, single-box search for customers or premises. Once Unified Search has populated a customer and/or premise, you are brought to the Customer 360 portal to see an overview of the customer.

> **NOT_SUPPORTED**
>
> For more information about the relationships between persons, accounts, and premises, see Understanding The "V".

- Control Central
- Customer 360

Copyright © 2026, Oracle and/or its affiliates.

- Customer Intelligence Hub

Control Central

The following sections describe Control Central features.

- Control Central - Search
- Control Central - Account Information
- Control Central - Customer Information
- Control Central - Account Tree
- Control Central - Premise Tree
- Control Central - Bill/Payment Tree
- Control Central - Payment Agreements
- How To Add A New Customer From Control Central

Control Central - Search

Open **Menu**, and then **Control Central Search** to find a customer and/or a premise.

Keyboard Shortcut	Alternate Shortcut
`Alt+L`	`Alt+Shift+L`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

> **Note**
>
> **Automatic transfer to Control Central — Account Information.** You are automatically transferred to Control Central - Account Information when an account / premise is selected on this page.

The contents of this section describe how to use control central search.

- Search Facilities
- Search Options
- Wild Cards
- Search Results

Search Facilities

Control Central allows users to search for a person, account and/or premise using a variety of search criteria.

The following search options are provided in the base product:

Copyright © 2026, Oracle and/or its affiliates.

- **Name and Address:** searches for an accounts, person and / or premise using a person's name and / or a service address.
- **Account ID:** searches for an account using an Account ID.
- **Person Contact:** searches for accounts / persons using a person contact.
- **Phone Number:** searches for accounts / persons using a phone number.
- **Person ID Type/Value:** searches for accounts / persons using a person identifier.
- **Geo Type/Value:** searches for a premise using the geographic identifier of the premise or of one if its service points.

Refer to each search option's zone help for details on the search criteria.

From the search results lists, you can select a record to navigate to Control Central - Account Information, where information about the customer is displayed. You can also navigate to other pages in Control Central to view more information about the account, person and / or premise.

> **Note**
>
> **Automatic transfer to Account Information tab.** If your search criteria result in a single customer being retrieved, you are automatically transferred to Control Central - Account Information with that customer's information displayed.

Meter Search

Control Central does not provide an option to search using a meter ID. If you know the meter's identifier, transfer to 360 Search by Device and search for the meter. Use the hyperlinks to navigate to 360 Degree Views, to see more details about the meter and its related measuring component(s) and service point.

Item Search

Control Central does not provide an option to search using an item ID. If you know the item's identifier, transfer to 360 Search by Device and search for the item. Use the hyperlinks to navigate to 360 Degree Views, to see more details about the item and its related service point.

Search Options

The **CIS Division** filter can be specified to limit the search to accounts or premises in a particular CIS Division. The matching is done on either the account's CIS Division or the premise's CIS Division depending on whether the search is account-oriented or premise-oriented:

- If searching by only a name, the search results that have an associated account will be matched on the account's CIS Division.
- If searching by a name and any of the address constituents, the search results that have an associated account will be matched on the account's CIS Division.
- If searching by only the address constituents, the search results will be matched on the premise's CIS Division, regardless of whether or not the premise is linked to an account.

The **Show All Premises** search option controls whether you see all of an account's premises in the search results. If turned on, every premise linked to an account is shown in the search

Copyright © 2026, Oracle and/or its affiliates.

results. If turned off, only one premise is shown per account (and this premise is selected at random).

This switch pertains to the following search methods:

- Search by name only (i.e., all address constituents are left blank)
- Search by account ID
- Search by person phone number
- Search by person identifier

This switch does NOT pertain to the following search methods as these are premise-oriented, not account-oriented:

- Search by name when any address constituent is entered
- Search by address constituents only (i.e., name is blank)
- Search by geographic identifier

The benefit of turning this switch off is that an account will be immediately selected when you enter unique account-oriented search criteria. For example, if you specify an **Account ID** and press enter, the system automatically selects the account; you don't have to wait for the search results to be populated with the multitude of premises linked to the account (and then select one of the rows). Please note, an alert highlights if the selected account has other premises.

> **Note**
>
> **Default.** The first time you open this page, **CIS Division** and **Show All Premises** are defaulted from your user preferences. When you return to this page, the last values are defaulted.

The **Expand Results** search option is available if your implementation has enabled expanded / fuzzy searching. (Refer to Advanced Search Options for more information.) This functionality relies on special indexes configured in the database, to attempt to include similar words or alternate spellings of a supplied search value. This expanded searching is applied only to searches by name and / or address constituents.

Wild Cards

The control central searches against **Name**, **Address**, and **City** support wild cards. The wild card character is % and represents any number or character. The system always appends a % for you when you use these search methods. Examples will help clarify this functionality:

- If you enter a B, the system searches for B% (and will find all objects that begin with B).
- If you enter B%N, the system searches for B%N%. It will find anything that
  - Starts with a B, and
  - has zero or more other characters followed by an N.
- If you enter %B the system searches for %B%. It will find anything with a B in it, no matter what it starts and ends with.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Certain wildcard searches can result in lengthy response times.** If you use the wildcard character to prefix your search criteria (e.g., %San Fran), lengthy response times can result. Why? Because the system will have to look at EVERY customer in the system before it can return the results.

Search Results

The search results at the bottom of the page contain persons, accounts, and premises that match your search criteria. The information displayed in this area differs depending on the type of search you perform. You can perform searches on a customer's name, address, phone number, person contact, person identifier and geographic value.

The information strings that display in the **Person/Account** and **Premise** columns vary depending on what context information is present:

- For a person that is not linked to an account, the Person/Account information string will include the text ‘No Account Information’ and the Premise column will be blank.
- For an account that does not have a service agreement, the Person/Account information string will include the text ‘No Service Agreement’ and the Premise column will be blank.
- For an account with a service agreement that is not currently linked to a service point, the Premise column will display ‘No Premise Information’.
- For a premise that is not linked to an account, the Person/Account column will display ‘No Person Information’.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The first 300 matches are retrieved and displayed.** The base search zones are configured with the *Number of Rows to Display* and *Number of Rows to Retrieve for SQL* parameters set to 300. If you do not find the account / person / premise you are looking for in the set of results, refine your search criteria. A *Control Central Search Factor* feature configuration option (in the *General System Configuration* feature type) controls the maximum number of indexed results that will be evaluated by name and/or address before filtering is done. For instance, if the search factor is set to 3000 and a search is for 'Smith%' on '%Main St', the search will get the first 3000 'Smith%' from the name index and the first 3000 '%Main St' from the address 1 index. Then it will filter out non-matching combinations. Regardless of the resulting set, the search will not return more than the Number of Rows to Retrieve for SQL, which is set to 300. The search factor can be set to any number from 301 to 50000. Note that the higher the factor used, the more possibility of affecting performance. If this option is not specified, the default is 3000.
>
> **Automatic selection if only one match.** You don't have to select a record if there is one and only one object that matches your search criteria. The system automatically selects it and transfers you to Control Central - Account Information.
>
> **Use Tab and Enter to select an account / person / premise.** You don't have to use the mouse to select a record when multiple matches are returned. The system highlights the first row in the search results. If this is the account / person / premise you are looking for, press **Enter** to select it. If not, press **Tab** until you've highlighted the desired record and then press **Enter** to select it.
>
> **Use the Work List sidebar zone to navigate to any account / person in the search results.** When you select a record from the search results, you are automatically transferred to Control Central - Account Information. If you want to look at a different account that appeared in the search results, you do not have to return to the **Main** tab. You can use the **Work List** sidebar zone to load another account without leaving the Control Central - Account Information page. To build the work list, run the search and click on the Work List button that appears in the **Person /Account** column heading. This will copy over the contents of the **Person/Account** column into the **Work List** zone in the sidebar. Refer to Work Lists for more information.

Control Central - Account Information

Once Control Central - Search has populated an account, you are brought to the Account Information page to see an overview of the account.

Keyboard Shortcut	Alternate Shortcut
`Alt+I`	`Alt+Shift+I`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

The following points should be noted about this page:

Copyright © 2026, Oracle and/or its affiliates.

- Selecting an account on control central causes the global context information to be refreshed. Various zones available on this page use the value in the global context to display relevant data for the appropriate account, person and premise.
- This portal page may be configured by the user. Refer to Each User Can Customize Which Zones Appear for information about how to configure which zones appear.
- The Go To Control Central option on the account context menu navigates to this page.

The contents of this section describe the zones that are available on this portal page.

- Account Activity History Zone
- Account Financial History Zone
- Alert Zone
- Applicable Campaigns Zone
- Bill Graph Zone
- Billed Consumption Zone
- Current Customer Zone
- Credit and Collections Info Zone
- Customer Information Zone
- Financial Information Zone
- Premise Information Zone
- SA Premise List Zone
- Service Credit Membership Zone
- Scheduled Payments Log
- Timeline Zone - Account Info

Account Activity History Zone

The **Account Activity History Zone** is a grid that summarizes a variety of account-related events (in reverse chronological order). Clicking on an information hyperlink transfers you to an appropriate page. The following table lists the type of information that may appear in this zone and the portal to which you will be transferred on click of the information hyperlink.

Activity Label	What Is Displayed	Drill Down Transfers You To
Severance Event. *One row is displayed for every* ***Pending*** *and* ***Awaiting Field Activity Completion*** *severance event linked to the account's service agreements.*	The severance event's description, status and trigger date (if non-blank). If the event is dependent on earlier events, the earlier events' type and status is appended	Severance Process - Events

Copyright © 2026, Oracle and/or its affiliates.

Severance Process. *One row is displayed for every **Inactive** severance process linked to the account's service agreements. **Inactive** processes are displayed (as opposed to **active**) because **active** severance processes have **pending** events and **pending** events are shown separately.*	The severance process's SA type description, severance process template description, status and create date/time	Severance Process - Main
Collection Event. *One row is displayed for every **Pending** collection event linked to the account.*	The collection event's description, status and trigger date	Collection Process - Events
Collection Process. *One row is displayed for every **Inactive** collection process linked to the account. **Inactive** processes are displayed (as opposed to **active**) because **active** collection processes have **pending** events and **pending** events are shown separately.*	The collection process's control, template, status, status reason and create date/time	Collection Process - Main
Overdue Event. *One row is displayed for every **Pending** overdue event linked to the account.*	The overdue event's information based on the overdue event type's Event Information algorithm, if configured. Otherwise, default overdue event information.	Overdue Process - Events
Overdue Process. *One row is displayed for every **Inactive** overdue process linked to the account. **Inactive** processes are displayed (as opposed to **active**) because **active** overdue processes have **pending** events and **pending** events are shown separately.*	The overdue process's information based on the overdue process template's Overdue Process Information algorithm, if configured. Otherwise, default overdue process information.	Overdue Process - Main
Cut Event. *One row is displayed for every **Pending** cut event linked to the account.*	The cut event's information based on the cut event type's Event Information algorithm, if configured. Otherwise, default cut event information.	Cut Process - Events
Cut Process. *One row is displayed for every **Inactive** cut process linked to the account. **Inactive** processes are displayed (as opposed to **active**) because **active** cut processes have **pending** events and **pending** events are shown separately.*	The cut process's information based on the cut process template's Cut Process Information algorithm, if configured. Otherwise, default cut process information.	Cut Process - Main
Write-off Event. *One row is displayed for every **Pending** write-off event linked to the account.*	The write-off event description, status and trigger date	Write-off Process - Events

Copyright © 2026, Oracle and/or its affiliates.

Write-off Process. *One row is displayed for every* ***Inactive*** *write-off process linked to the account.* ***Inactive*** *processes are displayed (as opposed to* ***active****) because* ***active*** *write-off processes have* ***pending*** *events and* ***pending*** *events are shown separately.*	The write-off process template's description, status and create date/time	Write-off Process - Main
Customer Contact. *One row is displayed for every customer contact associated with the main customer linked to the account.*	The main name of the person associated with the customer contact, the customer contact's class and type, and the customer contact's date	Customer Contact - Main
Field Activity. *One row is displayed for every unique non-cancelled field activity linked to the account's non-cancelled/ non-closed service agreements.*	If the field activity not Canceled, the field activity type is orchestrated by Service Order Management (SOM) and the field activity is synchronized: SOM field activity information, Outbound Communication information. Otherwise, premise information and field activity information (from installation options)	Field Activity - Main
Pay Plan. *One row is displayed for every non-cancelled pay plan linked to the account*	The pay plan's type, status, date, and amount	Pay Plan - Main
Payment Arrangement. *One row is displayed for every non-cancelled payment arrangement (i.e., service agreement with a special role of Payment Arrangement) linked to the account*	The service agreement's division, SA type, status, start option (if applicable) and effective period.	Payment Arrangement - Main
Credit Rating History. *One row is displayed for every* credit rating transaction *that affects the account's credit rating*	The effect on the account's credit rating, the end date, creation date and the process that created the transaction.	Account - C&C
Cash-Only History. *One row is displayed for every* credit rating transaction *that affects the account's cash-only score*	The effect on the account's cash-only score, the end date, creation date and the process that created the transaction.	Account - C&C
Case. *One row is displayed for every case associated with the account.*	The case's information based on the Case Information algorithm, if configured on installation options. Otherwise, default case information.	Case - Main
Rebate Claim. *One row is displayed for every* rebate claim .	The rebate claim's standard information.	Rebate Claim
Business Flag. *One row is displayed for every* business flag .	The business flag's standard information.	Business Flag

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> The first 25 rows are displayed when this zone is initially built. If more rows exist, a **Next** button appears. If you click this button, the system appends the next set of records.

Account Financial History Zone

The **Account Financial History Zone** lists an account's financial events in reverse chronological order. You can use this grid to both view high-level information about these events and to transfer to the respective page in which an object is maintained.

> **Note**
>
> **Approximately a year and half is shown.** This zone shows all financial transactions within 20 months of the latest financial transaction. This limitation exists to prevent this zone from becoming unmanageably long.

The following columns are displayed in the grid:

Arrears Date This is the date the event starts aging. This column will be blank if the FT has not started aging yet.

Financial Transaction Type This column indicates the type of financial event: Bill, Payment, Bill Cancellation, Pay Cancellation, Adjustment, and Adjustment (Cancel). If the event is related to an adjustment, the adjustment type's description is displayed instead of "Adjustment".

Current Amount This column shows the financial event's effect on the account's current balance.

Current Balance This column shows the account's current balance after the financial event.

Payoff Amount This column shows the financial event's effect on the account's payoff balance. **Payoff Amount** will be dimmed if it is the same as its **Current Amount**.

Payoff Balance This column shows the account's payoff balance after the financial event. **Payoff Balance** will be dimmed if it is the same as its **Current Balance**.

If you need to see more information about a specific financial transaction, press the go to button to transfer to the respective page in which the information is maintained.

> **NOT_SUPPORTED**
>
> For information about current and payoff balance, refer to Current Amount versus Payoff Amount.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> The first 25 rows are displayed when this zone is initially built. If more rows exist, a **Get All** button appears. If you click this button, the system retrieves a maximum of 750 rows.

Alert Zone

The **Alert Zone** is a grid that contains messages highlighting a variety of situations. Clicking on the hyperlink transfers you to an appropriate page.

Account alerts are implementation-specific. On Account - Alerts, a user can define account-specific alerts that should be displayed whenever the account is selected. The Alert Types control table contains the possible alert messages.

Alerts are produce from one of two sources.

- Alerts are primarily produced by plugging in alert algorithms on the installation record. The alerts are shown based on the configured sequence on the installation record. Refer to Count Customer Contacts of a Given Type (CC-NBR-VALID) and Count Pay Plans with a Given Status (PP BY STATUS) for examples of such algorithms.
- Alerts can also be created by system-generated alerts, which are also referred to as legacy alerts. To enable the system-generated alerts, set the Use Legacy Alerts **Option Type** on the General System Configuration Feature Configuration to Y. When this feature is enabled, the system-generated alerts are executed before the plug-in alert algorithms.

> **Note**
>
> A plug-in version for each of the system-generated alerts is provided with the product in addition to many without a system-generated equivalent. If the legacy alert feature is enabled, you would not use the equivalent plug-in alert algorithm, otherwise, these alerts may appear twice on the **Alert Zone**.
>
> You can introduce your own alert algorithms to highlight additional alert conditions based on your business rules.

The table below lists the system-generated alerts that may appear if the use legacy alert feature is enabled and the page to which you will be transferred if you click on the hyperlink. The table below lists alerts in alphabetical order, but it includes a column indicating the order in which the alert will appear.

Alert Text	Order	Description	Drill Down Transfers You To
Account alerts	6	Appears for each user-defined alert associated with an account.	Account - Alert

Copyright © 2026, Oracle and/or its affiliates.

Account has Multiple Premises	24	Appears when the account has service agreements whose service points reference multiple premises	Control Central - Account Tree
Account SA(s) linked to Umbrella Agreement	41	Appears when the account has one or more service agreements linked to a non-canceled terms of service record for an umbrella agreement that is in effect on the current date.	Terms of Service - Main
Active Write Off Process	10	Appears when an account has an active write off process. Refer to How Are Write off Processes Cancelled for information describing when a write-off process is deactivated.	Write Off Process
Appointment Exists	37		Appointment
Auto-Pay Active: *auto pay method description*	25	Appears when the account has an automatic payment option effective on the current date. The auto pay method description (i.e. Direct Debit or Payment Advice) appears only if payment advice functionality is enabled. Refer to Payment Advices for more information on payment advice functionality.	Account - Auto Pay
Cash-Only Account	11a	Appears when the account's cash-only score exceeds the cash-only threshold on the installation record. Refer to How are credit rating transactions created for information describing how a cash-only score is affected by various system and user events.	Account - C & C

Copyright © 2026, Oracle and/or its affiliates.

Collection Process Active	8	Appears when an account has an active collection process. Refer to How Are Collection Processes Cancelled for information describing when a collection process is deactivated.	Collection Process
Collection Referral Active	9	Appears when the account has active collection agency referrals.	Collection Agency Referral
Comment Exists On Account	5	Appears when a free-form comment has been entered for the account (on Account - Main)	Account
Credit rating: NNNN	11b	Appears when the account's credit rating falls below the credit rating threshold on the installation record or the account's CIS Division. Refer to How are credit rating transactions created for information describing how a customer's credit rating is affected by various system and user events.	Account - C & C
Field Activity Pending	19	Appears when any of the premise's service points have a pending field activity	Field Activity
Field activity type alert message	12	Appears when the premise has a service point with a completed field activity whose type has been marked with an alert message (refer to Field Activity Type)	Field Activity
Held Order Exists	32	Appears when the person has a held order.	Order

Copyright © 2026, Oracle and/or its affiliates.

Installation dynamic alerts	42	You can install plug-in algorithms on the installation record that will create alerts (if given situations exist). Refer to Count Customer Contacts of a Given Type (CC BY TYPCL) and Count Pay Plans with a Given Status (PP BY STATUS) for examples of such algorithms	Varies by plug-in
Landlord Exists	27	Appears when a landlord is linked to the premise	Premise
Last Contact: *days old - user name who added the contact*	3	Appears when the person and/or premise has a customer contact. The "days old" presents how old the contact is (based on the contact's date). The word Today is shown if the last contact was on the current date. The word Yesterday is shown if the last contact was on the current date - 1 day. If neither is applicable, the number of days old is shown.	Customer Contact
Life Support / Sensitive Load on Person	1	Appears when any of the account's persons has life support or sensitive load information	Person
Life Support / Sensitive Load on Premise	2	Appears when the premise has life support or sensitive load information	Premise - Misc.
Multiple Financially Resp	26a	Appears when the account has more than one financially responsible person linked to it	Control Central - Account Tree
Non-cash Deposit	15	Appears when the account has non-cash deposits effective on the current date	Account - Deposit

Copyright © 2026, Oracle and/or its affiliates.

n Open Contact(s) for Person / n Open Contact(s) Premise	4	Appears if there are open customer contacts associated with the person and/or premise. Refer to Customer Contacts Can Be Used As Case Files for more information.	Customer Contact
Pending Order Exists	31	Appears when the person has a pending order.	Order
Pending Proposal SA Exists	36	Appears when the account has a proposal service agreement in the pending state. Refer to Proposal SA State Transition for more information.	Service Agreement
Pending Quote Exists	34	Appears when the account has a pending quote. Refer to Quote Lifecycle for more information.	Quote
Pending SA Relationships	16	Appears when the account has SA relationships that are pending	SA Relationship
Pending Start	17	Appears when there is a service agreement in the state of pending start linked to the account. Refer to The Lifecycle Of A Pending Start Service Agreement for more information.	Start/Stop
Pending Stop	18	Appears when there is a service agreement in the state of pending stop linked to the account. Refer to The Lifecycle Of A Pending Stop Service Agreement for more information.	Start/Stop
Person is linked to Multiple Accounts	22	Appears when the person has multiple accounts. A person's accounts are maintained on the Account - Person page.	Control Central - Customer Information
n Persons on Account	26c	Appears when the account has multiple persons linked to it. An account's persons are maintained on the Account - Person page.	Control Central - Account Tree

Copyright © 2026, Oracle and/or its affiliates.

Premise has Multiple Accounts	23	Appears when the premise has service points that are linked to non-closed non-canceled service agreements that are linked to different accounts.	Control Central - Premise Tree
Premise has n Child Premise(s)	40	Appears when the premise is defined as the parent premise for one or more premises. Refer to Define Premise Hierarchy for more information.	Premise Management
Premise is linked to Parent Premise	39	Appears when the premise defines a parent premise. Refer to Define Premise Hierarchy for more information.	Premise
Quotable Proposal SA Exists	35	Appears when the account has a proposal service agreement in the quotable state. Refer to Proposal SA State Transition for more information.	Service Agreement
Quote Awaiting Customer Response Exists	33	Appears when the account has a complete quote with a quote detail that references a quotable proposal SA. Refer to Quote Lifecycle for more information.	Quote
Reactivated SAs Exist	21	Appears when any of the account's service agreements exist in the state of reactivated (i.e., a financial transaction has been generated after the service agreement was closed). Refer to The Lifecycle Of A Service Agreement for more information.	Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

SA type alert message	13	Appears when the customer has a service agreement with an SA type that has been marked with an alert message (refer to SA Type - Details) Note that if the account in control central has more than one service agreement, pressing the drill down button will also launch the service agreement search dialog.	Service Agreement
SA type description and status of a non-billed budget service agreement	14	Appears for every pending start, pending stop, and active Non-Billed Budget that is linked to the account.	Non-Billed Budget Service Agreement
Seasonal Address Exists	38b	Appears if the person has a seasonal address and there is no seasonal address active on the current date.	Person - Correspondence Info
Seasonal Address Currently Effective (MM/DD - MM/DD)	38a	Appears if the person has an active seasonal address that is effective on the current date. The date range that the seasonal address is effective is included in the alert.	Person - Correspondence Info
Severance Process Active	7	Appears when any of the account's service agreement has an active severance process Appears when an account has an active severance process. Refer to How Are Severance Processes Cancelled for information describing when a severance process is deactivated.	Severance Process
Statement Construct Active	28	Appears if the person has an active statement construct	Statement Construct

Copyright © 2026, Oracle and/or its affiliates.

Third Party Guarantor	26b	Appears when the account has a third party guarantor. An account's third party guarantors are maintained on the Account - Person page.	Control Central - Account Tree

Applicable Campaigns Zone

The **Applicable Campaigns Zone** lists the campaigns that may be of interest to the customer. If you press a campaign's Add Order button, you will be transferred to the Order transaction where you can walk the customer through the campaign's provisions and determine if they are interested in its goods and services.

NOT_SUPPORTED

Refer to The Big Picture Of Campaign Eligibility for more information.

Warning

The first 25 rows are displayed when this zone is initially built. If more rows exist, a **Get All** button appears. If you click this button, the system retrieves a maximum of 750 rows.

Bill Graph Zone

The **Bill Graph Zone** illustrates the ending balance and new charges on each of the customer's historical bills.

Note

Approximately a year and half is shown. This zone shows all bills within 20 months of the latest bill. This limitation exists to prevent this zone from becoming unmanageably wide.

For balance-forward accounts, both ending balances and current charges are shown in the graph. For open-item accounts, only the current charges are shown in the graph.

The following points describe unique features of this zone:

- If you hover the mouse over a bar in the graph, summary information about the bill will display.
- If you click on a bar in the graph, you will be transferred to the bill page with the respective bill displayed.

Copyright © 2026, Oracle and/or its affiliates.

Billed Consumption Zone

The **Billed Consumption Zone** illustrates the amount of consumption on each of the customer's historical bills. Up to three units of measure (UOM) are shown for each bill. The system uses the graph UOM's defined on the SA types associated with the customer's active service agreements. If more than three graph UOM's are detected for a customer, the system uses those associated with the service agreement types with the highest bill print priorities.

The following points describe unique features of this zone:

- If you hover the mouse on a bar in the graph, summary information about the bill will display.
- If you click on a bar in the graph, you will be transferred to the bill page with the respective bill displayed.

> **Note**
>
> **Approximately a year and half is shown.** This zone shows all bills within 20 months of the latest bill. This limitation exists to prevent this zone from becoming unmanageably wide.

Current Customer Zone

The **Current Customer Zone** contains basic information about the Person / Account / Premise on which you are working.

Person This part of the context area contains the person's standard information. Note, this information is formatted by a plug-in algorithm on the installation record. Refer to the base package's person information algorithm (PERS-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately

Account ID This part of the context area contains the unique identifier of the account and the name of its main customer.

Adjacent to the account ID appears a "check digit". This is for information purposes only and is not needed to operate the system. Refer to the Description of Page section under Account - Main for a description of how "check digit" is calculated.

Premise This part of the context area contains the premise address. The address information is formatted by a plug-in algorithm on the installation record. Refer to the base package's premise format algorithm (PREM-INFO) for an example. If you prefer a different format, your system administrator should configure the system appropriately

Below **Premise**, the account's **Current Balance** and **Payoff Balance** are displayed. **Payoff Balance** will be hidden when it's the same as the **Current Balance**. Refer to Current Amount versus Payoff Amount for more information.

Aged Debt Bar If the account's **Current Balance** is greater than zero, a colored bar is displayed adjacent. This bar provides information about the age of the account's debt.

The number of segments that appear in this bar is dependent on the elements of the customer's aged balance. The following segments may appear:

Copyright © 2026, Oracle and/or its affiliates.

- disputed debt (If you handle disputed debt as described in Disputing Items),

- new charges and debt between 0 and X days old (where X is the account's customer class's bill due days),

- debt between X to 30 days old,

- debt between 31 to 60 days old,

- debt older than 60 days

> **Note**
>
> If you hover the mouse pointer over a segment of the bar, the respective amount of debt will appear. If you click on the arrears bar you will be taken to the Account Financial History page. Note that information about debt balances by age is not applicable to open-item accounts.

Start Service Push this button if you want to start service for the account and the service isn't at a premise (e.g., a deposit). Refer to The Big Picture Of Starting Service for more information.

Stop Service Push this button if you want to stop service for the account at the premise. Refer to The Big Picture of Stopping Service for more information.

Credit and Collections Info Zone

The **Credit and Collections Info Zone** is a grid that contains a variety of credit & collections-oriented events. Pushing the button adjacent to the information transfers you to an appropriate page. The following table lists the type of information that may appear in this zone and the page to which you will be transferred if you push the adjacent button.

Activity Label	What Is Displayed	Drill Down Transfers You To
Severance Process. *One row is displayed for every **Active** severance process linked to the account's service agreements.*	The severance process's description and create date/ time	Severance Process - Main
Collection Process. *One row is displayed for every **Active** collection process linked to the account.*	The collection process's description and create date/ time	Collection Process - Main
Write-off Process. *One row is displayed for every **Active** write-off process linked to the account.*	The write-off process's description and create date/ time	Write-off Process - Main
Overdue Process. *One row is displayed for every **Active** overdue process linked to the account.*	The overdue process's standard information string (this is dynamic as is constructed by an algorithm)	Overdue Process - Main
Cut Process. *One row is displayed for every **Active** cut process linked to the account's service agreements.*	The cut process's standard information string (this is dynamic as is constructed by an algorithm)	Cut Process - Main

Copyright © 2026, Oracle and/or its affiliates.

Pay Plan. *One row is displayed for every **Active** pay plan linked to the account*	The pay plan type's description and create date/time	Pay Plan - Main
Payment Arrangement. *One row is displayed for every non-cancelled, non-closed payment arrangement (i.e., service agreement with a special role of Payment Arrangement) linked to the account*	The service agreement's SA type description, start date and status	Payment Arrangement - Main

> **Warning**
>
> The first 25 rows are displayed when this zone is initially built. If more rows exist, a **Get All** button appears. If you click this button, the system retrieves a maximum of 750 rows.

Customer Information Zone

The **Customer Information Zone** is a grid that contains information about the current person and account. Pushing the button adjacent to the information transfers you to an appropriate page. The following table lists the type of information that may appear in this zone and the page to which you will be transferred if you push the adjacent button.

> **Note**
>
> Rows are suppressed if the related data is blank for the person / account.

Label	What Is Displayed	Drill Down Transfers You To
Account ID	The account's unique identifier (i.e., account ID) is displayed	An account context menu has been provided (allowing you to drill to a variety of account-oriented pages)
Main Customer	The main customer's name. Note, the person's name is formatted by a plug-in algorithm on the installation record. Refer to the base package's name format algorithm (PERS-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately	A person context menu has been provided (allowing you to drill to a variety of person-oriented pages)
Set Up Date	The account's setup date	Account - Main
CIS Division	The account's CIS division	Account - Main
Customer Class	The account's customer class	Account - Main

Copyright © 2026, Oracle and/or its affiliates.

Bill Cycle	The account's bill cycle	Account - Main
Current Credit Rating	The account's current credit rating	Account - C&C
Next Credit Review Date	The next date on which the account's debt will be reviewed by the account debt monitor. Refer to When Is An Account's Debt Monitored for more information.	Account - C&C
A separate row is displayed for each characteristic linked to the account. Each row's label is the characteristic type's description. If the characteristic is future dated, the effective date enclosed in parentheses prefixes the description.	The characteristic's value	Account - Characteristic
Auto Pay Source	The account's auto pay source	Account - Auto Pay
A separate row is displayed for each miscellaneous person linked to the account. Each row's label is the person relationship type's description.	The main name of the miscellaneous person	A person context menu has been provided (allowing you to drill to a variety of person-oriented pages)
Budget Plan	The account's budget plan	Account - Budget
A separate row is displayed for each characteristic linked to the person. Each row's label is the characteristic type's description.	The characteristic's value	Person - Characteristic
A separate row is displayed for each telephone number linked to the person. Each row's label is the phone type's description.	The telephone number	Person - Main

Copyright © 2026, Oracle and/or its affiliates.

A separate row is displayed for each identifier linked to the person. Each row's label is the ID type's description.	The person identifier value	Person - Main

Note

The system supports encryption for Person Identifier. If your implementation has configured the system to encrypt Person Identifier, the data will be displayed with a masked value; such as, ******.Refer to Encrypting Sensitive

Copyright © 2026, Oracle and/or its affiliates.

		Data for more information.
Email Address	The person's email address	Person - Miscellaneous
Service Provider.	The service provider's name and a description of the service provider's billing relationship as of the effective date.	SA Relationship - Main

> **Warning**
>
> This zone does not support User Interface Masking. If your implementation masks any of the information that appears on this zone, you must disable it and implement a zone that will mask your data appropriately. The demonstration system provides a sample zone that your implementation can import and then customize if necessary; this zone's code is CI_CIMAP. To disable the base package zone simply deny access to its associated application service. You should only grant access to your new Customer Information zone.

Financial Information Zone

The **Financial Information Zone** is a grid that contains financial information related to the account. Clicking the hyperlink transfers you to the appropriate page. The following table lists the type of information that appears in this zone and the page to which you are transferred if you click the adjacent icon.

> **Note**
>
> Rows are suppressed if the related data is blank for the person / account.

Label	What Is Displayed	Drill Down Transfers You To
Current Balance	The account's current balance. If the account's current balance is greater than zero, a colored bar is displayed adjacent. This bar provides information about the age of the account's debt. Refer to Current Customer Zone for more information about this bar.	Clicking on the Arrears Bar or the hyperlink takes you to Account Financial History - Main

Copyright © 2026, Oracle and/or its affiliates.

Payoff Balance	The account's payoff balance. It will only be displayed if the amount differs from the current balance.	Not applicable
Last Bill	The date and amount of the account's last bill along with its due date.	Bill - Main
Last Payment	The date and amount of the account's last payment.	Payment Event - Main
Previous Bill	The date and amount of the bill prior to the last bill for the account.	Bill - Main
Next Bill Date	The next date on which a bill is scheduled to be produced for the account (based on the account's bill cycle's schedule)	Not applicable
Pending Bill Exists	The date of the pending bill. This row only appears if there is a pending bill associated with the account	Bill - Main
Freezable Bill Segments	If freezable bill segments exist, this row contains the number of segments and their total amount.	Bill Segment - Main
Incomplete Adjustments	If incomplete adjustments exist, this row contains the number of adjustments and their total amount.	Adjustment - Main
Freezable Adjustments	If freezable adjustments exist, this row contains the number of adjustments and their total amount.	Adjustment - Main
Payment(s) Pending Upload	Appears when the account has a pending payment staging record (i.e., the account has an external payment that has not been uploaded into the system). To prevent payments from being created and frozen before the actual payment is received, external payments are not uploaded into the system until their accounting date is reached. Note that the amount displayed represents the amount tendered for the account, but does not reflect how the payment may be distributed (i.e., the payment may be distributed to other accounts). For information about payment staging records, refer to Interfacing Payments from External Sources.	Payment Upload Staging

Copyright © 2026, Oracle and/or its affiliates.

Incomplete Payment(s)	Appears when the account has payments that are incomplete. This row contains the number of payments and their total amount. Note, incomplete automatic payments that are pending distribution are not included in this alert as they are highlighted on the **Auto Pay will be distributed on** described below.	Payment - Main
Payment(s) with Errors	Appears when the account has payments that are in error. This row contains the number of payments and their total amount.	Payment - Main
Freezable Payments)	If freezable payments, this row contains the number of payments and their total amount.	Payment Event - Main
Auto Pay will be created on	This row shows the date that the next auto pay will be created. This alert will appear after a bill is completed for an account on auto pay and when your installation option's **Auto Pay Creation Option** is Create on Extract Date. If the installation option is Create at Bill Completion, then the Autopay will be created immediately at bill completion time and no information is displayed here. Refer to How And When Are Automatic Payments Created? for more information.	Bill - Main
Auto Pay will be distributed on	This row shows the date that the next auto pay will be distributed. This alert will appear after a bill is completed for an account on auto pay and when your installation option's **Auto Pay Creation Option** is Create on Extract Date. If the installation option is Create at Bill Completion, then the payment for the Autopay will be created immediately at bill completion time and no information is displayed here. Refer to How And When Are Automatic Payments Created? for more information.	Payment Event - Main

Copyright © 2026, Oracle and/or its affiliates.

Premise Information Zone

The **Premise Information Zone** is a grid that contains information about the current premise. Pushing the button adjacent to the information transfers you to an appropriate page. The following table lists the type of information that may appear in this zone and the page to which you will be transferred if you push the adjacent button.

> **Note**
>
> Rows are suppressed if the related data is blank for the premise.

Label	What Is Displayed	Drill Down Transfers You To
Premise Information	The premise's address. Note, the address information is formatted by a plug-in algorithm on the installation record. Refer to the base package's premise format algorithm (PREM-INFO) for an example. If you prefer a different format, your system administrator should configure the system appropriately	A premise context menu has been provided (allowing you to drill to a variety of premise-oriented pages)
CIS Division	The premise's CIS division	Premise - Main
A separate row is displayed for each characteristic linked to the premise. The label in each row is the respective characteristic type's description.	The characteristic's value	Premise - Characteristic
Landlord	The name of the premise's landlord	Landlord Agreement
Service Point Information. *Note, a separate row is displayed for each service point linked to the premise*	The service point's type and service cycle (if specified)	A service point context menu has been provided (allowing you to drill to a variety of service point-oriented pages)
Device *This row is displayed if a device is currently installed at the service point. If so, this row's label is indented and displayed beneath the related service point*	The device's badge number, device type, installation date, and status.	A meter configuration context menu has been provided (allowing you to drill to a variety of meter configuration-oriented pages)
Next Meter Read. *This row is displayed if the service point references a service cycle.*	The next scheduled meter read date and an indication of the service point's *service cycle*	No drill down is available

Copyright © 2026, Oracle and/or its affiliates.

Item Information This row is displayed if an item is currently installed at the service point. If so, this row's label is indented and displayed beneath the related service point.	The item's type, badge number, and status.	An item context menu has been provided (allowing you to drill to a variety of item-oriented pages).

SA Premise List Zone

The **SA Premise List Zone** contains a row for each non-cancelled, non-closed service agreement linked to the account. The **Premise** (if any), **Current Balance, Payoff Balance**, and **SA Information** is displayed for each service agreement.

The service agreements appear in the following order:

- Service agreements linked to the premise on which you're working appear first.
- Next appear service agreements linked to the account that don't have a premise (deposits, charitable contributions, etc.).
- Following this are the remaining service agreements linked to the account.

> **Note**
>
> **Dimmed Payoff Balance.** A service agreement's **Payoff Balance** will be dimmed if it is the same as its **Current Balance**. Refer to Current Amount versus Payoff Amount for more information.

> **Warning**
>
> The first 25 rows are displayed when this zone is initially built. If more rows exist, a **Get All** button appears. If you click this button, the system retrieves a maximum of 750 rows.

Service Credit Membership Zone

The **Service Credit Membership Zone** contains a row for each pending and active service credit membership linked to the account. It will also show any inactive membership that contains a balance. The **Service Credit Membership Type**, **Membership Status** and **Balance** (if applicable) are displayed for each membership.

Scheduled Payments Log

This zone displays an account's pending payments. It includes the following

- Pending one time payments initiated from a third party payment processing system.
- Pending one time payments initiated from web self-service

Copyright © 2026, Oracle and/or its affiliates.

- Pending bill-related automatic payment records

Refer to Third Party Payment Processing for more information on third party payment processing.

Timeline Zone - Account Info

If your implementation team has configured one or more timeline zones, they may be visible on this page. Timeline zones show when significant events have occurred in the past and when significant events will occur in the future. For example, a timeline can show when payments and bills have been received for a customer.

Refer to Timeline Zone for a detailed explanation of how a timeline zone displays its information and other functionality that it supports.

Refer to Configuring Timeline Zones for how to add and change timeline zones.

Control Central - Customer Information

Once Control Central - Search has a person context, you can navigate to this page to view an overview of the related persons, customer contacts, accounts, premises, and service agreements linked to the account.

> **Note**
>
> **User configurable.** Refer to Each User Can Customize Which Zones Appear for information about how to configure which zones appear.

Description of Page

> **Note**
>
> **Navigation hint.** The Go To Control Central option on the person context menu navigates to this page.
>
> **A mouse with a roller is useful.** This page can extend vertically past the normal desktop boundary. You will find that a mouse with a "roller" will facilitate navigating through the page.

The contents of this section describe the zones that are available on this portal page.

- Communication History Search
- Person Tree Zone
- Active Account Summary Zone
- Timeline Zone - Customer Info

Copyright © 2026, Oracle and/or its affiliates.

Search Fields

The following options allow you to search for and find the desired communications:

- **Search By** (dropdown) allows you to specify the type of communication that you want to find. You can search for Customer Contacts, Notifications from Oracle Utilities Notification Center and Lead Event logs, or all communications, which combine the previous two.
- **Start Date** and **End Date** allow you to set a targeted range of communications.
- **Search By Context** (dropdown) allows you to specify whether the communication is associated to the Person, Account, or Premise.
- **Hide Notification Logs** allows you to display the search results with or without the log entries.

> **Note**
>
> The Hide Notification Logs option displays when applicable to the selected Search By option.

Results Grid

The following details are displayed for each communication:

- **Date / Time:**
  - For a customer contact related entry, the Customer Contact Date / Time for the communication
  - For lead / lead event related entries, the Log Date / Time where the Log Type is Notification for the communication
- **Details:**
  - For a customer contact related entry, the Contact Class Description / Contact Type Description is displayed
  - For lead / lead event related entries, the Log Messages for both the lead / lead event for the communication is displayed. There will be one entry for the lead and another for the lead event. If the lead / lead event created a customer contact for a communication, then an entry for the customer contact will also exist.
- **Related Object:**
  - For a customer contact related entry, a hyperlink to the customer contact record
  - For lead / lead event related entries, there will be one entry for the lead and another for the lead event.
    - For the lead related entry, a hyperlink to the lead related to the communication
    - For the lead event related entry, a hyperlink to the lead event related to the communication

Person Tree Zone

Copyright © 2026, Oracle and/or its affiliates.

The tree in this zone shows a great deal of information including:

- All accounts linked to the person in context. Refer to Understanding The V for more information.
- Customer contacts linked to this person.
- The hierarchy of parents and children linked this person.
- All aliases linked to this person.

You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Active Account Summary Zone

The **Active Account Summary Zone** lists all accounts related to the person in context where they are the main customer on the account. In addition, accounts linked to this person's children also appear in this zone, and if a child has children, the grandchildren's accounts are included. In fact, the system will look for accounts up to five levels deep (meaning that the great, great grandchildren's accounts will be included in this zone).

> **Note**
>
> **Only accounts linked to child persons with a financial relationship are shown.** When you set up a customer hierarchy, you can define both subsidiaries and "key contacts" (i.e., individuals that you contact at a company). As described above, this zone will include accounts related to these subsidiaries and key contacts. However, you might not want to include the personal accounts related to the key contacts in this zone. You can control which accounts appear in this zone when you set up a hierarchy. You do this by turning on the Financial Relationship switch for those persons whose accounts should be included.

The following columns are displayed in the grid:

Account Info This column contains the standard account information.

Current Balance and **Arrears** This column contains the account's current balance. If the account's current balance is greater than zero, a colored bar is displayed in the **Arrears** column. This bar provides information about the age of the account's debt. Note that information about debt balances by age is not applicable to open-item accounts. Refer to Current Customer Zone for more information about this bar.

Last Billed Info This column contains information about the last completed bill sent to the account.

Last Contact Info This column contains information about the last customer contact associated with the account's main customer.

Timeline Zone - Customer Info

Your implementation may have configured this page to show one or more timeline zones. Refer Timeline Zone - Account Info for a description of how to use these zones.

Copyright © 2026, Oracle and/or its affiliates.

Control Central - Account Tree

Once Control Central - Search has an account context, you can navigate to the **Account Tree** to see an overview of the persons, premises, and service agreements linked to the account.

Description of Page

This page is dedicated to a tree that shows the various objects linked to an account. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Control Central - Premise Tree

Once Control Central - Search has a premise context, you can navigate to the **Premise Tree** to see an overview of the accounts, service points, and service agreements linked to the premise.

Description of Page

This page is dedicated to a tree that shows the various objects linked to a premise. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Control Central - Bill/Payment Tree

Once Control Central - Search has an account context, you can navigate to the **Bill/Payment Tree** to see an overview of the financial transactions linked to the account.

Description of Page

This page is dedicated to a tree that shows the account's bills and payments. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

For balance-forward accounts, bill nodes contain the balance presented on the respective bill, and pay nodes contain the amount of the respective payment. However, for open-item accounts, the tree behaves differently:

- The amount on bill nodes is equal to the sum of the current charges, adjustments and corrections on the bill.
- Each bill or payment will contain an indication if all of its financial transactions are fully matched. If not, the node will become red to highlight that unmatched financial transactions exist.
- If a bill or payment node is expanded, a summary of the match status of its financial transactions is shown.
- Credit and correction notes appear as separate nodes.

Copyright © 2026, Oracle and/or its affiliates.

Control Central - Payment Agreements

Once Control Central - Search has an account context, you can navigate to the **Payment Agreements** page to see an overview of the account's payment arrangement requests and/or pay plans.

The contents of this section describe the zones that are available on this portal page.

- Payment Arrangement Requests
- Payment Arrangement Request Details
- Pay Plan Tree

Payment Arrangement Requests

This zone shows the account’s in-progress and finalized (e.g. granted, denied, canceled, etc.) payment arrangement requests.

You can use this zone to: view high-level information about the payment arrangement request, navigate to the **Payment Arrangement Request** or start a payment arrangement request.

The **Broadcast** icon on each row displays the Payment Arrangement Request Details zone that shows summary information about the payment arrangement request.

The **New Payment Arrangement** action located at the header of this zone is used to start a payment arrangement request.

Refer to this zone’s embedded help for information on other available actions.

Payment Arrangement Request Details

This zone is displayed when a Payment Arrangement Request is broadcast from the Payment Arrangement Requests zone.

It shows key information about the selected payment arrangement request.

Pay Plan Tree

This zone shows the pay plans linked to an account. It displays both current and past (expired) pay plans.

You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

How To Add A New Customer From Control Central

There are three ways to add a new customer from control central:

- You can add a new customer by invoking the Person and Account pages from the Menu dropdown. This is not the recommended approach, as it requires several screen interactions; either of the following approaches is preferable.

Copyright © 2026, Oracle and/or its affiliates.

- If your team has set up sales and marketing campaigns, you can use the order transaction to add a new customer. Refer to Order User Interface Flow for more information. Refer to Sales & Marketing for background information about sales and marketing campaigns.
- Adding a Person from Control Central. You navigate as follows:
  - You start from Control Central - Search. On this page, you determine if the customer already has an account. If so, select it and you're done. If not, proceed to the next step.
  - Click the + button adjacent to the Name field to transfer to Person - Main Information page (with the name you entered already defaulted). Use the Person pages to record all relevant information about the new customer.
  - Enter the customer's names, phone numbers, and IDs. Click Save (or press ALT+S) to add the person and account. The value of **Add Account and Start Service** is defaulted to "on" and the **Customer Class** defaults based on the **Person Type**. This means that when you click save the following will happen: a new person is added, a new account is added, the new person is linked to the new account, and you will be automatically transferred to Start/Stop where you can start service. (**Note:** This is the default behavior of the system. However, an implementation can define different functionality by setting up a customer information options feature configuration.

> **Note**
>
> **Default.** When the system links the person to the account, it defaults the relationship type from the installation options. A different relationship type can be defined for persons versus businesses on the installation record.

If you need to tinker with information on the account, transfer to account maintenance (use the context menu) and make any necessary changes.

Customer 360

Use the unified search box to find a customer and/or premise. After selecting a search result, you are brought to the Customer 360 portal to see an overview of the customer.

Refer to Search for details on how to use the unified search box.

- Customer 360 - Unified Search
- Customer 360 - Program Enrollment
- Customer 360 - Customer Insights
- Customer 360 - Customer Activity
- Customer 360 - Financial Activity

Customer 360 - Unified Search

Use the unified search box to find a customer or premise, in addition to the standard search for menu items. For more complex queries, click the **Advanced Search** to navigate to the advanced search portal.

The contents of this section describe how to use unified search for Customer 360.

Copyright © 2026, Oracle and/or its affiliates.

- Unified Search - Filters and Hints
- Unified Search - Search Results

Unified Search - Filters and Hints

Depending on the text entered into the toolbar search box, filter rules are dynamically performed to evaluate which filter is applied when searching for a customer.

The following filters and associated filter rules are used:

Filter	Filter Rule
Name	Minimum of 5 characters excluding whitespaces
Address	One or more numeric characters followed by space and an alphanumeric character
Account ID	10 numeric characters
ID Number	Social Security Number: 3 numeric characters followed by a dash, 2 numeric characters, a dash and 4 numeric characters
Phone Number	Plus character and 1-3 numeric characters (country code, optional) followed by a space, period, dash or open parentheses, 3 numeric characters (area code), followed by a space, period, dash or close parentheses, followed by 3 numeric characters (exchange code), followed by a space, period or dash, followed by 4 numeric characters (subscriber number).
Email Address	Non-whitespace characters followed by an @ sign, non-whitespace characters, a period, non-whitespace characters

To search explicitly using a specific filter, precede the filter value with one of the following predefined hints:

Filter	Hint	Examples
Name	n:	n:brazil
Address	ad:	ad:123 main
Account ID	ac:	ac:1234567890
ID Number	id:	id:123-45-6789
Contact Info	c:	c:555-2287 c:mbrazil@oracle.com

For more information on configuring unified search, filters, and hints, refer to Understanding Unified Search.

Unified Search - Search Results

When a search filter is applied, search results matching your search criteria are populated below the toolbar search box. The information displayed in this area can include the person, account, or premise information. Additionally, the search filter and search filter value that was used for determining the matching search criteria is displayed. Select the customer or premise to navigate to the Customer 360 portal.

> **Note**
>
> **The first 300 matches are retrieved and displayed**. The base search zones are configured with the **Number of Rows to Display** and **Number of Rows to Retrieve for SQL** parameters set to 300. If you do not find the account / person / premise you are looking for in the set of results, refine your search criteria.

Copyright © 2026, Oracle and/or its affiliates.

Customer 360 - Program Enrollment

The Program Enrollment Zone contains messages highlighting a variety of insights related to the account in context. Insight Types associated to the Insight Class of Program Enrollment are displayed.

The base product provides Insight Types for: Auto Pay Enrollment, E-Bill Enrollment, Budget Billing Enrollment and Customer Web Enrollment. Refer to the Insight Type configuration in the application, for more details.

> **Note**
>
> For more information about Insight Types and Insight Groups, see Contextual Insights.

The following provides high-level steps to define insight types that can be associated to the program enrollment zone:

- Define new Insight Types. For the program enrollment zone, the insight types are intended to be defined with a visual structure of List Insight. Specify an insight algorithm that receives account as input context value.
- Define an Insight Group that references the **Program Enrollment** insight class. Configure the valid visual structure as List Insight.
- Associate the insight types to the insight group.

Customer 360 - Customer Insights

The Customer Insights Zone contains messages highlighting a variety of insights related to the customer (person, account, premise) in context. Insight Types associated to the Insight Class of Customer Insights are displayed.

> **Note**
>
> For more information about Insight Types and Insight Groups, see Contextual Insights.

The following provides high-level steps to define insight types that can be associated to the customer insights zone:

- Define new Insight Types. For the customer insights zone, the insight types are intended to be defined with a visual structure of List or Card. Specify an insight algorithm that receives person, account or premise as input context values. The base package provides algorithm type C1CIALERT (Contextual Insight - Control Central Alert) for exposing control central alerts as contextual insights.
- Define an Insight Group that references the **Customer Insights** insight class. Configure the valid visual structures as List or Card.
- Associate the insight types to the insight group.

In addition to the Customer Insights zone, contextual insights can be configured to potentially render within certain areas of the other Customer 360 zones. The following table lists the available insight classes included in the base package and the associated UI placement within the corresponding customer 360 zone that insights can render. Additionally, the intended valid

Copyright © 2026, Oracle and/or its affiliates.

visual structures are listed as a guideline when defining the associated insight group. Follow the steps as described above to define insight types and insight groups for these zones.

Insight Class	Zone	UI Placement	Input Context	Valid Visual Structures
Customer Context (Upper)	Customer (sidebar zone)	Upper section of the zone, under person name field	PER_ID, ACCT_ID	Badge
Customer Context (Lower)	Customer (sidebar zone)	Bottom section of the zone, under person name field	PER_ID, ACCT_ID	Inline
Financial History	Financial Activity	Upper section of Financial History tab, above the financial history fields	ACCT_ID	Inline
Usage and Billing	Financial Activity	Upper section of Usage and Billing tab, below the filters	ACCT_ID	Inline
Payment Agreements	Financial Activity	Upper section of Payment Agreements tab	ACCT_ID	Inline
Credit and Collections	Financial Activity	Upper section of Credit and Collections tab, above the credit and collections trees	ACCT_ID	Inline

Customer 360 - Customer Activity

The Customer Activity History Zone is a grid that summarizes a variety of account-related events (in reverse chronological order). Use the Filter By drop down to select or deselect activity types to display. Clicking on the activity record transfers you to an appropriate page.

This zone displays similar information as the Account Activity History Zone, with a number of additional activity types added. The following table lists the additional types of information that may appear in this zone and the page to which you will be transferred when selecting the activity record:

> **Note**
>
> The Activity History Configuration Master Configuration can be used to exclude certain activities from appearing in this zone.

Activity Label	What Is Displayed	Drill Down Transfers You To
Bill. One row is displayed for every **Completed** bill linked to the account.	The bill amount, bill due date, number of days in the bill period, and bill date.	Bill

Copyright © 2026, Oracle and/or its affiliates.

Activity Label	What Is Displayed	Drill Down Transfers You To
Payment. One row is displayed for every **Frozen** payment linked to the account.	The payment amount, the payment events tender type(s), payment status, and payment date.	Payment
Self Service Task. One row is displayed for every self service task linked to the account.	The self service task type description, self service task status, and creation date time.	Self Service Task
Business Flag. One row is displayed for every SP-based business flag linked to the accounts SA/SP's.	The business flag type description, status, and create date/time	Business Flag
Device Event. One row is displayed for every device event pertaining to paired device event types (e.g., power outage/restoration) and device event types configured to trigger service issue monitors.	The device event type description, status, and event date/time	Device Event
Service Order Field Activity. One row is displayed for every service order field activity that is not linked to a parent orchestrator activity.	The activity type description, service address information, status	Activity
Service Order Orchestrator Activity. One row is displayed for every service order orchestrator activity.	The activity type description, service address information, status	Activity

Customer Activity History Summarization

Customer Activity History Summarization uses generative Artificial Intelligence (Gen AI) models to compile and provide a summary of recent customer activity history information. The intent is to help agents and other system users understand recent customer activity more easily instead of having to read through a list of recent activity.

> **Note**
>
> Customer Activity History Summarization is available **only** in Oracle Utilities Customer Cloud Service.

See Enabling and Configuring the Customer Activity History Summarization Feature for more information about enabling and configuring this feature. Once enabled, the zone is loaded with the summarization already in place.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> - Customer Activity History Summarization is available under Limited and Controlled Availability for select Early Adopters in approved regions.
> - Customer Activity History Summarization relies on generative AI models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Customer Activity History Summarization.
> - Customer Activity History summaries are prepared by generative AI. Carefully review and verify them against trusted sources, as they may include out-of-date, inaccurate, or incomplete information.

Enabling and Configuring the Customer Activity History Summarization Feature

The Customer Activity History Summarization feature is enabled via the **Activity History AI Summary Configuration** master configuration. This master configuration also allows you to configure the amount of data displayed. Refer to the embedded help for details concerning configuration options in this master configuration.

Customer 360 - Financial Activity

The Financial Activity Zone provides an overview of Usage and Billing, Financial History, and Payment Agreements linked to the account.

The **Usage and Billing** tab provides a graphical view of bills and metered usage. Use the Cost / Usage drop down to toggle between graphs. If the account is associated to multiple premises, use the Premise drop down to view a particular premise.

> **Note**
>
> Usage and Billing settings are configured by adding or changing the Billing and Usage Display Configuration. You can access the portal through **Admin**, and then **General**, and then **Master Configuration**.

The **Financial History** tab shows financial detail highlights and lists the account's financial events including charges, payments, and adjustments in reverse chronological order. The financial details highlight section contains the following financial information about the account:

- Current Balance
- Last Bill
- Next Bill Date
- Autopay Source
- Deposit Amount
- Payoff Amount
- Payoff Balance
- Current Credit Rating
- Next Credit Review Date

Copyright © 2026, Oracle and/or its affiliates.

You can use the financial history grid to both view high-level information about these events and transfer to the respective page in which an object is maintained. Additionally, charge and payment events may be expanded for viewing additional details.

The **Payment Agreements** tab lists scheduled payments for the account's active payment arrangements and/or pay plans, and whether the scheduled payment has been received.

The **Credit & Collection** tab appears when the customer in context is active in a collections, severance, overdue, cut, or write-off process. The tab contains a tree showing the respective process and associated events, along with badge insights alongside each node indicating its status.

> **Note**
>
> Refer to the C1-CollectionsSeveranceProcess, C1-OverdueCutProcessTree, and C1-WriteOffProcessTree tree definitions for more information about the Credit & Collections node definitions and hierarchy.

Customer Intelligence Hub

The Customer Intelligence Hub (the Hub) simplifies standard work processes for customer service agents and managers by providing a variety of tools in a single location that reduce the cost-to-serve and provide elevated customer service. You use the Hub to manage utility billing account information and transactions, and to assist you in answering customer calls.

Accessing the Hub

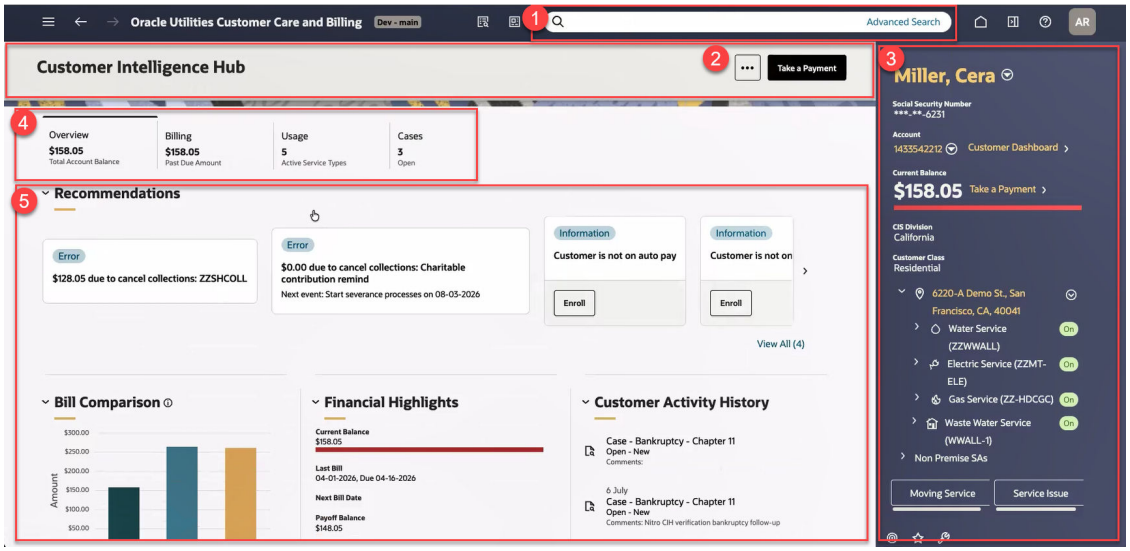
You access the Hub in a number of ways, depending on the features your utility uses:

- After answering a call using Service Guide, if the system recognizes the phone number, the Hub for that account is displayed.
- Use the search bar at the top of your screen to search for and select the account by name or account number.
- Use the Customer 360 Search screen where you can search using multiple filters.

Understanding the Hub Interface

Before using the Hub, you should have a solid understanding of the components that are available to you. This image, and the following table describe each area of the interface.

Copyright © 2026, Oracle and/or its affiliates.



Area	Description
1. Search Bar	Use the Search bar to search for and select the accounts you want to view in the Customer Intelligence Hub.
2. Hub Header	Use the Hub Header to access the **Take a Payment** button, or to complete the following actions by selecting an option from the **Action** menu: • Start service • Stop service • Transfer service • Service call • Outage call • Email customer • Add case • Bookmark • Portal maintenance • Refresh

Copyright © 2026, Oracle and/or its affiliates.

Area	Description
3. Sidebar Portal	The Sidebar is a portal that always appears on the Oracle Utilities Customer application desktop. Its zones contain tools and data that are useful regardless of the object being displayed. Additionally, if your system is integrated with Oracle Utilities Opower, this section can display utility configured badges to notify users of important information. For example, you might see a badge that the customer is potentially a Low or Moderate Income (LMI) customer, and might be eligible for affordability programs. **Note** When using the Hub, it is recommended that the Sidebar Portal be configured to show the **Current Context Zone** at or near the top so that the user can see which account they are viewing.
4. Tabs	Use tabs to view different types of account information. Each tab header include an important customer data point. For example, the Overview tab header displays the total account balance. Tabs include: • Overview tab • Billing tab • Usage tab • Cases tab
5. Work Area	Below the tabs is the work area. This area of the form is made up of different zones, and changes depending on which tab you select and which task you are completing. Use this section of the screen to search for, select, and manage records. If your utility is using Oracle Fusion Knowledge Management, zones can be configured to display a book icon in the header. These icons can be configured to launch specific knowledge articles that can assist agents during their calls.

- Using the Hub Header
- Using the Overview Tab
- Using the Billing Tab
- Using the Usage Tab
- Using the Cases Tab

Copyright © 2026, Oracle and/or its affiliates.

Using the Hub Header

Use the buttons in the Customer Intelligence Hub header to carry out tasks associated with the account. The primary action available in the header is the **Take a Payment** button. Additional actions are available in the action menu. Each option navigates you to a page, drawer, or process flow where you can complete your task.

- **Take a Payment:** Use this option to open a payment window where you can record a payment from a customer. See Payments for more information.
- **Start Service:** Use this option to open the Process Flow: Start Service Request Process. See The Start Service Process and Request for more information.
- **Stop Service:** Use this option to open the Process Flow: Stop Service Request Process. See The Stop Service Process and Request for more information.
- **Transfer Service:** Use this option to open the Process Flow: Transfer Service Request Process. See The Transfer Service Process and Request for more information.
- **Service Call:** Use this option to open the Process Flow: Service Call Process. See The Service Call Process and Service Call Communication for more information.
- **Outage Call:** Use this option to open a drawer that you can use to specify the outage type. The system then opens another drawer where you can specify outage details. See Outage Calls for more information.
- **Email Customer:** Use this option to open an email window that you can use to send email to the customer. This option is only valid if your system is integrated with Oracle Fusion Service.
- **Add Case:** Use this option to open a drawer that you can use to specify the type of case you want to create. The system then opens another drawer where you can specify case details. See Case Management for more information.
- **Bookmark:** Use this option to create a bookmark that you can use to easily return to this customer's record. See User - Bookmarks for more information.
- **Portal Maintenance:** Use this option to update your Customer Intelligence Hub portal. This option is only available to system administrators. See Defining Portals for more information.
- **Refresh:** Use this option to refresh the Customer Intelligence Hub screen.

Using the Overview Tab

You use the Customer Intelligence Hub Overview tab to view basic information about the account's current status. The tab header includes the current account balance, and provides these zones where you can view additional information:

- Recommendations
- Customer Activity Summary (Cloud customers only)
- Bill Comparison
- Financial Highlights
- Customer Activity History

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The above image does not include the Customer Activity Summary zone. This zone is only available to Cloud customers who have enabled AI.

- Recommendations
- Customer Activity Summary
- Bill Comparison
- Financial Highlights
- Customer Activity History

Recommendations

The Recommendations zone provides specific recommended actions or insights you can review and take for the customer. Recommendations are displayed as cards across the top of the Overview tab, and you can use the arrows to the right and left of the cards to scroll through them. You can also select the **View All** link below the list of recommendation cards to view all of the recommendations in a drawer.

Some cards are actionable, and include buttons that let you to perform an action, such as enrolling a customer in an e-bill program. For information on how to enroll customers in programs from these insights, see Billing Highlights.

Other cards are informational, and provide details that might assist in answering the customer's query. For example, a card might tell you that the customer is in the severance process, and provide details about the severance dates and the amount due.

If you are using the Service Guide, you can configure the system can detect what is being said while you are talking with the customer. If the list contains recommendations that can assist you with the call, based on the conversation, Service Guide highlights the recommendation cards and moves them to the beginning of the Recommendations zone during the call so you can easily access them.

If your system is integrated with Oracle Fusion Knowledge Management, some cards also contain a book icon that enables you to view a knowledge article about the specific recommendation, standard operating procedures, and program rules, as defined by the utility. For additional information about viewing and using Knowledge Management, see Using Knowledge Management.

Customer Activity Summary

The Customer Activity Summary zone of the Overview tab provides a summary of the activities associated with the customer's account over the period of time that your utility defines in the Master Configurations. For example, the summary cold include activities for the past three months.

This zone provides a quick and easy way to catch up on the important activities that could be impacting your customer, and can help you better understand the state of their account.

The summary is generated using AI, and is based on the details in the customer's activities. At the bottom of the summary, are two notes:

- A description of what information was used to generate the summary.

Copyright © 2026, Oracle and/or its affiliates.

- A disclaimer that the summary was created using generative AI, and should be reviewed for accuracy.

> **Note**
>
> This zone is available only for Cloud customers who have enabled AI. For customers who don't have this zone, the Recommendations zone spans the top portion of the Overview tab.

Bill Comparison

The Bill Comparison zone includes a chart that you can use to compare a customer's current bill to their previous bill and to the corresponding bill from the same time period the previous year. When you place your cursor over a bar in the chart, you can view the date and amount associated with the bill. If you select a bar, the **View Bill Summary** drawer opens on the right side of the screen and provides details about the bill. You can then select specific bill segments to view additional details, or you can select the **bill ID** link to view the Bill page.

Financial Highlights

You can use the Financial Highlights zone of the Overview tab to get an at-a-glance view of the customer's current financial situation. This section provides:

- Current balance
- Date of the last bill, along with the due date
- Date of the next bill
- Payoff balance
- Current credit rating
- Next credit review date

This zone is informational only.

Customer Activity History

The Customer Activity History zone provides a list of the most recent activities that were created for the account. Activities represent specific interactions that the utility has had with the customer. You can select an activity to see additional details, or you can scroll down to the bottom of the zone and select the **View All** link to open the **Customer Activity History** drawer, where you can filter and view all of the customer's activities.

When you select the **View All** link, the drawer opens on the right of the screen, and includes a longer list of activities. You can use the **Filter By** button to refine your activity list to specific categories. For example, to view agent notes that were recorded for a previous call from this customer, you would filter for activities in the Conversation Message category.

By default, the list displays activity records for the past three years. You can use the **Date Range** button to enter a different time frame, but you can't search for more than three years of data at a time.

Activities that display here are grouped by categories, which include:

- **Conversation Message:** These include:

Copyright © 2026, Oracle and/or its affiliates.

  - **Calls**: Call activity records for both inbound and outbound calls are displayed only if you are using a CTI integration. When you select a call activity record, the system displays the name of the agent that took the call, the date and time of the call, and if available, the call resolution and wrap up notes. For customers that have multiple billing accounts, calls in the activity feed are filtered to display only the calls that are associated with the billing account you are viewing.

- **Customer Contact:** Email activity records show the date and time on which an email was sent, the user that sent the communication, and the contact class. When you select the **View Communication** link, you can view the body of the email.

> **Note**
>
> If your system is integrated with Oracle Utilities Opower, when you click the View Communication link, the system opens the Opower email that was sent to the customer in a new window. Examples of Opower emails include High Bill Alerts and Home Energy Reports.

- **Self-Service Task:** Service task records show interactions that were completed by contacts using self-service methods. When you click on a self-service activity record, the system displays details about the task.
- **Financial Information:** This category includes:
  - **Bill:** A bill record displays the date and amount a bill issued on the billing account. Many bills can be produced for the billing account over time. Bills charge for the utility services and products supplied to a customer. When you select a bill activity record, the system opens the Bill drawer, which displays details about the bill.
  - **Payment:** Payment records show all payments that have been made on the billing account. When you select a payment activity record, the system opens the Payment received drawer, which displays details about the payment.
- **Cases:** Case records show all open, closed, an in-progress cases associated with the billing account. The activity feed displays the case title, status, and category. When you select a case activity record, the system opens the case in a drawer.
- **Note:** Note activity records show all notes an agent adds to a billing account. The activity feed displays the user ID of the agent who created the note, the note title, the beginning of the note text, and the date and time the note was created.
- **Rebate Claim:** Rebate claim records identify all instances of a customer applying for a rebate.
- **Service Order:** Service order activities track work requests at service point locations. These activities dynamically orchestrate the required actions based on the current state of the service point and desired outcome. For example, creating an Enable Service activity for a new service request may trigger additional related activities. Service order activities may be completed by field crews or executed automatically, such as through smart meter commands.

> **Note**
>
> Field activities generated through Service Order Management orchestrations and integrations are listed under the Service Order category. For additional information, see the topic "Service Orders" in the Business User Guide.

Copyright © 2026, Oracle and/or its affiliates.

- **Field Order:** Field order activities track work performed by utility field personnel at service point locations. These activities, such as manual meter installations, are sent to the field work system for scheduling and assignment to crews. When the work is completed, the activity status is updated by the field work system.

  > **Note**
  >
  > Field activities created using classic Customer Care and Billing templates and integrations are listed under the Field Order category. For additional information, see the topic "Field Activity Maintenance" in the Business User Guide.

- **Credit and Collection:** This category contains the following record types:
  - Collection Process
  - Severance Process
  - Write Off Process
  - Pay Plan Overdue Process
  - Payment Arrangement
  - Cut Process
  - Credit Rating History
  - Cash Only History
- **Outage Call:** These records identify instances when the customer called in regarding an outage.

Using the Billing Tab

To answer customer inquiries or determine how to best serve a customer, you might need to view information about their utility bills. After accessing a billing account in the Customer Intelligence Hub, you can use the Billing tab to view and analyze billing history, data, and trends. The scorecard in the header includes the amount of the last bill to give you a quick view of how much the customer owes.

Billing account information is displayed in the following sections:

- **Financial History:** This section of the Billing tab displays a record for each bill, payment, and adjustment associated with the billing account over the selected time period. You can view all financial history records, or filter the list to display only bills, payments, or adjustments. You can also specify a time period to display. When you select the **Transaction Type** link for a record in this list, the system displays detailed information about the transaction.
- **Billing Highlights:** This section of the Billing tab provides important information about whether the customer is enrolled in billing programs such as auto pay, budget billing, and e-bill enrollment.
- **Billing History:** This section of the Billing tab displays a chart that represents the billing history for the last six months for the selected account. For each bill period, the chart displays two bars. The first bar can include multiple sections, which represent the amounts associated with each service type included on the bill. The second bar represents the ending balance for that bill. You can select a bar in the chart to view bill details, or select the **View more details** link to view additional billing history.
- Financial History

Copyright © 2026, Oracle and/or its affiliates.

- Billing Highlights
- Billing History

Financial History

Use the Financial History section of the Billing tab to view bills, payments, and adjustments. By default, all three types of history records are displayed in the Financial History list on the Billing tab. To view details about a record, select the hyperlink in the **Transaction Type** column. The system opens the record in a drawer, where you can view additional detail, or navigate to the page where you can view and manage the record.

If your utility is using Oracle Fusion Knowledge Management, a book icon appears in the header of the zone. When you select the icon, a drawer opens with a pre-confgured knowledge article that can help agents answer calls about financial history.

You can use the filter fields above the list to narrow the results to the type of record or to a specific date range. For example, you can filter by:

- All Activity
- Bill
- Payment
- Adjustment

When specifying a custom date range, you can't select a date range that spans more than three years.

- View a Payment
- View a Bill
- View an Adjustment

View a Payment

To answer customer inquiries, or determine how to best service a customer, you might need to view detailed information about a specific payment that was made. You can access payment history details from the Billing tab.

To view payment details:

1. Access the Customer Information Hub for the billing account you want to review and select the Billing tab.
2. If necessary, use the filters above the table in the Financial History section to locate the payment you want to view.
3. Select the hyperlink in the **Transaction Type** column for the payment record.
4. The system displays the **View Payment Summary** drawer on the right side of the screen, which includes:
    - Tender
    - Payment ID
    - Payment Amount
    - Payment Status
    - Payment Date
    - Payment Method

Copyright © 2026, Oracle and/or its affiliates.

- Payment Segment Distributions

5. Optionally, you can navigate to the Payment page by selecting the hyperlinked **payment ID**. Use your browser's **Back** button to return to the **View Payment Summary** drawer.
6. To close the drawer, select the **X** at the top.

View a Bill

To answer customer inquiries, or determine how to best service a customer, you might need to view detailed information about a specific bill that was issued to the customer. You can access bill history details from the Billing tab.

To view bill details:

1. Access the Customer Information Hub for the billing account you want to review and select the Billing tab.
2. If necessary, use the filters above the table in the Financial History section to locate the bill you want to view.
3. Select the hyperlink in the **Transaction Type** column for the bill record.
4. The system displays the **View Bill Summary** drawer on the right side of the screen, which includes:
   - Bill Issued Date and Amount
   - Bill ID
   - Bill Due Date
   - Bill Period
   - Number of Days in Period
   - Bill Summary section, which includes:
     - Previous Balance
     - Total Payments
     - Total Adjustments
     - Total Current Billing Charges
     - Corrections
     - Ending Balance This Period
   - Bill Segments
5. To view details about a bill segment, select a Bill Segment card. Select the **X** at the top to close the bill segment detail.
6. Optionally, you can navigate to the Bill page by selecting the hyperlinked **bill ID** at the top of the screen. Use your browser's **Back** button to return to the **View Bill Summary** drawer.
7. To close the drawer, select the **X** at the top.

View an Adjustment

To answer customer inquiries, or determine how to best service a customer, you might need to view detailed information about a specific adjustment that was mad to a bill. You can access adjustment history details from the Billing tab.

To view adjustment details:

Copyright © 2026, Oracle and/or its affiliates.

1. Access the Customer Information Hub for the billing account you want to review and select the Billing tab.
2. If necessary, use the filters above the table in the Financial History section to locate the adjustment you want to view.
3. Select the hyperlink in the **Transaction Type** column for the adjustment record.
4. The system displays the **View Adjustment Summary** drawer on the right side of the screen, which includes:
    - Adjustment Status and Title
    - Adjustment ID
    - Arrears Date
    - Bill ID
    - Amount
    - Status
    - Last Bill Issued
    - Adjustment Type
    - Premise
    - Service Agreement
    - Financial Details
    - Calculation Details
5. Optionally, to view additional details, select the hyperlinks in the **Bill ID**, **Premise**, **Service Agreement**, or **Financial Transaction** fields to navigate to the detailed page for that record. Use your browser's **Back** button to return to the **View Adjustment Summary** drawer.
6. To close the drawer, select the **X** at the top.

Billing Highlights

The Billing Highlights section of the Billing tab provides key data points about the customer's current billing situation, including whether they are enrolled in certain billing programs. Billing highlights are configured by the utility, and can include items such as:

- **Auto Pay Enrollment:** This item identifies whether the customer is enrolled in automatic payments, and provides a button to either enroll the customer if they are not currently enrolled, or to update current enrollment information. Both options open the **Auto Pay Enrollment** drawer, where you can manage your customer's enrollment.
- **Budget Billing Enrollment:** This item identifies whether the customer is enrolled in budget billing, and provides a button to either enroll the customer if they are not currently enrolled, or to update current enrollment information. Both buttons open the **Budget Billing** drawer, where you can manage your customer's enrollment.
- **Paperless Billing (E-Bill) Enrollment:** This item identifies whether the customer is enrolled in paperless billing, and provides a button to either enroll the customer in the program if they are not currently enrolled, or to view current enrollment information. Both buttons open the **Paperless Billing** drawer, where you can manage your customer's enrollment.

Utilities can determine which insights are available in this section of the form by setting up insights and insight groups.

Copyright © 2026, Oracle and/or its affiliates.

If your utility is using Oracle Fusion Knowledge Management, a book icon appears in the header of the zone. When you select the icon, a drawer opens with a pre-confgured knowledge article that can help agents answer calls about billing programs.

- Manage Paperless Billing
- Manage Budget Billing
- Manage Auto Pay

Manage Paperless Billing

You can enroll and unenroll customers from paperless billing, also known as E-Bill, using the Billing Highlights section of the Billing tab.

> **Note**
>
> The names of billing programs can vary by utility.

Enrolling a Customer in Paperless Billing

To enroll a customer in paperless billing:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Enroll** button next to the **Customer is not on e-bill** option. **Note:** If the customer is already enrolled, the **Enroll** button is replaced by the **Update** button.
3. The system opens the **Paperless Billing** drawer.
4. Optionally, select the **Go to Account** link to navigate to the Account page. To return to the Hub, use the browser's **Back** button.
5. Complete the following fields and then select **Save**:
   - Bill Route Type
   - Email Contact Type - This field is displayed only if the user does not already have an email address.
   - Primary Email - This field is displayed only if the user does have an email address.
   - Add Primary Email - Use this field to change the email address for paperless billing.
6. The system closes the drawer, and the Billing Highlights section is updated to reflect the new enrollment status.

Unenrolling a Customer from Paperless Billing

To unenroll a customer from paperless billing:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Update** button next to the **Customer is on e-bill** option. **Note:** If the customer is not enrolled, the **Update** button is replaced by the **Enroll** button.
3. The system opens the **Paperless Billing** drawer.
4. Update the **Bill Route Type** field with a non-email bill route type, such as postal, and click **Save**.

Copyright © 2026, Oracle and/or its affiliates.

5. The system closes the drawer, and the Billing Highlights section is updated to reflect the new enrollment status.

Manage Budget Billing

You can enroll and unenroll customers in budget billing programs using the Billing Highlights section of the Billing tab.

> **Note**
>
> The names of billing programs can vary by utility.

Enrolling a Customer in Budget Billing

To enroll a customer in budget billing:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Enroll** button next to the **Customer is not on budget billing** option. **Note:** If the customer is already enrolled, the **Enroll** button is replaced by the **Update** button.
3. The system opens the **Budget Billing** drawer.
4. Optionally, select the **Go to Account Budget** link to navigate to the Account Budget page. To return to the Hub, use the browser's **Back** button.
5. Select a Budget Plan and then specify the date the plan should begin in the **New Budget Date** field.
6. The system populates the budget fields, which you can review with the customer.
7. Select **Save**.
8. The system closes the drawer and the new enrollment status is reflected in the Billing Highlights section.

Unenrolling a Customer from Budget Billing

To unenroll a customer from Budget billing:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Update** button next to the **Customer is on budget billing** option. **Note:** If the customer is not enrolled, the **Update** button is replaced by the **Enroll** button.
3. The system opens the **Budget Billing Update** drawer.
4. Select the **Cancel Budget** button.
5. The system closes the drawer and the new enrollment status is reflected in the Billing Highlights section.

Manage Auto Pay

You can enroll and unenroll customers from auto pay using the Billing Highlights section of the Billing tab.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The names of billing programs can vary by utility.

Enrolling a Customer in Auto Pay

To enroll a customer in auto pay:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Enroll** button next to the **Customer is not on auto pay** option. **Note:** If the customer is already enrolled, the **Enroll** button is replaced by the **Update** button.
3. The system opens the **Auto Pay Enrollment** drawer.
4. Optionally, select the **Go to Account Auto Pay** link to navigate to the Account Auto Pay page. To return to the Hub, use the browser's **Back** button.
5. Complete the following fields and then select **Save**:
   - Auto Pay Source
   - External Account ID
   - Expires On
   - Auto Pay Method
   - Name
   - Maximum Withdrawal Amount
   - Start Date
   - End Date
   - Comments
6. The system closes the drawer and the Billing Highlights section is updated with the new enrollment status.

Unenrolling a Customer from Auto Pay

To unenroll a customer from auto pay:

1. Access the Hub for the billing account you want to review and select the Billing tab.
2. In the Billing Highlights section of the form, select the **Update** button next to the **Customer is on auto pay** option. **Note:** If the customer is not enrolled, the **Update** button is replaced by the **Enroll** button.
3. The system opens the **Auto Pay Enrollment** drawer.
4. Specify an end date and then select **Save**.
5. The system closes the drawer and the Billing Highlights section is updated with the new enrollment status.

Billing History

The billing history charts in the Billing tab provide multiple views of the customer's billing history. The small chart at the bottom right of the Billing tab provides an at-a-glance view of up to the last six months of billing history. Each month has two bars:

Copyright © 2026, Oracle and/or its affiliates.

- **Top bar:** This bar can be segmented to show the billed amount for each service on the account.
- **Bottom bar:** This bar represents the total ending balance for the month.

When you select the **View more details** link below this chart, the Billing History drawer opens, displaying a larger chart. In this view, the two bars are side by side. You can navigate through the available data using the date arrows above the chart.

In either view, you can select a bar to view details about the selected bill. The system opens the Bill Summary drawer. See View a Bill for additional information.

Using the Chart in the Billing History Section

To use this chart:

1. From the Customer Intelligence Hub, select the Billing tab.
2. In the Billing History section of the Billing tab, scroll down to view up to six months of billing history.
3. Place your cursor over a bar to view dates, service types, and billing amount details.
4. Select a bar to view billing details. The system opens the Bill drawer. For additional information about viewing billing details, see View a Bill.
5. Select the **X** at the top of the drawer to close it and return to the Billing tab.

Using the Chart in the Billing History Drawer

To use this chart:

1. From the Customer Intelligence Hub, select the Billing tab.
2. In the Billing History section of the Billing tab, select the **View more details** link below the chart.
3. In the **Billing History** drawer, use the **<** and **>** buttons above the chart to view the desired date range.
4. Place your cursor over a bar to view dates, service types, and billing amount details.
5. Select a bar to view billing details. The system opens the Bill drawer. For additional information about viewing billing details, see View a Bill.
6. Click the **X** at the top of the drawer to close it and return to the Billing tab.

Using the Usage Tab

You use the Usage tab to view historical usage information for the selected billing account. The tab header shows the number of active service types associated with the account. The work area includes these main sections:

- **Filters:** Use the usage filters to select the service type and service agreement you want to view usage data for. The data in the rest of the form changes based on your selections here.
- **Usage Report:** Use this section of the Usage tab to view a list of usage details for the specified service type and service contract. You can view up to three years of data at a time, and you can change your date range as needed.
- **Usage Highlights:** Use this section of the Usage tab to get an at-a-glance view of the main highlights related to the selected service contract, including:
  - Average Daily Usage

Copyright © 2026, Oracle and/or its affiliates.

  - Average Bill Usage
  - Highest Bill in the Last 12 Months
  - Lowest Bill in the Last 12 Months
  - Usage for the Last 12 Months
- **Usage History:** Use this section of the Usage tab to view a chart of the usage data for the selected service type. By default, this chart shows the last six months of data. Select the **View more details** link to open the **Usage History Details** drawer and view a larger chart with additional details.
- Usage Filters
- Usage Report
- Usage History Charts

Usage Filters

Data on the Usage tab is specific to a single service agreement for the selected account. You use the filters, available at the top of the tab, to specify the service type and related service agreement you want to view. After you set these filters and perform a search, all data on the Usage tab is updated to reflect your selections.

Select a Service Agreement

To select a service contract:

1. From the Customer Intelligence Hub, select the Usage tab.
2. By default, the system displays the first service contract associated with the billing account.
3. To change the display, select the **Expand Filters** button, and enter your search criteria in these fields:
   - Service Type
   - Service Agreement
   - Date Range
   - Start Date and End Date (if applicable)
4. Select **Search**.
5. The data on the screen changes to reflect your selection.

Usage Report

To answer customer inquiries, or determine how to best serve a customer, you might need to view detailed information about usage data for a specific service agreement. You can access usage history details from the Usage tab. After you select a service type and service agreement, the related bill segments are displayed in the **Usage History Report** section. You can select the hyperlink in the **Bill Amount** column of any record to view the details associated with the bill segment.

View Usage History Details by Bill Segment

To view usage history details:

1. Access the Customer Intelligence Hub for the billing account you want to review and select the Usage tab.

Copyright © 2026, Oracle and/or its affiliates.

2. Use the filters to select the service type and service agreement you want to view, along with any additional filters, and select **Search**.
3. In the Usage History table, select a value in the **Bill Amount** field to view the bill segment record.
4. The system displays the **View Bill Segment Summary** drawer on the right side of the screen.
5. To view additional details about the bill segment, select the **ID** hyperlink at the top of the drawer.
6. The system navigates you to the Bill Segment page. To return to the Customer Intelligence Hub, use your browser's **Back** button.

Usage History Charts

The usage history charts in the Usage tab provide multiple views of the customer's usage history. The small chart at the bottom right of the tab provides an at-a-glance view of the last six months of usage history. You can hover over the bars to view additional details.

When you select the **View more** link below this chart, the **Usage History Detail** drawer opens, displaying a larger chart. In this view, you can navigate through the available data using the date arrows above the chart, and change the unit of measure or level of detail (if available).

When in the **Usage History Details** drawer, you can change your view by:

- Selecting a different unit of measure. For example, select Kilowatt Hours.
- Selecting a different usage interval. For example, select View Bill Period.
- Using the **<** and **>** buttons to change the time period you are viewing.
- Clicking on a bar to view more granular detail, if it is available.

You can also view the service agreement details that are displayed above the chart. You can place your cursor over a bar in the chart to view additional details in a tooltip, or you can select a bar to change the chart view to a more granular view.

For example, if you place your cursor over a bar that represents a bill period, you can select that bar to view a chart that includes a bar for each day in that bill period.

> **Note**
>
> You can only access more granular detail in the chart if the detailed data is available. For example, if hourly data is not available, you can't select a bar that represents a day to display an hourly chart.

Using the Usage History Charts

To use usage charts:

1. From the Customer Intelligence Hub, select the Usage tab.
2. Using filters, select the service agreement you want to view and select **Search**.
3. The system displays the last six months of usage data in the **Usage History** chart at the bottom right of the work area.
4. Complete one of these actions:
   - Scroll to view the last 6 months of data in the chart.

Copyright © 2026, Oracle and/or its affiliates.

- Select the **View more details** link to open the **Usage History Details** drawer.

5. In the Usage History Details drawer, you can change your view by:
   - Selecting a different unit of measure. For example, select Kilowatt Hours.
   - Selecting a different usage interval. For example, select View Bill Period.
   - Using the **<** and **>** buttons to change the time period you are viewing.
   - Selecting a bar to view more granular detail (if available). Note that you can tell if the bar is clickable by placing your cursor over it. The tooltip that displays will include "Click to view <interval>" if more detailed data is available.
6. To view additional information about the service agreement, you can:
   - Select the hyperlink above the chart to access the Service Agreement page. Use your browser's **Back** button to return to the Hub,
   - Select the icon next to the service agreement name to see a list of additional pages you can navigate to. After selecting a page, use your browser's **Back** button to return to the Hub.
7. Select the **X** at the top of the drawer to close it and return to the Usage tab.

Using the Cases Tab

The Cases tab provides a list of cases that are associated with the selected billing account. You can access the case detail by selecting the hyperlink in the **Case ID** column, which opens the **View Case Summary** drawer, where you can:

- View the case title, status, comments, and additional details. The fields that appear are determined by the case type, and are customizable by the utility.
- Use the hyperlinks in the drawer to access the Case, Person, Premise, Account, Contact, or Characteristic pages.

If you navigate away from the Cases tab, use your browser's **Back** button to return to the Hub.

> **Note**
>
> If your system is configured to integrate with Oracle Fusion Service, related service requests appear as hyperlinks in the **SR Number** column, which you can select to view the service request details. If your system is not integrated with Oracle Fusion, the service request number, if available, displays as text only.

For more information about using cases, see Case Management.

- View a Case

View a Case

To answer customer inquiries, or determine how to best service a customer, you might need to view detailed information about a specific case or related service request. You can access case details from the Cases tab.

To view case details:

1. Access the Customer Information Hub for the billing account you want to review and select the Cases tab.

Copyright © 2026, Oracle and/or its affiliates.

2. Select the hyperlink in the **Case ID** column for the case you want to review.
3. The system displays the **View Case Summary** drawer on the right side of the screen, which might include fields such as:
   - Case Details
   - Contact Information
   - Characteristics
   - Bill Segments

   > **Note**
   >
   > The fields that are displayed are determined by the case type.

4. The drawer can also include hyperlinks that let you to navigate outside of the Customer Intelligence Hub and see more detail about entities, which might include:
   - Case
   - Person
   - Premise
   - Account
   - SR Number - This field displays as a hyperlink only if your system is set up to integrate with Oracle Fusion Service. Selecting the service request number opens Oracle Fusion in a new tab.
5. If you navigated away from the drawer, either close the new tab or use your browser's **Back** button to return to the Hub.

Maintaining The V

As described in Understanding The "V", the "V" objects are those that form the core of your business processes. In this section, we describe the pages that maintain these objects.

> **Note**
>
> **Start/Stop Also Maintains The "V".** The Start/Stop Pages also maintain many of the objects in the "V". For example, service agreements are created behind-the-scenes when you start service for a customer. This means you won't have to use most of the pages described in this section as part of your day-to-day activities. Rather, they exist in case you need to override what the system automatically set up for you.

- Maintaining Persons
- Maintaining Accounts
- Account Management Portal
- Account - SAs For A Debt Class
- Using The Account/Person Replicator

Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Service Agreements
- Maintaining SA Relationships
- Service Provider SA Relationship
- Maintaining Premises
- Maintaining Service Points

Maintaining Persons

On the person page, you define demographic information about your customers and every other individual or business with which your company has contact. The topics in this section describe the person page.

> **NOT_SUPPORTED**
>
> For more information about persons and their place in the "V", refer to Understanding The "V". For more information about how most persons are added, refer to How To Add A New Customer From Control Central. For more information about how a person is required to set up a customer, refer to Customer Overview.

- Person - Main Information
- Person - Correspondence Information
- Person - Characteristics
- Person - Persons
- Person - Person Portal

Person - Main Information

The Main page contains core person information like names, telephone numbers, and forms of identification. Open this page using **Menu**, and then **Customer Information**, and then **Person**, and then **Search.**

Description of Page

Person Information contains basic information about the person. These values only appear after the person exists in the database. The **ID** is a system-assigned random number that stays with a person for the life of the system.

> **Note**
>
> **Formatting is performed by a plug-in.** The format of **Person Information** is controlled by a plug-in algorithm on the installation record. Refer to the base package's person information algorithm (PERS-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Copyright © 2026, Oracle and/or its affiliates.

The following information may be recorded about a person:

Person/Business indicates if the entity is a person or a business. Valid values are: Person, Business. This value controls how the person's primary name is validated.

Life Support / Sensitive Load indicates if the person has life support or sensitive load equipment. Valid values are: LS/SL (i.e., the person has life support / sensitive load equipment), None. If the customer has LS/SL equipment, make a note of the type of equipment in the **Life Support / Sensitive Load Notes**.

A premise can also have life support / sensitive load information. If the equipment is physically linked to the premise (e.g., a hospital has life support equipment), you should NOT specify the life support information on the person. Rather, transfer to Premise - Misc and specify life support information on the premise.

> **Note**
>
> **Alert and C&C.** If the base package's C1-LSSL-PER algorithm is enabled on the installation record, and life support / sensitive load information is specified, an alert will appear when the person is displayed on control central. In addition, if a person has life support or sensitive load equipment, it is possible for a different credit & collection severance process to be kicked off if the person's account associated has overdue debt. Refer to Designing Your Severance Procedures for more information.

Person Names are used by Control Central to look for accounts and persons. In addition, a person's primary name is the addressee on the person's bill unless overridden by the Override Mailing Name (maintained on the **Misc** tab). The following fields display:

- Use **Name Type** to indicate if the name is an Alias, Alternate Representation, Doing Business As, Legal, or Primary name. Note, for new persons, a value of Primary is defaulted. The values for the name type field are customizable using the Lookup table. This field name is NAME_TYPE_FLG
- Use **Person Name** to define the person's name. Note well, the name is case sensitive.

> **Note**
>
> **Alternate representations of a person's name.** You would use an Alternate Representation for a person's name when you have an alternate ways to define the person's primary name. Alternate representations are typically used in countries that use multiple character sets (e.g., the Primary name is entered in Chinese, the Alternate Representation is entered in English). When a person has an alternate name, both the main and alternate names can be used to search for a person. The Alternate Representation **Name Type** only appears if you have enabled alternate names on the installation record. Refer to the description of the Alternate Representation field under Installation - Main for more information.
>
> **Validation and formatting are performed by plug-ins.** The validation that is applied to **Person Name** (e.g., a comma separating the last and first name - Smith,Patricia) is controlled by a plug-in algorithm on the installation record. More than one algorithm can be plugged into this plug-in spot. Algorithms plugged into this plug-in spot can also perform formatting such as removing special characters that your implementation's business rules do not allow in person names. Other algorithms can be configured to remove or replace text.

Copyright © 2026, Oracle and/or its affiliates.

- The **Do Not Format** switch can be used to bypass any person name formatting done by the algorithms described above. Validation such as requiring name separator is still applied, but no further formatting is applied. Setting the switch only applies while the record is being saved and is then reset. The switch is reset because the formatting is only applied when the record is subsequently changed. This switch is only shown if the Enable Person Name Format Bypass **Option Type** on the Customer Information Options Feature Configuration has been enabled

> **Note**
>
> The Person Phone grid appears when the Legacy Person Phone and Email **Option Type** on the Customer Information Options Feature Configuration has been enabled.

Person Phone numbers are used by Control Central to look for accounts and persons. The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business, ...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

Person Contacts provide a central location to capture contact information such as phone numbers and email addresses. Any contact information can be captured. Person Contacts are used on contact preferences for notifications.

While person contacts can be used in lieu of a person phone and all system processes such as start/stop and orders support person contacts, your implementation may still opt to use person phone. Functionality has been provided to create and maintain person contacts from person phone records. When a person contact type is associated with a phone type, the person contact cannot be deleted nor can the fields maintained via person phone be edited. Refer to Maintaining Person Contact via Person Phone for more information.

Person contacts can also be edited or created from the Account Contact Information Zone. *(See Account - Account Portal for more information.)*

- **Contact Routing** represents a general grouping of person contacts such as email or phone. It is displayed on this page, and is defined on the person contact type.

> **Note**
>
> Contact Routings of type whose Contact Routing type is *Self Service User* can be viewed here and not elsewhere in the system. Furthermore, they can be viewed, but not edited, added, or deleted.

Copyright © 2026, Oracle and/or its affiliates.

- **Person Contact Type** indicates the type of contact such as, home phone, cell phone, or work email.
- **Contact Information** contains the specific contact information such as, the phone number or email address. Enter the contact information in the format described.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to **Contact Information** is controlled by the algorithm that is plugged in on the respective person contact type (refer to Setting Up Person Contact Types for more information). If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the contact information. (This field is only active when the contact's contact routing is phone or fax.)
- The **Primary** switch must be set for exactly one person contact for each contact routing.
- **Do Not Disturb (From)** and **Do Not Disturb (To)** indicate the times when a person contact should not be used. (**Note:** these fields are active when the Person Contact Type's Allow Do Not Disturb information is set to *Optional*.
- **Contact Status** indicates the option contact's status. The initial status can be defaulted from the Person Contact Type. The status can be manually maintained, as appropriate. (**Note:** This dropdown is enabled when the Person Contact Type allows Status to be used. It is also disabled when the person contact type's status is configured to be controlled by a process such as an Opt-In process.) See Person Contact Status Can Be Controlled By A Process for more information.
- **Opt-in** provides functionality to initiate an opt-in process for person contact types configured to support opt-in. This is used to obtain permission to use the person contact type to send notifications. (Note: The functionality on this page is not the primary way of initiating an opt-in process.) See Person Contact Status Can Be Controlled By A Process for more information about Opt-In Processes.
- **Active/Inactive** defaults to Active. Most contact preferences will remain active. When the contact information changes, the information on the existing contact should be updated.

> **Warning**
>
> It is strongly recommended that existing contacts are updated whenever possible such as when a person's phone number or email address changes. When a person no longer has a contact such as an alternate phone number, the contact should be deleted. Inactive should only be used when the person contact cannot be deleted because it is referenced on another object such as a contact preference.

- **Contact Nickname** *(display only)* if you implementation is using self-service, users can assign nicknames to contacts and they are shown here.

A **Person's ID's** have several uses:

- They are used by Control Central when you look for a customer / premise based on their ID.
- They are used to highlight potential duplicate persons.

Copyright © 2026, Oracle and/or its affiliates.

- Control Central displays a person's **primary** identification in the search results area to help a user identify the customer when multiple customers match the search criteria.

> **Note**
>
> **Person ID may be required or optional.** The person ID usage flag on the installation record indicates whether at least one id for a person is required or optional.

> **Note**
>
> The system supports encryption for Person Identifier. If your implementation has configured the system to encrypt Person Identifier, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information.

The following fields are used to define a person's ID(s).

- Turn on **Primary ID** for the piece of **ID** that is the primary means of identifying the customer.
- Indicate the **ID Type**. The **ID Type** defaults from the Installation Record based on the **Person Type** (Person versus Business).
- Enter the identification number in the adjacent fields. Please note that if the **ID Number** should be formatted (e.g., dashes in an American social security number), you do not have to enter the dashes. Rather, you can enter the information as a contiguous value and the system will format this for you. The format is shown in the adjacent **Identifier Format** column.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to an **ID Number** (e.g., dashes in an American social security number) is controlled by the algorithm that is plugged in on the respective ID Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

If you want to add an account for the person when you save the person information, turn on **Add Account and Start Service**. When this switch is on, you must also define the new account's **Customer Class**. If this information is specified, the system creates a new account and links the person to the account using the relationship type from the installation options. In addition, you will be transferred to Start / Stop where you can start service for the newly added account.

If you have set the Installation Option CIS Division Control flag, you must first define the new account's **CIS Division.** The values are restricted to the user's CIS Division(s). See Users and CIS Division for more information. The value is defaulted to user's primary CIS Division. The new account's Customer Class must then be defined.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Default note.** The account's customer class defaults from the Installation Record based on the **Person Type** (Person or Business). If shown, the customer class defaults first from the CIS Division and then from the Installation Record, each based on the **Person Type** (Person or Business)

A message appears in the lower right corner of the page that describes the number of accounts the person has. You can click the adjacent go to button to transfer to Control Central - Customer Information on which all of a person's accounts appear.

Person - Correspondence Information

This page contains information that may be used to address bills, letters, quotes and statements. Use **Menu**, and then **Customer Information**, and then **Person**, and then **Search** and navigate to the **Correspondence Info** page.

Description of Page

If the person does not want their primary name (defined on the Main page) used on bills, letters, quotes and / or statements, specify the desired name in **Override Mailing Name 1, 2, and 3**.

Specify the **Override Mailing Address** fields if the person wants their bills, letters, quotes, and / or statements sent to an address other than their service addresses (for example, a post office box). In addition, if the person's account doesn't already indicate that the person's mailing address should be used, you must update this person's account(s). This information resides in the Address Source field on Account - Person.

> **Note**
>
> Both the Start/Stop and Order transactions allow you to update the person's **Override Mailing Address** and their account's address source using a single transaction.
>
> **Address isn't everything.** In addition to defining the person's **Override Mailing Address**, there may be additional tasks you must perform in order to route information to this address. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.

If you enter an **Override Mailing Address**:

- The **Country** defaults from your installation options.
- The address constituents may differ depending on the **Country**. Refer to Defining Countries for more information on the address constituents.
- If you have set up postal defaults, the system will default the address constituents when you tab out of the postal code.
- The Validate button on the person page will only be available if you have set the Allow Address Validation option type to Y on the General System Configuration feature configuration. Once clicked, the Address Validation Script on the General System

Copyright © 2026, Oracle and/or its affiliates.

Configuration feature configuration will be invoked to validate the entered address. Refer to Implementing Address Validation for more information.

Specify the customer's **Email Address** if you communicate with the customer via Email. In addition to defining the person's **Email Address**, you're correspondence routing software must support sending the information via Email. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.

> **Note**
>
> The Email Address field appears when Legacy Person Phone and Email **Option Type** on the Customer Information Options Feature Configuration has been enabled.

Define the **Language** in which the person prefers their bills and correspondence printed.

> **Note**
>
> **Default note.** The person's language defaults from Installation Options - Person.

> **NOT_SUPPORTED**
>
> Refer to Customer Language for more information on options for supporting multiple languages for your customers.

Define the person's **Time Zone** preference if it is different from the installation's time zone. Refer to Installation Options – Framework, Main Page for more information. The time zone appears as part of the display anywhere the Do Not Disturb start and end information is displayed; for example, in the Person Portal or Account Portal.

> **Note**
>
> **Default Note.** The Time Zone defaults to blank, which indicates that the installation's time zone will be used.

The **Seasonal Addresses** scroll is only used if the person wants their correspondence sent to an alternate address during predefined periods. For example, the customer may want their bills sent to their vacation cottage during the summer. Please be aware of the following:

- Seasonal addresses will only be used if the customer's bills, statements, quotes, and / or letters are routed via the postal service. For example, if you route bills to the customer via Email, the seasonal address will never be used to route bills to the customer. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.
- You don't have to specify a seasonal address for every part of the year. For example, if the customer wants their bills sent to their service address except during the summer, you need only enter a seasonal address for the summer.

Copyright © 2026, Oracle and/or its affiliates.

- You can enter a seasonal address with or without an **Override Mailing Address**. If an **Override Mailing Address** is not specified, the person's correspondence will be addressed as per the instructions on the person's account(s). These instructions reside in the Address Source field on Account - Person.
- You can enter multiple seasonal addresses if the customer so desires.
- The **Seasonal** period is defined in the two adjacent fields. The first field contains the day and month when the season starts; the second field contains the day and month when the season ends. The day and month should be entered in the format defined in your display profile.
- The **Status** of the address must be Active. You can set the status to Inactive if you want a seasonal address ignored (alternately, you can just remove the seasonal address).
- The **Country** defaults from your installation options. The address constituents may differ depending on the **Country**. Refer to Defining Countries for more information on the address constituents.
- If you have set up postal defaults, the system will default the address constituents when you tab out of the postal code.

Person - Characteristics

The characteristics page contains information that describes miscellaneous information about the person. Use **Menu**, and then **Customer Information**, and then **Person**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

> **Note**
>
> You can only choose characteristic types defined as permissible on the person record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Define the date on which the characteristic becomes effective. The effective date defaults from the Installation Record.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Person - Persons

This page is used to define this person's relationship with other persons. For example, if the person being maintained is a conglomerate, you can define its subsidiaries on this page. Refer to How To Set Up Customer Hierarchies for more information about hierarchies.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You're defining "child relationships".** It's important to understand that the persons being defined on this page's grid should be thought of as "children" of the person on the top of the page.
>
> **Children may have children.** It's possible for one of the children to have children itself (for example, if you have a situation where a subsidiary of a conglomerate itself has subsidiaries). To define a child's children, simply display the child person on this page and then define its children.
>
> **Equal relationships.** It's possible to link persons where no hierarchy is implied by the relationship (e.g., spouses). There are two ways to do this: 1) you can nominate one spouse as the "parent" and the other as the "child" or 2) you can define the spousal relationship for both persons (i.e., you would define Robert Chopin as the husband of Jeanette Chopin, and Jeanette Chopin as the wife of Robert Chopin). If you choose the latter approach, a recursive relationship will exist.

> **Warning**
>
> If your organization enters multiple levels of person, we want to note that we do not prevent recursive relationships. This means that you could set up a nonsensical situation where person 2 is a child of person 1 and person 1 is a child of person 2.

Use **Menu**, and then **Customer Information**, and then **Person**, and then **Search** and navigate to the Persons page.

Description of Page

The grid contains the "children" associated with the "parent" person who is defined at the top of the page. The following fields display:

Child Person This is the unique identifier of the "child" person. This person's main name appears adjacent.

Child Information An informational message appears to highlight if the child person itself has children. You can click the corresponding go to button to view the child's children.

Relationship Type Indicate the type of relationship between the parent and the child person.

Start Date Indicate the date on which this relationship began. This field defaults from the Installation Record.

End Date If the relationship expires, indicate the date the relationship stops.

Financial Relationship Turn on this switch if the child person has account(s) and information about these account(s) should be displayed when the parent person's hierarchy is displayed in the Active Account Summary Zone.

Person - Person Portal

Select **Menu**, and then **Customer Information**, and then **Person**, and then **Search** and navigate to the **Person Portal** page to view additional information associated with a Person.

Copyright © 2026, Oracle and/or its affiliates.

Depending on your implementations requirements, you may define additional zones to appear on this Portal.

The contents of this section describe the zones that are available on this portal page.

General Information

Person ID, **Person Info** are displayed.

Customer Notification Preferences

The Customer Notification Preferences Zone displays notification preferences for the customer in context.

The following table lists the notification types that may appear in this zone and the page to which you will be transferred to if you click on a hyperlink (if applicable).

Type	What Is Displayed
Bill Routing for each account the person receives a copy of the bill. **Type hyperlink navigates you to the Account – Persons tab**	**Customer Information:** Customer Information hyperlink navigates you to the Account **Contact Method:** Bill Route Type specified on the account **Contact Information:** Contact details for the Bill Route Type defined for the account
PPB New Charge if the person has an active prepaid biller. Type hyperlink navigates you to the Prepaid Biller Task	**Customer Information:** Customer Information hyperlink navigates you to Prepaid Biller Task **Contact Method:** Notification Preference for how the customer will be notified each time a new charge is calculated (specified on the Prepaid Biller Task) **Contact Information:** Contact details for the new charge Notification Preference
PPB Payment Request if the person has an active prepaid biller. Type hyperlink navigates you to the Prepaid Biller Task	**Customer Information:** Customer Information hyperlink navigates you to Prepaid Biller Task **Contact Method:** Notification Preference for how the customer will be notified when their prepaid balance is less than the required minimum balance (specified on the Prepaid Biller Task) **Contact Information:** Contact details for the payment request Notification Preference
Quote Routing for each account where the person receives a copy of quotes. **Type hyperlink navigates you to the Account – Persons tab**	**Customer Information:** Customer Information hyperlink navigates you to the Account **Contact Method:** Quote Route Type specified on the account **Contact Information:** Contact details for the Quote Route Type defined for the account
Statement Routing for each statement the person will receive.	**Customer Information:** Customer Information hyperlink navigates you to Person

Copyright © 2026, Oracle and/or its affiliates.

Type	What Is Displayed
Type hyperlink navigates you to the Statement Construct – Main tab	**Contact Method:** Statement Route Type specified on the statement construct **Contact Information:** Contact details for the Statement Route Type defined for the statement construct for each statement

Self-Service Notification Preferences

This zone displays active notification preferences for the customer in context and is only applicable if your implementation integrates with Oracle Utilities Customer Self Service.

The applicable notification types (self-service task types) are defined on the Self-Service Integration master configuration.

The following details are displayed for each active notification task (self service task) and the page to which you will be transferred if you click on a hyperlink (if applicable):

- Notification Task Information – Summary information for each of the customer's active notification tasks. The hyperlink navigates you to the notification task (self-service task).
- Account Info – Details of the account linked to the notification task (self-service task). The hyperlink navigates you to the account.
- CSS Web User – End customer's User ID supplied by Oracle Utilities Customer Self Service when the customer registers for the notification task (self-service task).
- Delivery Types – List of customer's preferred communication channels specified on the notification task (self-service task).

Self Service Payment Options

This zone provides the ability to add, edit, and delete payment options used in customer self-service. This information is only applicable if your implementation integrates with Oracle Utilities Customer Self Service.

Refer to the zone help for more details.

Maintaining Accounts

The account record contains information that controls billing and collection processes. You should only need to use this page if you need to fine-tune the information that the system has set up by default because:

- Most accounts are created when you add a new person or when you complete an order for a new customer.
- Many fields are populated behind-the-scenes as part of the service activation process.

The topics in this section describe the pages on which account related information is maintained.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about accounts and their place in the "V", refer to Understanding The "V".

- Account - Main Information
- Account - Auto Pay
- Account - Person Information
- Account - Financial Balances
- Account - Bill Messages
- Account - C&C
- Account - Budget
- Account - Deposits
- Account - Characteristics
- Account - Alerts
- Account - Account Portal

Account - Main Information

The Main page contains core account information. Open this page using **Menu**, and then **Customer Information**, and then **Account**, and then **Search.**

Description of Page

The primary name of the **Account**'s main customer and the **Account ID** are displayed on every tab in this page. These values only appear after the account exists on the database. The **Account ID** is a system-assigned random number that stays with an account for life.

Note

Formatting may be performed by a plug-in. The basic information about an account that appears at the top of this page (and on many other pages in the system) may be formatted by a plug-in algorithm on the installation record. Refer to the base package's account information algorithm (C1–AC-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

A "check digit" is displayed adjacent to the account ID. This is for information purposes only, and is not needed to operate the system. The following points describe how the check digit is calculated for an Account ID equal to 0011883422

- Calculate the sum of the first and every alternate digit in the account id. A = 14 = 0+1+8+3+2
- Calculate the sum of the second and every alternate digit in the Account ID and multiply by 2. For example, B = 30 = (0 + 1 + 8 + 4 + 2) * 2
- Add A and B. For example, C = 44 = 14 + 30

Copyright © 2026, Oracle and/or its affiliates.

- Count the number of digits used to calculate B that are greater than 4. For example, D = 1 since only 8 is greater than 4.
- Multiply D by 9. For example, E = 9 = 1 x 9
- Subtract E from C. For example, F = 35 = 44 - 9
- Subtract the units position of F from 10 to find the check digit. For example, check digit = 5 = 10 - 5 (5 is subtracted from 10 because F = 35 and there the units position is 5).

> **Note**
>
> **Technical note.** The above is calculated in the common routine called CIPCACDN .

Set Up Date is the date the account was initially set up. This is purely informational.

Currency Code defines the currency in which the account's financial transactions are expressed. All rates and payments associated with this account must be denominated in this currency.

> **Note**
>
> **Default note.** The currency defaults from the Installation Record and may be overridden here.

Customer Class plays a part in:

- If and when a customer is subject to late payment charges.
- The account's default collection class and when the account debt monitor reviews an account. Refer to The Big Picture of Credit & Collections for more information about how and when an account's debt is reviewed.
- And several other functions. Refer to and Setting Up Customer Classes for more information.

> **Note**
>
> **Default note.** The customer class defaults from Installation Options - Account (Person Customer Class) and may be overridden at will. If you have set the Installation Option CIS Division Control flag, the customer class defaults first from the CIS Division and then from the Installation Record. When this option is set, the CIS Division is shown on the page before customer class. You may only select Customer Classes with the account's CIS Division.

CIS Division defines the jurisdiction that governs this account. You may only select **CIS Divisions** associated with the account's Customer Class. If you have set the Installation Option CIS Division Control flag, you need to choose the CIS Division before choosing the Customer Class. The values are restricted to the user's CIS Division(s). See Users and CIS Division ngfor more information. The value is defaulted to user's primary CIS Division. This field is updated behind the scenes every time a service agreement is activated (the system uses the **CIS Division** associated with the service agreement's SA type). If you have assigned

Copyright © 2026, Oracle and/or its affiliates.

a **CIS Division** and do not want the system to change it when a service agreement is activated, turn on **Protect CIS Division**. If the CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the Protect CIS Division is turned on and cannot be turned off.

> **Note**
>
> **CIS Division governs many functions.** An account's CIS Division impacts its subsequent bill due dates, late payment charge dates, credit & collections review dates, its default budget plan, the roles assigned to To Do entries, and the calendar of workdays. Refer to Setting Up CIS Divisions and Setting Up Customer Classes for more information.

Access Group controls which users are allowed to view and update this account's information (including bills, service agreements, payments, and premises). The system defaults this value from one of three sources, stopping after an access group is determined. The first is the Customer Class Control – Determine Access Group system event. The next is the access group defined directly on Customer Class Control. Lastly, it is defaulted from the user's default access group. Refer to The Big Picture of Row Security for a complete description of how account security is implemented in the system.

The optional Account **Management Group** controls the roles assigned to To Do entries associated with an account. Refer to Setting Up Account Management Groups for more information.

Enter a **Comment** to define unusual information about the account. If this field is populated, an alert will highlight such in the Alert Zone.

Bill Cycle controls when a bill is produced for an account. This field is updated behind the scenes every time a service agreement with a service point is activated. The route the system takes to do this update is as follows:

- It looks for all service points linked to the service agreement.
- It extracts the *service cycle* from one of the service points.
- Finally, it takes the bill cycle from the *service cycle*.

> **Note**
>
> The list of bill cycles is restricted to those without a CIS Division and those with a CIS Division that matches the account's CIS Division. When the account does not have a CIS Division, the list is not restricted.

While this is a little complicated, it makes sense because it keeps the bill cycle in sync with the *service cycle*.

If you have assigned a bill cycle and do not want the system to change the bill cycle when a service agreement is activated, turn on **Protect Bill Cycle**.

If you want to stall billing until after some future date, enter the date in **Bill After**. Note that the bill's cutoff date should be after this date, for the account to bill.

Mailing Premise ID defines the address on bills for persons who have their bill sent to the account's mailing premise. This field may be updated behind the scenes when a service

Copyright © 2026, Oracle and/or its affiliates.

agreement is activated. The word "may" is used because the system will only update an account's mailing premise if there are no active service agreements currently linked to the account. Refer to Account - Person Information for more information about how to define where a person has their correspondence and bills sent.

If you do not want the mailing premise to change when a service agreement is activated, turn on **Protect Mailing Premise**.

If a user should review the account's printed bills before they are sent to the customer in an on-going basis, enter this user's ID in **Bill Print Intercept**. This User ID will then be copied to future Bill - Bill Routing pages when they are created. If the user no longer desires to intercept future bills, their User ID can be removed from this field.

The balances shown in the scroll are the total amounts of debt belonging to every debt class linked to the account's service agreements. The following information is displayed:

- **Debt Class** is the debt class associated with the account's service agreement(s).
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt.

To see summary information about the service agreements that contribute to the financial totals click the go to button adjacent to the respective amount.

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

Account - Auto Pay

The Auto Pay page is used when an account pays their bills automatically (e.g., by direct debit or credit card).

> **NOT_SUPPORTED**
>
> Refer to How And `When Are Automatic Payments Created for more information.

Open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Auto Pay** page to maintain this information.

Description of Page

The **Account Auto Pay** scroll defines the bank account / credit card from which the customer's automatic payments are debited. Multiple auto-pay options may exist if the customer changes auto-pay options over time. The following fields display:

- **Start Date** is the date on which the automatic payment information comes into effect. The system creates an automatic payment for any bill produced for this account with a due date on/after this date and on/before the **End Date**. If **End Date** is not specified, this means the automatic payment option applies indefinitely. You need only specify an **End Date** if the customer wants to stop paying automatically.

Copyright © 2026, Oracle and/or its affiliates.

- **Auto Pay ID** is the unique, system-assigned identifier of the auto-pay record. This value is assigned after the information is saved on the database and may not be modified.
- Use **Auto Pay Source** to define the source of the funds used to satisfy the automatic payment request. For example, if a customer indicates that they want to use their checking account to pay their bill, you would specify the **Auto Pay Source Code** associated with their bank. The source's description and external source ID (e.g., bank routing number) are displayed adjacent.
- Use **Auto Pay Method** to specify whether you want the system to process automatic payments as Direct Debit or Payment Advice. This field is visible only if feature configuration is set up to support payment advice functionality. Refer to Payment Advices for more information on the payment advice functionality.
- Use **External Account ID** to define the customer's bank account / credit card number.

> **Note**
>
> – The system supports encryption for the External Account ID. If your implementation has configured the system to encrypt the External Account ID, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information
>
> – If the system is configured to perform Web Debit Account Validation and the specified bank account is being used for the first time, the Account Auto Pay record is not added / updated until bank account validation is complete. The Bank Account Auto Pay Setup service task manages the web debit account validation process and creates/update the Account Auto Pay record if the bank account is valid. If the account has open Bank Account Auto Pay Setup Task(s), a message appears above the Auto Pay scroll, to indicate that pending bank account validation(s) exist. The Alerts sidebar zone will also highlight the pending auto pay enrollment(s) if the Control Central alert algorithm is enabled. Click on the informational message on this page or on the sidebar alert, to view the service task. Note that if there are multiple open service tasks, the most recently updated service task (based on Status Update Date/Time) is shown. Refer to Validating Bank Accounts on Auto Pay Setup for more details.

- You will be required to define an **Expires On** date if the **Auto Pay Source Code** references a tender type that requires an expiration date (e.g., if the Auto Pay Source is a credit card company).
- Enter the **Name** of the customer as it appears in the financial institution's system. This name is routed to the financial institution.

> **Note**
>
> **Default note.** The **Name** will default to the primary name of the main customer linked to the account after you enter a **Start Date**. This defaulting is only possible for accounts that exist on the database.

- In some locales, customers can define a **Maximum Withdrawal Amount** to limit the amount of money that is automatically debited from their account. Refer to How To Implement Maximum Withdrawal Amounts for more information.

Copyright © 2026, Oracle and/or its affiliates.

- Use **Comments** to describe anything interesting / unusual about the automatic payment request.

Account - Person Information

The Account Persons page contains information about the person(s) linked to the account.

> **Note**
>
> Both customers and non-customers (e.g. the accounting service that handles the bills of a commercial customer) can be linked to an account.

When you add an account from Person - Main or from Order - Main the system automatically links the person to the account, so you typically won't have to open this page. However, if you need to link multiple persons to an account (e.g., because you need to send a duplicate of the account's bill to someone other than the main customer) or change information about a person on an account, open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Persons** page.

Customer Self Service and Account - Person Information:

It is possible that an implementation may have configured the system to integrate with Customer Self-Service. Account - Persons can be created through the self-service application and will use a reserved relationship type that is defined on the self-service master configuration.

Records with this relationship type can be thought of as “owned by the self-service application.” Changes to fields on the page are disabled because the record can be deleted by the self-service application.

You can change the relationship type to another value, which will indicate that the record is no longer owned by self-service. Self-service can no longer delete this record and changes are allowed to the fields on the page. Please note that the relationship cannot be changed back to the self-service designation

Description of Page

The **Account Persons** scroll contains one entry for every person linked to the account. The following fields display:

Person ID The unique identifier of the person linked to the account. The person's name is displayed adjacent.

Main Customer Turn on this switch if the person is the main customer on the account. Only one person on an account may be designated as the main customer.

The significance of the main customer is that its name will appear adjacent to the account ID throughout the system.

Financially Responsible Turn on this switch if the person is financially responsible for the account's debt.

This switch is on and gray if the person is the **Main Customer** because the main customer is always financially responsible. If multiple persons are linked to the account, you use this switch

Copyright © 2026, Oracle and/or its affiliates.

to indicate which ones are financially responsible versus those that are linked for other purposes.

Third Party Guarantor Turn on this switch if the person is a third party guarantor of the account's debt. This switch is off and gray if the person is the **Main Customer**.

Relationship Type Define the relationship between the person and the account. Refer to Setting Up Account Relationship Codes for information about setting up relationship types.

> **Note**
>
> **Default note.** When the system links the person to the account when you add an account from Person - Main, it defaults the relationship type from the Installation Record.

Self-Service Role indicates that the Account - Person record has a corresponding entry in the self service application and contains the value for the access in self-service. If there is no corresponding entry in self-service, this field will not display. If configured in the Self-Service Master Configuration, a hyperlink appears allowing direct access to self-service.

Prefix/Suffix and **Pfx/Sfx Name** Use these fields if you need to append additional information to a customer's name when correspondence or bills are sent to this person. For example, if the account has declared bankruptcy you might attach "Debtor In Possession" to the person's name. Use **Prefix/Suffix** to indicate if the **Pfx/Sfx Name** should be appended to the front or the back of the customer's name.

Receives Copy of Bill Turn on this switch if the person receives a copy of the account's bills. Turning off this switch grays the remaining fields. This switch is on and gray if the person is the **Main Customer** because the main customer always receives a copy of the bill.

> **NOT_SUPPORTED**
>
> Refer to The Source Of Bill Routing Information for more information about how the remaining information is used to format the address of a bill.

Bill Route Type Indicate how the bill is sent to the customer. This field's value defaults from the Installation Record.

- If the **bill route type** you select indicates that bills are routed via Fax, the person's fax number is displayed adjacent The fax number displayed is from a fax person contact associated with an active contact preference, the person’s primary fax person contact identified by the contact routing, or a person’s fax phone type identified by a flag on the phone type. Which is shown is dependent on how your system is configured.
- If the **bill route type** you select indicates that bills are routed via Email, the person's Email address is displayed adjacent. The email address displayed is from an email person contact associated with an active contact preference, the person’s primary email person contact identified by the contact routing, or a person’s email. Which is shown is dependent on how your system is configured.
- If the **Bill Route Type** you select indicates that bills are routed via the Postal service, you must choose an appropriate **Address Source** to define which address should be used.

Refer to Setting Up Bill Route Types for more information about bill route types.

Copyright © 2026, Oracle and/or its affiliates.

Bill Format Indicate if the customer should receive a Detailed or a Summary bill.

> **Note**
>
> The values for this field are customizable using the Lookup table. The values need to match the formats supported by your bill print software. This field name is BILL_FORMAT_FLG.

Number of Bill Copies Indicate how many copies of the bill the person receives.

Customer PO ID Indicate the purchase order ID the customer wants printed on their copy of the bill.

> **Note**
>
> **Receives Copy of Quote** and **Quote Route Type** only appear if the quotation Contract Management module is not turned off.

Receives Copy of Quote Turn on this switch if the person receives a copy of the account's quotes. Turning off this switch grays the remaining fields. This switch is on and gray if the person is the **Main Customer** because the main customer always receives a copy of the account's quotes.

> **NOT_SUPPORTED**
>
> Refer to Printing Quotes for more information about how the remaining information is used to format the address of a quote.

Quote Route Type Indicate how the quote is sent to the customer. This field's value defaults from the Installation Record.

- If the **quote route type** you select indicates that quotes are routed via Fax, the person's fax number is displayed adjacent. The fax number displayed is from a fax person contact associated with an active contact preference, the person's primary fax person contact identified by the contact routing, or a person's fax phone type identified by a flag on the phone type. Which is shown is dependent on how your system is configured.
- If the **quote route type** you select indicates that quotes are routed via Email, the email address displayed adjacent. The email address displayed is from an email person contact associated with an active contact preference, the person's primary email person contact identified by the contact routing, or a person's email. Which is shown is dependent on how your system is configured.
- If the **quote route type** you select indicates that quotes are routed via the Postal service, you must choose an appropriate **Address Source** to define which address should be used.

Refer to Setting Up Quote Route Types for more information about quote route types.

Receives Collection and Overdue Notices Turn on this switch if the person receives letters (i.e., notifications) initiated by collection, severance, write-off, overdue and cut events. These events will send letters to the main customer and any other person linked to the account that

Copyright © 2026, Oracle and/or its affiliates.

Receives Notifications. This switch is on and gray if the person is the **Main Customer** because the main customer always receives notifications. Refer to Printing Letters for more information about how letters are produced by the system.

Address Source If the **Bill Route Type** or the **Quote Route Type** you selected indicates that bills / quotes are routed via the postal service, you must choose the appropriate **Address Source**.

Choose Mailing Premise on Account if bills / quotes should be sent to the address associated with the Mailing Premise on the first page. This address is displayed adjacent.

Choose Person if bills / quotes should be sent to the person's mailing address. This address is displayed adjacent.

Choose Account Override if bills / quotes should be sent to an override address specified below. Typically, you would only choose this option if the person has multiple accounts and each account's bills / quotes should be sent to a different address. If you choose this option, the bottom of this page will become populated with fields that should be used to specify this override address.

Override address constituents If you selected an **Address Source** of Account Override, you must enter the address to which bills / quotes will be sent in the address constituents. The number and type of address constituents is based on the Country. These fields will be invisible for other **Address Sources**.

Note, the **Country** defaults from Installation Options - System. The City, County and State default based on the **Country** and **Postal** if a Postal Default exists for the postal code.

The Validate button on the account page will only be available if you have set the Allow Address Validation option type to Y on the General System Configuration feature configuration. Once clicked, the Address Validation Script on the General System Configuration feature configuration will be invoked to validate the entered address. Refer to Implementing Address Validation for more information.

Allow Communication Preference: Turn on this switch if person contacts can be used on communication preferences for this account. The switch is on and disabled when the person is the **Main Customer** because their contact information can always be used on contact preferences for notifications.

Account - Financial Balances

The financial balances page is an information-only page that subdivides an account's debt by debt class. The balances shown on this page are the total amounts of debt belonging to every debt class linked to the account's service agreements.

> **Note**
>
> **The source of debt class.** An account's debt comes from its service agreement(s). Every service agreement has an SA type. Every SA type references a debt class.

Open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Financial Balances** page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

The account's **Current Balance** displays at the top of the page.

The **Debt Class Balances** scroll contains one entry for every debt class linked to the accounts non-closed / non-cancelled service agreements. The following information is displayed for each debt class.

- **Debt Class** is the debt class associated with the account's service agreement(s). To see summary information about the service agreements that contribute to the debt class totals click the adjacent Go To button.
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt. Payoff balance is only displayed if it differs from current balance.

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- The **Debt Class Arrears** shows the age of all amounts that make up the current balance.
- The **Days Old** and **Amount** information shows the age of the debt class's debt.

> **Note**
>
> Information about debt balances by age is not applicable to open-item accounts.

Account - Bill Messages

The Account Message page contains information about the message(s) to appear on an account's bills. Both one-time and permanent messages may be defined. This information is used when the system assembles messages to appear on an account's bill. Refer to The Source of Bill Messages for more information about bill messages.

Open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Bill Messages** page.

Description of Page

The following fields are displayed:

Bill Message The message code and description of the message that appears on the customer's bill.

Bill Message Type Use Temporary to indicate the message should only be linked to the next bill produced for the account. Use Permanent if the message should appear on every bill. Note, a value of Temporary defaults.

Account - C&C

Copyright © 2026, Oracle and/or its affiliates.

The Credit Rating page contains the credit rating transactions that impact an account's credit rating and cash-only score. To view or modify these transactions, open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **C&C** page.

Description of Page

Collection Class controls how the account's debt is compared against collection criteria to determine if a collection process should be started. Refer to The Big Picture of Credit & Collections for more information about how and when an account's debt is reviewed.

> **Note**
>
> **Default note.** The **Collection Class** defaults from the account's customer class when the account is first added. It may be overridden at will.

If you need to prevent an account from being reviewed by credit and collections processes (i.e., the account debt monitor and the write-off monitor), use **Postpone Credit Review Until** to define the future date when these processes can again review the account's debt.

Last Credit Review Date is the date when the account's debt was last reviewed by the account debt monitor (ADM).

> **Note**
>
> **Forced ADM review.** If you want the account debt monitor to review an account's debt the next time it runs, simply change the **Credit Review Date** to a date in the distant past (e.g., 1/1/1900). The next time the ADM background process runs with the Include Minimum Days Review (Y/N) set to Y, it'll see that a great deal of time has passed since the last time it reviewed the account's debt and review it (and then set the **Credit Review Date** accordingly).

The account's **Current Credit Rating** is displayed. For an explanation of how this number is derived, see How is an account's credit rating calculated?.

The account's **Current Cash-Only Score** is displayed. For an explanation of how this number is derived, see How is an account's cash-only score calculated?.

The **Credit Rating History** scroll contains one entry for each credit rating history record associated with the account. The following fields display:

Start Date This is the first date the credit rating transaction affects the account's credit rating and cash-only score.

End Date This is the last date the credit rating transaction affects the account's credit rating and cash-only score.

The Created On message is formatted using an algorithm on the Installation Options.

Affect Credit Rating By This is the effect of the credit rating transaction on the account's credit score. This should be a *negative* number because the *lower* the score the worse the credit rating.

An account's credit rating is equal to the start credit rating amount defined on the installation record plus the sum of credit rating demerits that are currently in effect. When an account's

Copyright © 2026, Oracle and/or its affiliates.

credit rating is less than the credit rating threshold defined on the installation record, the account's credit rating is displayed as an alert in the Alert Zone.

Affect Cash-Only Score By This is the effect of the credit rating transaction on the account's cash-only score. The *higher* the score, the greater the chance this account will be marked as cash-only. Therefore, a *positive* number should be used to make the score go up.

An account's cash-only score is equal to the start cash-only score defined on the installation record plus the sum of cash-only points that are currently in effect. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record, the account is flagged as cash-only in the Alert Zone and on the payment event page.

Comments Enter comments to clarify the reason(s) for the creation of the transaction.

- How are credit rating transactions created?
- How is credit rating calculated?
- What impact does credit rating have in the system?
- How is cash-only score calculated?
- What impact does cash-only score have in the system?

How are credit rating transactions created?

Credit rating transactions may be created as follows:

- Certain types of severance events add credit rating transactions. The number of credit rating and cash-only points is defined on the severance event type.
- Certain types of severance events add credit rating transactions. The number of credit rating and cash-only points is defined on the severance event type.
- Certain types of write-off events add credit rating transactions. The number of credit rating and cash-only points is defined on the write-off event type.
- If a payment tender is canceled with a cancellation reason that indicates the cancellation was due to non-sufficient funds, a credit rating transaction is created. The number of credit rating and cash-only points is defined on the payment cancellation reason code.
- You may configure your system to create a credit rating transaction when a pay plan is broken. To do this, you must plug in an appropriate Broken Algorithm on the pay plan type.
- A user may create credit rating transactions at their discretion.
- Algorithms may add credit rating transactions. The system provides sample overdue event and cut event activation algorithms that add credit rating transactions. Refer to C1-OE-CR-RT and C1-CE-CR-RT for more information. Some examples of algorithms your implementation may decide to introduce: the creation of a late payment charge. To do this you could introduce a new adjustment freeze algorithm.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Created By Flag**. The credit rating transaction includes a **Created By** flag that is customizable using the Lookup table. This field name is CR_RAT_CRE_BY_FLG. If your implementation uses algorithms to create credit rating history transactions, you may consider adding appropriate values to the Created By flag. Also note that the Created By flag includes the value Other , which may be used by your implementation specific algorithms if you do not want to add a new custom value.

How is credit rating calculated?

An account's credit rating is calculated by summing the Affect on Credit Rating from credit rating transactions effective on the current date. This value is added to the beginning credit rating defined on the installation record. The result is the customer's current credit rating.

> **Note**
>
> CIS Division can override the beginning credit rating. The system processes first check the account's CIS Division to see if the beginning credit rating has been specified on the CIS Division.

What impact does credit rating have in the system?

The following points describe the impact of an account's credit rating:

- An account's current credit rating is displayed in the Control Central Alert area when it's less than the threshold credit rating defined on the installation record.
- An account's credit rating may affect how the account's debt is collected. We used the word "may" because an account's credit rating will only affect collection criteria if you set up your collection processes to do this.

> **NOT_SUPPORTED**
>
> For more information, refer to Designing Your Collection Class Control Overrides.

How is cash-only score calculated?

An account's cash-only score is calculated by summing the Affect on Cash-Only Score from credit rating transactions effective on the current date. This value is added to the beginning cash-only score defined on the installation record. The result is the customer's current cash-only score.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> CIS Division can override the beginning cash-only score. The system processes first check the account's CIS Division to see if the beginning cash-only score has been specified on the CIS Division.

What impact does cash-only score have in the system?

If an account's current cash-only score exceeds the cash-only threshold on the installation record, and the base package's C1-CASH-ACCT algorithm is enabled on the installation record:

- The customer is flagged as cash-only in the Control Central Alert area.
- If a user attempts to add a payment whose tender type is a cash-equivalent, a warning is issued.

Account - Budget

The Budget page contains information about all service agreements linked to an account that are eligible to participate in a budget billing plan. Refer to Budget Billing for more information about how this information affects the account's bills.

> **Note**
>
> **You are actually changing service agreements.** When you change a service agreement's budget amount on this page, you are actually changing the service agreement's recurring charge amount. Refer to Service Agreement - Recurring Charge for more information.
>
> **Eligible service agreements.** It's important to be aware that some service agreements linked to an account may not be eligible for budget plans. For example, assume you have a customer with a deposit service agreement and one or more utility service agreements. This customer will be billed for their deposit regardless of the utility service agreements' budget amounts. You define whether a service agreement may participate in a budget plan on the service agreement's SA type.

Open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Budget** page to view this information.

Description of Page

The appearance and dialogue of this page differs slightly for an account with a small number of service agreements (100 or less) versus a large number of service agreements (more than 100).

- For an account with a small number of service agreements, the user may click the **Recommend Budget** button to generate a **New Budget** for each service agreement. At that point the service agreement's budget amount has not been updated. The user may review the calculated amounts; make desired changes and then click Save to update the budget amount for the service agreements.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> You must save the page in order for the budget changes to take effect.

- For an account with a large number of service agreements, the user may click the **Calculate and Apply New Budget** button. At that point the system will calculate the recommended budget and update the **Budget Amount** on all the eligible service agreements. To change the budget amount for one or more of the service agreements, the user navigates to the service agreement page using the drill down button adjacent to the service agreement in question.

Click **Cancel Budget** to have the system set the **New Budget** amount to zero for all the account's service agreements on budget.

> **Note**
>
> **Budget Recommendation.** Though this page lists all SAs eligible for a budget, the budget recommendation only calculates a budget amount for SAs whose start and end date (if any) includes the budget date.

Budget Plan controls how the account's debt is managed for budget-related purposes. Specifically, it controls how the recommended budget amount is calculated and how / when the customer's budget is periodically changed.

> **Note**
>
> **Default note.** Budget plan defaults based on the account's customer class / division combination.
>
> The list of budget plans is restricted to those without a CIS Division and those with a CIS Division that matches the account's CIS Division. When the account does not have a CIS Division, the list is not restricted

Use **New Budget Date** to define the effective date of any changes to the customer's budget amount (this date is the effective date on the respective service agreements' recurring charge history).

The system displays the sum of the service agreements' current budget amounts in **Current Account Budget**.

For accounts with a small number of service agreements, the system displays the sum of the service agreements' new budget amounts in **Recommended Account Budget**.

The grid that follows contains a row for each service agreement linked to the account that is eligible to participate in a budget billing plan:

SA Information This is a concatenation of basic information about the service agreement.

Last Changed Date This is the date when the service agreement's budget amount (i.e., recurring charge amount) was last changed.

Copyright © 2026, Oracle and/or its affiliates.

Current Budget This is the existing budget amount (i.e., recurring charge amount) for the service agreement.

Recommended Budget For accounts with a small number of service agreements, the service agreement's budget amount (i.e., recurring charge amount) will be changed to reflect this value when the page is changed. The effective date of the change is defined in **New Budget Date**.

Retroactive Budget Enrollment Process

Retroactive budget enrollment allows customers to sign up for budget billing inclusive of the current bill. It is a multi-step process that involves putting the customer's SAs linked to the current bill on budget effective as of their prior bill date, re-billing the bill segments, and deciding how to present the new information to the customer. This process is available if the following conditions are met:

- Account is not an open-item account.
- Account is not currently on budget.
- Current bill is not due.
- At least one service agreement linked to the current bill is eligible for budget.

When retroactively enrolling on budget, the system creates a budget enrollment process flow and assigns it a customer service request type. The process flow type (enroll customer on budget process) and customer service request type (enroll customer on budget request type or request type) are determined from the action method with Action Method Role of **Enroll Customer on Budget**. See Setting Up Retroactive Budget Enrollment Request Types for more information about setting up the request type and action method. The request type control some options provided in the process. For example, the recommended budget amount can be edited if the **Allow Budget Amount Edit** control is set. The process flow is designed to streamline the steps necessary to retroactively enroll the customer on budget billing and contains three steps or panels:

Budget Enrollment Details. This panel is used to set the budget amount of the service agreements linked to the current bill that are eligible for budget billing. The list of available budget plans as well as the default budget plan are the same as in the Account - Budget page. If a budget plan is already configured for the account, it will be used as the default in this panel. The budget amount must be defined for all service agreements to continue with the process.

Bill Comparison. This panel shows the current bill and a simulation of what the bill would have looked like had the account's eligible SAs been on budget.

Bill Presentment. This panel is used to determine how the changes to the current bill, resulting from enrollment in budget billing, will be presented to the customer. The changes can either be included in the current bill (*Reopen Last Bill*) or included in the next bill (*Show On Next Bill*). A new bill date must be specified when the *Reopen Last Bill* bill presentment option is selected.

> **Note**
>
> The system will prevent the process from continuing if *Reopen Last Bill* is selected as the bill presentment option and the current bill cannot be reopened.

The following are performed when the process is finished:

- Account's SAs that are eligible for budget linked to the current bill are enrolled on budget effective as of their prior bill date.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If any of the bill segments' period ended before the bill date, the earliest end date among those segments is used as the budget effective date.

- Bill segments for the SAs linked to the current bill are rebilled.
- If the selected bill presentment is *Reopen Last Bill*, the current bill is re-opened and completed to include the rebill details.
- The **Summary** panel is shown containing the simulated bill and the current bill.

Account - Deposits

The Deposits page contains information about all cash and non-cash deposits linked to an account.

> **Note**
>
> **Cash versus Non-cash Deposits.** Non-cash deposits (e.g., letters of credit, surety bonds, 3rd party deposits) can be added and maintained on this page. Cash deposits are maintained using service agreements. Refer to Cash Deposits for information about how to add and maintain cash deposits.

Open **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Deposits** page to view this information.

Description of Page

The following fields appear in the **Deposit Summary**:

Cash Deposits On Hand The total amount of cash deposits currently held for the account. This value is equal to the sum of the payoff balances from all service agreements whose SA type has a special role of Cash Deposit. Refer to Cash Deposits for information about how to add and maintain cash deposits.

Cash Deposits Requested This value is equal to the sum of Total Amount to Bill from all service agreements whose SA type has a special role of Cash Deposit. Refer to Cash Deposits for information about how to add and maintain cash deposits.

Non-Cash Deposits The total amount of active non-cash deposits currently held for the account. This value is equal to the sum of non-cash deposits displayed in the scroll below (note, non-cash deposits can be maintained in the same scroll).

Total Deposits on Hand This is the sum of cash and non-cash deposits held for the account.

Turn on **Do Not Review or Refund** if this account should not be considered by the Deposit Review background process (this process highlights customers who require an additional deposit) or by the Deposit Refund background process. Refer to The Big Picture Of Deposits for more information about these processes.

The **Non-Cash Deposit** scroll contains an entry for every non-cash deposit linked to the account. Refer to Non Cash Deposits for more information.

Copyright © 2026, Oracle and/or its affiliates.

To modify a non-cash deposit, simply move to a field and change its value. To add a new non-cash deposit, click the + button and then fill in the information for each field.

Start Date This is the date the non-cash deposit is effective.

End Date This is the expiration date of the non-cash deposit.

Non Cash Deposit Type Select the appropriate non-cash deposit type. The value selected is important as it controls whether the system requires a **Third Party Deposit** and whether the system will create a To Do entry near the expiration date of the deposit. Refer to Setting Up Non-Cash Deposit Types for more information.

Deposit Class Select the deposit class in which this non-cash deposit should be included. This value is used when the system amalgamates an account's total deposits when checking if the account holds an adequate deposit (amalgamation and recommendation happens within a deposit class). Refer to Deposit Class Controls Everything for more information about the significance of deposit class.

Deposit Amount Indicate the amount of the non-cash deposit.

Document Number Enter the document number, if any, associated with the non-cash deposit (e.g., the letter of credit number, the surety bond number, etc.).

Third Party Deposit SA If a 3rd party has paid the cash deposit for the account, reference the deposit SA associated with the 3rd party. Refer to 3rd Party Deposits for more information about how 3rd party deposits are maintained in the system.

Comments Enter any free-form comments about the non-cash deposit.

Account - Characteristics

The characteristics page contains information that describes miscellaneous information about the account. Use **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

> **Note**
>
> You can only choose characteristic types defined as permissible on the account record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Indicate the date on which the characteristic value becomes effective. Note, the effective date defaults to the account's setup date (defined on the **Main** page).

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Account - Alerts

Copyright © 2026, Oracle and/or its affiliates.

The alerts page contains information that describes various alerts assigned to the account. Use **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Alerts** page.

Alerts assigned to an account appear in the Alert Zone. They are used to bring important information to the attention of any customer service representative who looks at the account.

Description of Page

The following fields display:

Alert Type Indicate the type of alert to show on Control Central.

Start Date Indicate the date on which the alert starts showing in Control Central. The current date defaults.

End Date Indicate the date on which the alert stops showing in Control Central. The alert is shown through the end date. You can leave this field blank if the alert should be effective indefinitely. The date defaults to the **Start Date** plus the **Alert Type's** alert period.

Account - Account Portal

Select **Menu**, and then **Customer Information**, and then **Account**, and then **Search** and navigate to the **Account Portal** page to view additional information associated with an Account.

Depending on your implementations requirements, you may define additional zones to appear on this Portal.

The contents of this section describe the zones that are available on this portal page.

General Information displays Account ID and Account Info

Account Contact Information Displays person contacts that are associated with the account in context. Only records for persons that allow communication preferences are shown.

- This zone provides the ability to add or edit existing person contacts. The edit functionality is limited. This provides a convenient way to edit basic information for person contacts. When adding a person contact, person contact types associated with phone types are not available.
- Full maintenance of person contacts can be performed on the *Person – Person Contacts* page.

Push Communication Preferences Displays contact preferences and opt-out preferences for push-based notification types; meaning, those notifications that customers do not need to subscribe in order to receive. You can maintain the customer's preferred method(s) of contact (the preferences) for receiving these notifications, or process a customer's request to opt out of receiving them.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Push notifications have parent and individual notification types. Communication preferences are established for parent notification types. The individual notification types reference a parent notification type and define the message to be sent using the preferences established for the parent. As mentioned, for push notifications a subscription is not needed; a default contact can be determined in the absence of contact or opt-out preferences.
>
> Not all notification types are relevant to all customers. Notification types can be configured with a suppression criteria algorithm that can prevent some notification types from appearing. It is possible that a notification type was suppressed after active contact preferences were set up. These notification types and their contact preferences are also shown when the filter is set to show inactive preferences.

- Refer to Push Versus Subscription Notification Types and Parent versus Individual Push Notification Types for more information.
- Please see the zone's help text for information about this zone's fields.

Subscription Communication Preferences Zone displays contact preferences for subscription-based notification types - meaning those notifications to which customers need to subscribe in order to receive them. You can maintain the customer's preferred method(s) of contact (the preferences) for receiving these notifications. A notification for subscription-based notification cannot be sent without an active contact preference. Opting out is not applicable to subscription-based notification types.

> **Note**
>
> Unlike push notifications, communication preferences are established for the subscription notification types, which also defines the message to be sent.
>
> Not all notification types are relevant to all customers. Notification types can be configured with a suppression criteria algorithm that can prevent some notification types from appearing. It is possible that a notification type was suppressed after active contact preferences were set up. These notification types and their contact preferences are also shown when the filter is set to show inactive preferences.

- Refer to Push Versus Subscription Notification Types
- Please see the zone's help text for information about this zone's fields.

Account Management Portal

The Account Management portal is used to maintain multiple accounts linked to a particular customer.

Open this page using **Menu**, and then **Customer Information**, and then **Account Management**

Some examples of customers with large numbers of accounts include:

- Public school districts with many school locations

Copyright © 2026, Oracle and/or its affiliates.

- Corporate entities with many franchise locations
- Large apartment complexes with many open units
- Service providers billed for many end customers
- Cities, counties, or townships (for street lights, sewer service, etc.)

You can select one or more accounts to update, or you can broadcast an account's service agreements for group SA maintenance. Update actions can be performed on either accounts or SAs. The following items are examples of the account information that can be updated:

- Mailing address
- Characteristics
- Autopay information
- Bill cycle
- Account management group

The following items are examples of the service agreement information that can be updated:

- Rate schedules
- Contract riders and contract values
- Tax exemptions
- Characteristics

When you update information, you can select an account or SA to use as a template, and use this as a basis for updating multiple accounts. The system will copy information from the template to the selected accounts or service agreements. Refer to the embedded help on the Update Details page for more information about using a template to update accounts.

Description of Page

Use the Account Search zone to search for a person. The **Person ID** is required. **Include Hierarchy** determines whether you wish to search for accounts directly linked to the person id specified, or if you want to find all accounts in the hierarchy using person to person relationships. You can also enter one or more of the following values to further restrict the list of accounts returned: **Customer Class, Account Management Group, Bill Cycle** or **Characteristic Type/Value.**

Once a person is selected, you can view and maintain the accounts and the service agreements associate with the person. Click the broadcast icon next to an account to view SAs for the account. The Service Agreement List zone includes a summary listing of the non-closed, non-canceled service agreements of the broadcast account.

Select one or more accounts and click **Update Accounts** to modify account information. Select one or more service agreements and click **Update Service Agreements** to modify service agreement information. Refer to the embedded help on the Update Details page for descriptions of the values that can be updated.

The Person Tree zone, similar to that of Control Central, shows the hierarchy of child persons for the person that has been broadcast.

Account - SAs For A Debt Class

This is an information-only page that shows an account's service agreements grouped by debt class.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The source of debt class.** An account's debt comes from its service agreement(s). Every service agreement has an SA type. Every SA type references a debt class.

To access this page, drill down on a debt class shown on **Menu**, and then **Customer Information**, and then **Account**, and then **Search.**

Description of Page

The **Account ID** and the primary name of the account's main customer are displayed.

Debt Class is the debt class associated with the account's service agreement(s).

> **Note**
>
> An account may have service agreements that belong to other debt classes. Navigate to Account - Main Information to see a list of all debt classes associated with an account.

The **Service Agreement** scroll contains one entry for each service agreement that belongs to the **Debt Class**. The following information is displayed for each service agreement.

- **SA** (service agreement) **Info** is a concatenation of important details about the service agreement.
- **Premise** defines the service agreement's main premise.
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt. Payoff balance is only displayed if it differs from current balance.

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- **SA Arrears** displays the age of the amounts that make up the current balance.
- The **Days Old** and **Arrears Amount** information shows the age of the service agreement's debt.

> **Note**
>
> Information about debt balances by age is not applicable to open-item accounts.

Copyright © 2026, Oracle and/or its affiliates.

Using The Account/Person Replicator

There are two different ways to use this transaction:

- You can make many copies of a given Person / Account combination. This may be required if your company works with third party agencies who enroll customers on its behalf. In this situation, the third party agency may receive blocks of pre-allocated account numbers to be used when they enroll a customer. When a customer signs up, the person information can be updated with information about the newly enrolled customer.
- Perform the following steps if you need to create many copies of a given person and account:
  - Add a new person and account if you don't already have a customer to serve as your "template".
  - Use the Account/Person Replicator to create copies of the person and the account as described below (and make sure to use the Replicate Acct and Person option).
- You can create many Accounts for a single Person. This may be required if you have a customer who wants many separate accounts rather than a single consolidated account.
- Perform the following steps if you need to create many copies of a given account for a specific person:
  - Select the account that serves as the template account for the person. Note, this account must be linked to the person who wants the many accounts.
  - Use the Account/Person Replicator to create copies of the account as described below (and make sure to use the Replicate Account Only option).

Use **Menu**, and then **Customer Information**, and then **Account/Person Replicator** to open this page.

Description of Page

Choose the **Account** that serves as the template account.

> **Note**
>
> **Recommendation.** Carefully verify that the template account you have chosen is correct before you save the replicated accounts. After you save the replicated accounts, any corrections could prove time consuming.

At the top of the page, indicate the **Number** of accounts and persons **to Replicate**.

Indicate the **Main Person Name Base**, which will be used as the main person name for all the new persons to be created. This field will be protected if you choose a **Replicate Type** of Replicate Account Only (as no persons are created).

Use **Replicate Type** to define if you want to create multiple copies of an account for a given person or if you want to create many new person / account combinations. See the description that appears above for more information about this field.

After entering the above, click the **Replicate** button to generate your new accounts and persons. The grid displays informational messages describing what will happen when you click

Copyright © 2026, Oracle and/or its affiliates.

save. For example, it will verify how many accounts will be created, the person whose record will be copied, the name that will be used on all the new persons, etc..

If everything looks clean, click **save**.

- Information Replicated for New Persons
- Information Replicated for New Accounts

Information Replicated for New Persons

Except for the main name, which you indicate in the **Main Person Name** field, all information for the main person linked to the account will be copied to the new persons.

> **Warning**
>
> This copy includes the person's identification number. This will result in multiple persons with the same identification number. You should ensure that the identification number for your template person is a temporary number. When you sign up your actual customer, be sure to update the identification number to a valid number for the person.

Information Replicated for New Accounts

Although most information for an account will be copied, not all information will be. The following points describe the information that is NOT copied on newly replicated accounts:

- Mailing premise will be set to blank
- Setup Date is set to the current date
- Credit Review Date and Postpone Credit Review Until Date will be set to blank
- Auto Pay information will not be copied
- Credit Rating History will not be copied
- Non-cash deposit information will not be copied

> **Note**
>
> that all miscellaneous persons linked to the account (i.e., those who are not flagged as Main Customers) will be linked to the new accounts. With this functionality, for example, you can create a person for a third party enroller and link this enroller to the template account. This enroller will then be linked to all the new accounts.

Maintaining Service Agreements

Copyright © 2026, Oracle and/or its affiliates.

A service agreement contains the terms and conditions controlling how the system calculates charges for a specific service supplied to a customer. The topics in this section describe the pages on which service agreement related information is maintained.

> **Note**
>
> **The system creates most service agreements behind the scenes.** Most service agreements are created "behind the scenes" by the system as part of the Start Service and Order completion processes. You should only have to access the service agreement pages if you need to fine-tune what the system defaults. Refer to Start/Stop and Maintaining Orders for the details.

Service Agreement and Usage Subscription. A Usage Subscription will exist for each 'usage-oriented' service agreement.

> **NOT_SUPPORTED**
>
> For more information about service agreements and their place in the "V", refer to Understanding The "V". For more information about the source of information defaulted onto service agreements, refer to Setting Up Start Options.
>
> For more information about Usage Subscription, refer to About Usage Subscriptions.

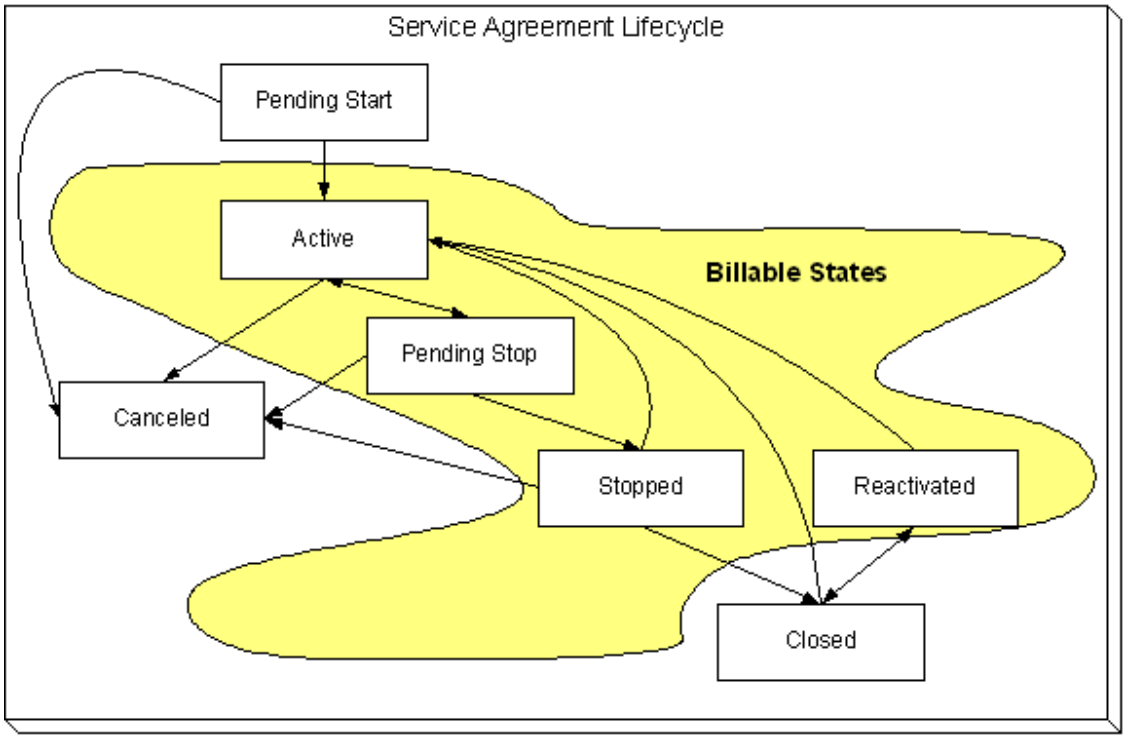
> **Warning**
>
> Take special care when adding a new service agreement to specify the appropriate Division and SA Type as it will affect how the customer is billed, what rate can be specified, how overdue debt is collected, and much more. After the service agreement is activated, you may not change its SA type.

- The Lifecycle Of A Service Agreement
- Service Agreement - Main Information
- Service Agreement - Rate Info
- Service Agreement - SA / SP
- Service Agreement - Characteristics, Quantities, & Recurring Charges
- Service Agreement - Miscellaneous
- Service Agreement - Contract Options
- Service Agreement - TOU Contract Values
- Service Agreement - Billing Scenario
- Service Agreement - SA Portal

The Lifecycle Of A Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

The following diagram shows the possible lifecycle of a service agreement.



Pending Start When you initiate a request to start service, the system initiates a service agreement in the state of Pending Start. A pending start service agreement remains in this state until everything necessary to start service is defined in the system.

Active The system automatically changes the state of Pending Start service agreements to Active when everything necessary to start service exists in the system. Refer to Activating Pending Starts for more information. While in this state, the system bills the service agreement's account for the service.

Pending Stop When you initiate a request to stop service, the system progresses the service agreement to the Pending Stop state. A pending stop service agreement remains in this state until everything necessary to stop service is defined in the system.

Stopped The system automatically changes the state of Pending Stop service agreements to Stopped when everything necessary to stop service exists in the system. Refer to Finalizing Pending Stops for more information. The service agreement remains in this state until it is paid in full.

A service agreement can be automatically stopped depending on the SA Type configuration of the **Stop Option Flag**. Service agreements configured to automatically stop will be stopped when all SP related SAs of the account are stopped. For example, if an account has a cable service and a charitable contribution service agreement then the charitable contribution service agreement can be configured to automatically stop when the cable service agreement is stopped. The stop date of the charitable contribution will be set equal to the stop date of the electric SA.

Closed The system moves a service agreement to the Closed state when it no longer has a financial balance.

Reactivated The system moves a service agreement to the Reactivated state if a financial transaction is created after a service agreement is Closed (e.g., when a payment is canceled

Copyright © 2026, Oracle and/or its affiliates.

after the service agreement is closed). Please be aware that Reactivated service agreements are processed by the Write Off Monitor.

When the service agreement once again has a zero balance (e.g., because the customer pays or the debt is written off), the service agreement returns to the Closed state.

Cancelled You may Cancel a service agreement if it shouldn't exist (because the customer doesn't want service). While cancellation doesn't physically remove a service agreement from the database, most queries and trees suppress information about canceled service agreements. This means cancellation is a nice way to logically delete undesired service agreements.

You'll notice in the above diagram that you can cancel an Active, Pending Stop or Stopped service agreement. You may only do this if the service agreement's financial transactions and non-billed billable charges are all cancelled.

You should rarely have to change a service agreement's status. Why? Because the system changes the status behind-the-scenes when various events take place. For example,

- When you request a service start using the Start/Stop page, the system creates a service agreement in the pending start state.
- If you cancel a pending start before service is activated, the system automatically changes the state of the service agreement to cancelled.
- When everything necessary to start service exists, the system automatically changes the state of the service agreement to active.
- When you request a service stop using the Start/Stop page, the system changes the state of the stopped service agreement to pending stop state.
- When everything necessary to stop service exists, the system automatically changes the state of the service agreement to stopped.
- When a final payment is received and the stopped service agreement's balance becomes zero, the system automatically changes the state of the service agreement to closed. The system can also close a service agreement as part of the write-off processing.
- If the payment bounces, the system automatically changes the state of the service agreement to reactivated. And the system will change it back to closed when the account's balance returns to zero.
- If a billable charge is interfaced into the system, the system automatically changes the state of the service agreement to reactivated.

You may need to manually change a service agreement's state for the following reasons:

- If a service agreement was activated inadvertently, you may cancel it (and therefore logically remove it from the system) after canceling all of its financial transactions. You do this by clicking the cancel button on the service agreement page. You may also cancel an incorrect service agreement if the status transitions to pending stop or stopped before you have a chance to correct the mistake.
- If you don't want to wait for the system to automatically activate a pending start service agreement, you may push the activate button on the service agreement page. You might do this in order to create a bill for an impatient customer.
- If you don't want to wait for the system to automatically finalize a pending stop service agreement, you may push the stop button on the service agreement page. You might do this in order to create a real time final bill for a customer.
- If you don't want to wait for the system to automatically close a stopped service agreement, you may push the close button on the service agreement page.

Copyright © 2026, Oracle and/or its affiliates.

- Sometimes a service agreement is closed or stopped by mistake. You can activate a stopped, closed, or reactivated service agreement using the **Reinstate SA** button on the service agreement page. Using the **Reinstate SA** button retains the current service's billing history.

> **Note**
>
> **One time invoices (i.e., one-time billable charges) are interesting.** One-time invoice service agreements follow the above lifecycle; it's just that they do it quickly. You see, when a one-time invoice service agreement is created, both its start and stop dates are known (they are the invoice date). If you followed the above state transition discussion, you'll understand that the system progresses these invoices from pending start to stopped when the service agreement is activated.

- Renewing Service Agreements
- Expiring Service Agreements

Renewing Service Agreements

Some service agreement types (e.g., non-billed budgets) may allow or require renewal. If renewal is optional, a renewal date may be specified when the service agreement is created. If renewal is required, you must specify a renewal date.

The SA renewal background process initiates the renewal for all active service agreements when the renewal date is reached (i.e., the renewal date is on or before the process date). When a service agreement is renewed:

- The SA renewal algorithm (specified on the SA type) is executed. The SA renewal algorithm can be used to customize the processing required to renew an SA.
- If the SA renewal algorithm is successful, the renewal and expiration date (if calculated by the renewal algorithm) on the service agreement are updated with the values returned from the renewal algorithm.
- If the renewal process is not successful, a To Do list entry (of type TD-SARN) is created for the account and service agreement.

Expiring Service Agreements

Service agreements may specify an expiration date. Some service agreement types (e.g., Non-Billed Budgets) require an expiration date. The SA expiration background process initiates the request to stop service for all active SAs when the expiration date is reached (i.e. the expiration date is on or before the process date).

Service Agreement - Main Information

The Main page contains basic service agreement information. Open **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** to maintain this information.

Copyright © 2026, Oracle and/or its affiliates.

Description of Page

SA (service agreement) **Info** and **SA ID** are displayed on every page. These values only appear after the service agreement exists on the database. The **ID** is a system assigned random number that stays with a service agreement for life. The **SA Info** is a concatenation of important details about the service agreement and its account.

> **Note**
>
> **Formatting may be performed by a plug-in.** The contents of **SA Info** may be formatted by a plug-in algorithm on SA Type. Refer to the base package's C1-SAT-INFO for an example. If such an algorithm is not plugged-in on the SA Type, the system looks for a corresponding algorithm on the installation record. Refer to the base package's C1-SAI-INFO for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

SA Status defines the state of the service agreement. The system typically changes the state of a service agreement behind-the-scenes. If you can't wait for the system, use one of the adjacent buttons:

- Click the **Activate SA** button to activate a Pending Start service agreement.
- Click the **Cancel SA** button to cancel a Pending Start or Active, Pending Stop or Stopped service agreement.
- Click the **Initiate Stop** button to populate the **End Date** and change the state of the service agreement to Pending Stop.
- Click the **Stop SA** button to stop a Pending Stop service agreement.
- Click the **Close SA** button to close a Stopped or Reactivated service agreement.
- Click the **Reinstate SA** button to reinstate a Closed, Reactivated, or Stopped service agreement to an Active service agreement.

> **NOT_SUPPORTED**
>
> Refer to The Lifecycle of a Service Agreement for more information about **SA Status**.

If the quotation module is not turned off, a "real" service agreement can be created when a proposal service agreement is accepted. If the service agreement you've displayed was created in such a way, information about the **Proposal SA** will appear beneath **SA Status**.

Enter the **Account ID** that is financially responsible for the service agreement. If you change a service agreement's **Account ID**, you are effectively transferring this service agreement (and its debt) to the new account.

Indicate the **CIS Division** and **SA Type**. These fields are very important as they control numerous aspects of the service agreement's behavior. These fields are gray when the status is not Pending Start.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the CIS Division is set to the account's CIS Division and cannot be changed.

> **NOT_SUPPORTED**
>
> For more information about what the SA type controls, refer to Setting Up SA Types.

The **Start Date** defines when the financial relationship begins. The **End Date** defines when the financial relationship terminates. If the end date is blank, the service agreement has not yet been terminated.

Use **Maximum Bill Threshold** if you want the system to generate a bill error when a bill segment is produced in batch that exceeds a given value. These bill errors will appear on the standard billing queries and To Do lists. If, after reviewing the high value bill segment, a user truly intends to send the bill out, they should regenerate the bill. Refer to How To Correct A Bill Segment That's In Error for more information.

> **Note**
>
> **Default note.** The value of **Maximum Bill Threshold** defaults from the service agreement's SA type.

If the **SA Type** has a special role of Bill Determinants Required, you must enter the **Cutoff Time** and **Start Day Option** that is used on a bill determinants usage request.

- The **Cutoff Time** is entered in "legal time". This value defaults from either the start option or the installation record.
- For **Start Day Option**, you may choose Current Day or Previous Day. This value defaults from either the start option or the installation record.

Use **Customer Read** to define if the customer reads their own meter. Valid values are: Yes, No.

Turn on **Allow Estimates** if it's OK for the billing process to use estimated usage from Oracle Utilities Meter Data Management for this service agreement. If the SA type does not allow estimates, this field will be gray.

If you've set up the system to use start options, you can have the system default many fields by copying the terms from a start option when service is started using either Start / Stop or the Order transaction. The **Start Option** field contains the last start option whose field values were copied to the service agreement. If you need to change a service agreement's terms by applying a different start option, you can. Refer to Changing A Start Option for a description of how to do this.

If the service agreement's rate uses premise-based characteristics for taxation and other purposes, **Char**acteristic **Premise ID** defines the premise that supplies these characteristics. This field also defines the premise under which the service agreement's bill segments are classified and is used to identify where the service agreement's service is located on various windows. This field is gray if the service agreement's SA type doesn't require a characteristic

Copyright © 2026, Oracle and/or its affiliates.

premise. For more information about rates and their use of characteristics, refer to Setting Up Bill Factors .

> **Note**
>
> **Selecting a characteristic premise.** If the service agreement has service points, only premises associated with these service points may be selected as a **Char Premise**. If the service agreement doesn't have service points, any premise may be selected.

Your conversion process will fill in **Old Account ID** for service agreements that were converted from your legacy CIS system. The payment upload process uses this field to locate the accounts for payments that reference a legacy account number.

If the service agreement requires a total amount to bill, the amount is displayed. This field is typically only used on loan or cash deposit service agreements. It holds either the loan amount or the desired cash deposit. If you increase this field's value, the customer will be billed for any difference between the new amount and the total amount billed (and paid for) to date. The label that appears for this field is defined on the SA type (on the Billing tab). We recommend that the system be set up so that this label is very specific, for example:

- It should be called **Cash Deposit Amount** if the total amount to bill is for a deposit.
- It should be called **Loan Amount** if the total amount to bill is for a loan.

Number of Payment Periods only appears for loan SA types. It displays the number of periods specified when the loan was first activated. (The number of payment periods may have been subsequently changed on Loan maintenance.)

If the service agreement uses a recurring charge, the current recurring charge amount is displayed. Transfer to the Service Agreement - Recurring Charge to update this value. This field is not visible if the service agreement's SA type doesn't allow recurring charges. The label that appears for this field is defined on the SA type (on the Billing tab). We recommend that the system be set up so that this label is very specific, for example:

- It should contain **Budget Amount** if the recurring charge is for a budget.
- It should contain **Installment Amount** if the recurring charge is used for a deposit being paid in installments.
- It should contain **Payment Amount** if the recurring charge is used for a loan.

SA Relationship ID contains the ID of the SA Relationship that is associated with a sub service agreement. This field is only visible on sub service agreements. Refer to Sub Service Agreements for more information.

CIAC Review Date displays the last date that the Contributions In Aid of Construction (CIAC) review process ran for this service agreement. This field is only visible if the SA type indicates this is a CIAC service agreement. Refer to CIAC Segmentation for more information on CIAC.

Deposit Interest Calculated contains the last date the system applied interest. This field is only visible if the SA type has a special role of Cash Deposit.

The **Sched**uled **Payment Autopay** flag determines if the Non-Billed Budget scheduled payments are excluded from automatic payment or included by automatic payment. If the account is not set up for automatic payment for the period that covers the Non-Billed Budget, this flag cannot be set. This field is only visible if the SA has a special role of Non-Billed Budget.

Copyright © 2026, Oracle and/or its affiliates.

The **NBB Recommendation Rule** displays the description of the non-billed budget recommendation rule used for this SA. This field is only visible if the SA has a special role of Non-Billed Budget.

Expiration Date contains the date at which the service agreement is set to expire.

Renewal Date contains the date at which the service agreement is to be renewed. Renewal date is disabled if renewal is not allowed on the SA type.

Debt Class indicates the debt class into which the service agreement's current balance is categorized by credit and collections. A service agreement's debt class is defined on its SA type. Refer to Different Collection Criteria For Different Customers And Different Debt for more information about debt class.

Current Balance indicates the amount the customer currently owes for the service.

Payoff Balance indicates the amount the customer would owe if they wanted to close the service agreement. This value is only displayed when it differs from current balance.

> **NOT_SUPPORTED**
>
> Refer to Current Amount versus Payoff Amount for more information.

The **Days Old** and **Arrears Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

> **Note**
>
> Information about debt balances by age is not applicable to open-item accounts.

This bottom of this page contains a tree that shows the various objects linked to the service agreement. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

The topics that follow describe miscellaneous features on this page.

- Changing A Start Option
- The Impact of Proposal Service Agreements

Changing A Start Option

If the service agreement is in the pending start state, you can use the Start/Stop - Pending SAs page to change the service agreement's SA type and/or start option.

If the service agreement is in the active or pending stop states and if the SA type uses start options, a button appears on Service Agreement - Main called **Apply New Start Option**. When clicked, the **Apply New Start Option** dialog box opens.

The fields in this dialog box allow you to define a **Start Option** whose terms should be copied to the service agreement as of a given effective date (**Apply Start Option on**). Because most terms on a service agreement are effective-dated, it's important that you understand the

Copyright © 2026, Oracle and/or its affiliates.

following ramifications of applying a start option (note, we use the word "terms" to generically reference any of the fields that can be defaulted from a start option):

- Terms are reused if **Start Option** contains the same value as defined in the service agreement. For example,
  - If the **Start Option** references the same rate as defined in the service agreement, a new rate record will NOT be created.
  - If the **Start Option** references a contract rider that's currently defined in the service agreement, a new rider will NOT be created.
- Terms that support explicit expiration (i.e., those with an end date) are expired if they are not referenced on the new **start option**. For example, if the service agreement has a rider that is not referenced on the new **start option**, it will be expired.
- Terms that support implicit expiration (i.e., those without an end date) are left unchanged if they are not referenced on the new **start option**. For example, a service agreement characteristic that isn't referenced on the new **start option** will NOT be modified when a new start option is applied.

> **Note**
>
> **Only start options related to the service agreement's SA type can be used.** It should be noted that only start options related to the service agreement's SA type can be specified. If the service agreement is in the pending start state, you can change the service agreement's SA type on the **Main** tab or via Start/Stop - Pending SAs. If the service agreement is active, you may not change its SA type.

When you click **OK** a warning window will appear that summarizes the changes that will be made to the service agreement. After confirming what will be changed, click **OK** to change the service agreement's terms. The **Start Option** that is shown on the **Main** tab will now reflect the start option whose terms were just applied.

The Impact of Proposal Service Agreements

If the quotation module is not turned off, you may display a proposal service agreement on Service Agreement - Main. This section describes how Service Agreement - Main differs when a proposal service agreement is displayed.

The **SA Status** section is replaced with a section that shows the status of the proposal service agreement (proposal SA).

Proposal SA Status defines the state of the proposal service agreement. The Quote Maintenance transaction typically changes the status of a proposal SA when users accept, decline and cancel quote details. If you wish to change the status of a proposal SA independent of this transaction, you can press one of the adjacent buttons

- Click the **Activate** button to change the status from Pending to Quotable. Note, you can set up the system to automatically make newly created proposal SAs quotable by plugging in the appropriate Proposal SA Creation algorithm on the SA's SA type.
- Click the **Accept** button to change the status from Quotable to Accepted. When you accept a proposal SA, you will be asked to supply the **Start Date** of the real service agreement that will be created.

Copyright © 2026, Oracle and/or its affiliates.

- Click the **Decline** button to change the status from Quotable to Declined. When you decline a proposal SA, you will be asked to supply a **Decline Reason**.
- Click the **Cancel** button to change the status from Quotable to Canceled.
- Click the **Reactivate** button to return a Canceled or Declined proposal SA to the Quotable state.

> **NOT_SUPPORTED**
>
> Refer to Proposal SA State Transition for more information about **Proposal SA Status**.

If the proposal SA has been declined, the **Decline Reason** appears beneath **Proposal SA Status**. Refer to Accepting / Declining Quote Details for more information.

It the proposal SA was changed after a quote detail was generated using the proposal SA's terms, a **Remark** appears beneath **Proposal SA Status** highlighting such.

> **Note**
>
> **Changing A Start Option.** You can Apply A New Start Option on pending and quotable proposal SAs.

If a real service agreement was created when the proposal SA was accepted, **Real SA ID** appears and contains the ID of the real service agreement. Refer to Accepting / Declining Quote Details for more information.

Service Agreement - Rate Info

The rate info page contains the service agreement's rate and information needed by the rate to calculate the service agreement's bill segments. This information is only displayed if the service agreement's SA type allows a rate.

Open **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and navigate to the **Rate Info** page.

Description of Page

The **Rate** grid controls the rate used to calculate the service agreement's bill segments. Multiple rows will only exist if the service agreement's **Rate** changes over time.

A service agreement's **Rate** is populated by the system when service is started. The method used to populate the initial **Rate** depends on how service is started and whether you use start options:

- If the service agreement's SA type uses start options:
  - If the Order transaction is used to start service, the rate is populated using the start option associated with the package that was selected by the user when the order was completed.

Copyright © 2026, Oracle and/or its affiliates.

  - If Start/Stop is used to start service, the rate is populated using the start option selected by the user when service was initiated.
- If the service agreement's SA type does not use start options, the SA type's default rate is populated on the service agreement.

Billing selects a single rate from the **Rate** grid when it calculates a bill segment for the service agreement. If multiple rates are defined in this grid, the rate is selected based on the service agreement's SA type's Rate Selection Date. The following fields display:

- **Effective Date** is the date the rate becomes effective. The service agreement's start date defaults.
- **Rate Schedule** defines the rate used to calculate the service agreement's bill segments. Note, you can only choose rates defined as permissible on the service agreement's SA type; the permissible rates are shown in the adjacent tree.

> **Note**
>
> **Possible override**. If your service agreement is linked to a terms of service record, it is possible that the rate used by the system could get overridden by one referenced on the terms of service. If the rate is overridden by the TOS, the tree on this page indicates the rate source.

The **Contract Rider** grid contains the contract riders that are in effect for the service agreement. This information is defaulted when service is started using a start option (using the contract rider information defined on the start option). You should only have to access this section if you need to fine-tune what the system defaults.

Billing uses this information when it calculates a bill segment for the service agreement if:

- The rate used to calculate the bill segment contains a calculation rule that references a bill factor.
- The referenced bill factor requires a contract rider to levy the associated charge.

If a **Contract Rider** does not exist or is not effective, the charge associated with the calculation rule is not applied to the customer's bill.

To modify a contract rider linked to the service agreement, simply move to a field and change its value. To remove a contract rider, click the - button. To add a new contract rider, click the + button and fill in the information for each field. The following fields display:

- Use **Start Date** to define the date on which the contract rider becomes effective. The service agreement's start date defaults.
- Use **End Date** to define the date on which the contract rider expires. This field need only be specified if the contract rider expires on an explicit date.
- Use **Bill Factor** to define the type of rider. You may only reference **Bill Factors** designated as being applicable for contract riders. Note, the **Bill Factors** linked to the rate that are marked with contract rider applicability are shown in the adjacent tree under the Contract Rider node.

The **Contract Values** grid contains the contract values that are in effect for the service agreement. This information is defaulted when service is started using the contract value information defined on the start option used to start service (if any). You should only have to access this page if you need to fine-tune what the system defaults.

Billing uses this information when it calculates a bill segment for the service agreement if:

Copyright © 2026, Oracle and/or its affiliates.

- The rate used to calculate the bill segment contains a calculation rule that references a bill factor.
- The referenced bill factor uses a contract value to define the amount charged.

The following fields display:

- Use **Start Date** to define the date on which the contract value becomes effective. The service agreement's start date defaults.
- Use **End Date** to define the date on which the contract value expires. This field need only be specified if the value rider expires on an explicit date.
- Use **Bill Factor** to define the type of value. You may only reference **Bill Factors** designated as allowing values in contract terms. Note, the **Bill Factors** linked to the rate that are marked as allowing values in contract terms are shown in the adjacent tree under the Contract Value node.
- Use **Value** to define contract value.

> **NOT_SUPPORTED**
>
> For more information about contract values, refer to Defining General Bill Factor Information .

The **Tax Exemptions** section contains the taxes from which the service agreement is all or partially exempt.

Billing uses this information when it calculates a bill segment for the service agreement if:

- The rate used to calculate the bill segment contains a calculation rule that references a bill factor.
- The referenced bill factor can have tax exemptions that limit the amount of the charge.

If the bill factor doesn't have a tax exemption or there are none that are effective during the bill period, the tax is levied in full. If the service agreement is exempt, the appropriate charge associated with the tax is calculated. We strongly recommend using the "%T" substitution character in the Description on Bill on tax-exemptible calculation rules so that an appropriate tax exemption message appears on the customer's bill.

The following fields display:

- Use **Start Date** to define the date on which the tax exemption becomes effective.
- Use **End Date** to define the date on which the tax exemption expires.
- Use **Bill Factor** to define the tax from which the service agreement is exempt. You may only reference bill factors designated as being tax exemptible. Note, the **Bill Factors** linked to the rate that are marked as tax exemptible are shown in the adjacent tree under the Tax Exemption node.
- Use **Percent Exempt** to define the customer tax exemption percent (e.g., 90 means the customer is 90% exempt and will only pay 10% of the normal tax rate).
- Use **Tax Exempt Type** to specify the type of exemption.
- Use **Tax Certification** if the customer has a tax exemption certificate.
- Rate Information Tree

Copyright © 2026, Oracle and/or its affiliates.

Rate Information Tree

The tree shows the service agreement's permissible rates and the related types of contract riders, contract values and tax exemptions that are possible given these rates.

- The Permissible Rates node displays all the rate schedules associated with the service agreement's SA Type.
- The Effective Rate Schedule node indicates the rate schedule currently effective.
- The Contract Rider node displays the bill factors linked to the service agreement's rate(s) that may have contract riders defined for them. If you expand a node, you can see the individual calculation rule(s) on which the bill factor is specified. If the service agreement has a value defined for the contract rider that is currently in effect, it is also shown.
- The Contract Value node displays all the bill factors linked to the service agreement's rate(s) that may have contract values defined for them. If you expand a node, you can see the individual calculation rule(s) on which the bill factor is specified. If the service agreement has a contract value defined that is currently in effect, it is also shown.
- The Tax Exemption node displays all the bill factors linked to the service agreement's rate(s) that may have tax exemptions defined for them. If you expand a node, you can see the individual calculation rule(s) on which the bill factor is specified. If the service agreement has a tax exemption defined for the bill factor that is currently in effect, it is also shown.

If the service agreement is currently linked to a terms of service record for an effective umbrella agreement, additional information is displayed in the tree.

- The Permissible Rates node displays the rate source value associated with the service agreement's SA Type.
- If the terms of service record linked to the service agreement has a template SA and the rate source is Check TOS First, then SA , the Effective Rate Schedule node displays the rate schedule for the template SA. An additional node indicating whether the rate source is the template SA or the current SA is displayed below.
- The Contract Rider, Contract Value and Tax Exemption nodes include the terms of service usage value from the bill factor.

NOT_SUPPORTED

Refer to Rate Terms Overridden By Terms Of Service for more information.

Service Agreement - SA / SP

The SA / SP page contains the service points that are linked to the service agreement. This information is only displayed if the service agreement's SA type allows service points.

This information is updated by the system when service is started. You should only have to access this page if you need to fine-tune what the system defaults.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> You must understand the "V" very well before attempting to maintain this information. Refer to Understanding The "V" **for more information**.

Use **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and navigate to the **SA/SP** page

Description of Page

Billing uses **SA / SP** information to amalgamate consumption when it calculates a bill segment for the service agreement. The following fields are displayed:

Service Point ID The identity of the service point. The type and location of the service point is displayed adjacent. This field is gray after the service point is committed to the database.

Meter/Item Info The meter / item that is currently installed at the service point is displayed.

How to Use (SP) Indicates how billing uses the consumption associated with the service point. This field is gray for item-type service points. Valid values are: Add, Check, Subtract. Add service points have their consumption added to the total amount of consumption billed under a service agreement. Subtract service points have their consumption subtracted from the total amount of consumption billed under a service agreement. Check service points are ignored by billing.

Use Percent Indicates the percentage of this service point's consumption that is billed under the service agreement. This field is gray for item-type service points.

Start Date/Time The date and time the service point's consumption starts being billed under the service agreement. This is defaulted from the service agreement's start date.

> **Note**
>
> For SAs whose type has a special role of Bill Determinants Required, the time component of the start date/time does not necessarily reflect the SA's cutoff time.

Start Meter Read is not applicable for Customer to Meter (C2M).

Stop Date/Time The date and time the service point's consumption stops being billed under the service agreement. This is defaulted from the service agreement's end date.

> **Note**
>
> For SAs whose type has a special role of Bill Determinants Required, the time component of the stop date/time does not necessarily reflect the SA's cutoff time.

Stop Meter Read is not applicable for Customer to Meter (C2M).

The **Field Activity** grid contains the list of field activity requests that have been generated in respect of this SA / SP. Refer to Field Activities to Start Service, Field Activities to Stop Service and The Big Picture Of Severance Events for more information.

Copyright © 2026, Oracle and/or its affiliates.

Field Activity The unique ID of any field activity associated with starting, stopping, investigating, or severing the service at the service point.

Field Activity Type Indicates why the field activity is linked to the SA/SP. Valid values are Start Activity, Stop Activity, Severance Activity, Cut Process, or Service Activity.

The **Characteristics** grid contains characteristics that are only applicable for a given SA / SP combination. These are typically only used for obscure SA/SP relationships for which unusual pricing terms

> **Note**
>
> You can only choose characteristic types defined as permissible on the SA/SP record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Indicate the date on which the characteristic value becomes effective. The service agreement's start date defaults.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Service Agreement - Characteristics, Quantities, & Recurring Charges

This page contains the Characteristics, Contract Quantities, and Recurring Charges that are in effect for the service agreement.

Use **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and navigate to the **Chars, Qty & Rec. Charges** page.

Description of Page

The **Characteristics** grid contains information that describes miscellaneous information about the service agreement. This information is defaulted when service is started using the characteristics defined on the start option used to start service (if any). You should only have to access this section if you need to fine-tune what the system defaults.

> **Note**
>
> You can only choose characteristic types defined as permissible on the service agreement record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Indicate the date on which the characteristic value becomes effective. The service agreement's start date defaults.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Copyright © 2026, Oracle and/or its affiliates.

The **Contract Quantity** grid contains the contract quantities that are in effect for the service agreement. This information is defaulted when service is started using the contract quantities defined on the start option used to start service (if any). You should only have to access this section if you need to fine-tune what the system defaults.

Billing may use this information when it calculates a bill segment for the service agreement if the rate has pre-processing calculation groups. Refer to All Calculation Rule-Based Rates Share a Common Structure for more information about pre-processing calculation groups.

The following fields display:

Effective Date The date this quantity becomes effective. The service agreement's start date defaults.

Contract Quantity Type The type of contract quantity.

Contract Quantity The contract quantity.

The **Recurring Charge** section contains effective-dated information that defines the recurring charge amount used to calculate the service agreement's bill segments. This information is only displayed if the service agreement's SA type allows a recurring charge amount.

A service agreement's recurring charge amount history is updated by the system behind-the-scenes as part of the Start Service process. The budget recalculation process can also update this information. You should only have to access this section if you need to change a service agreement's recurring charge amount on some future date.

Billing uses recurring charge information to select the appropriate recurring charge when it calculates a bill segment for the service agreement. To modify a recurring charge, simply move to a field and change its value. To change the recurring charge amount effective on a given date, click the + button to insert a row, then fill in the information for each field. The following fields display:

Effective Date The date the charge becomes effective. The service agreement's start date defaults.

> **Note**
>
> **Recurring charge changes during a bill period.** Only one recurring charge will be used on any bill segment. The system selects the recurring charge value effective on the end date of the bill segment.

Recurring Charge Amount The recurring charge amount.

Service Agreement - Miscellaneous

The miscellaneous page contains information that describes miscellaneous information about the service agreement. Use **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and navigate to the **Misc** page.

Description of Page

Start Requested By displays the information entered in the Requested By field on the Start Stop page when this SA was started.

Copyright © 2026, Oracle and/or its affiliates.

Stop Requested By displays the information entered in the Requested By field on the Start Stop page when this SA was stopped.

Select the **SIC** associated with the customer. This is typically only used for commercial and industrial customers.

Enter a **Business Activity** associated with the customer. This can be used to further define what types of activities are served by this contract.

The bill message grid contains information about the message(s) to appear on a service agreement's bill segments. Both one-time and permanent messages may be defined. This information is used when the system assembles messages to appear on an account's bill (as part of bill completion). Refer to The Source Of Bill Messages for more information. The following fields are displayed:

- For **Bill Message**, select a message code to appear on the customer's bill.
- For **Bill Message Type**, use Temporary to indicate the message should only be linked to the next bill produced for the account. Use Permanent if the message should appear on every bill. Note, a value of Temporary defaults.

The terms and conditions (T&C) grid contains the service agreement's T&C's. This information is defaulted when service is started using a start option (using the T&C information defined on the start option). You should only need to change this information if you need to fine-tune the defaults. Refer to Legal Terms and Conditions May Be Specified on Service Agreements for more information. The following fields display:

- Use **Start Date** to define the date on which the T&C becomes effective. The service agreement's start date defaults.
- Use **End Date** to define the date on which the T&C expires. This field need only be specified if the T&C expires on an explicit date.
- Use **Terms and Conditions** to define the type of T&C.

Service Agreement - Contract Options

The contract options page contains the contract options and contract option events for service agreements that request usage (or billing determinants) based on interval data.

Open this page using **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and then navigate to the **Contract Options** page.

Description of Page

The **Contract Option** collection contains effective-dated information that defines the contract options linked to the service agreement. The following information may be defined for each contract option:

Contract Option Id This is the id of the contract option to link to the service agreement. The **SA Contract Option ID** is displayed adjacent. This is the unique identifier of the SA / contract option record.

Effective Date/Time This is the date and time that the contract option is in effect for the service agreement.

Expiration Date/Time This is the date and time that the contract option is no longer in effect for the service agreement.

Copyright © 2026, Oracle and/or its affiliates.

The **Override Events Collection** contains information about the override events for a shared contract option.

If a contract option event for a shared contract option should not be applicable to this service agreement, enter the **Contract Option Event ID**, the **Override Date/Time** that the override is effective and the **Status**. The status value of Active means that the override is in effect. The status value of Inactive 'cancels' the override, meaning that the event is applicable to this service agreement.

If seasonal time shift records are linked to both the base time zone and to the time zone on the service agreement's characteristic premise, the **Seasonal Time Shift** associated with the time zone on the characteristic premise is displayed.

The **Seasonal Time Shift Remark** indicates whether the data is displayed and should be entered in legal time or standard time. If the data is displayed in legal time use the **Show in Standard** button to toggle to standard time. If the data is displayed in standard time use the **Show in Legal** button to toggle to legal time.

> **Note**
>
> **This tab may not appear.** This tab is suppressed if the interval billing Complex Billing module is turned off.
>
> **Contract Option and Dynamic Option.** A Dynamic Option will exist for each Contract Option.

> **NOT_SUPPORTED**
>
> For more information about Dynamic Option, refer to Dynamic Option.

Service Agreement - TOU Contract Values

This page is not applicable for Customer To Meter (C2M).

- TOU Contract Values Tree

TOU Contract Values Tree

This page is not applicable for Customer To Meter (C2M).

Service Agreement - Billing Scenario

> **Note**
>
> **This tab may not appear.** This tab is suppressed if the quotation Contract Management module is turned off.

Copyright © 2026, Oracle and/or its affiliates.

This page allows you to define a proposal service agreement's billing scenario. Refer to Proposal SAs Contain Billing Scenarios for more information.

Open this page using **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search,** then navigate to the **Billing Scenario** page.

Description of Page

This page is protected if the service agreement being displayed is not a proposal service agreement .

When a quote is generated for a proposal SA, the system creates a simulated bill segment for each billing scenario linked to the proposal SA. Each simulated bill segment shows the prospect how much their bill would be if the pricing terms on the proposal SA were applied to the billing scenario's consumption in the related time period.

> **Note**
>
> **You should not have to enter this information from scratch.** You can set up the system to automatically generate billing scenarios when a proposal SA is created. Refer to Proposal SAs Contain Billing Scenarios for the details.

The following information is defined for each billing scenario:

- **Start Date** and **End Date** define the simulated bill segment's period.
- The grid contains the consumption that will be priced during this period:
  - **Unit of Measure** is the unit of measure of the service quantity.
  - **TOU** is the time-of-use code of the service quantity.
  - **SQI** is the service quantity identifier of the service quantity.
  - **Initial Service Quantity** is the initial quantity passed to the rate (and massaged by the rate's pre-processing calculation group(s), if any exist).
  - **Service Point ID** is used if your rate uses pre-processing calculation groups that require characteristic values from the service point or premise. For example, the calculation rule that converts CCF to therms uses a bill factor to store the conversion factor. This bill factor may have been configured to use characteristics on the service point / premise (e.g., the conversion factor may differ based on the storage location associated with the service point).

Service Agreement - SA Portal

Select **Menu**, and then **Customer Information**, and then **Service Agreement**, and then **Search** and navigate to the **SA Portal** page to view additional information associated with an SA.

Depending on your implementation’s requirements, you may define additional zones to appear on this Portal.

The contents of this section describe the zones that are available on this portal page.

General Information displays **SA ID** and **SA Info**.

Consumer Contract Products displays the service agreement’s consumer contract products. **Add**, **Edit** and **Delete** actions are provided.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining SA Relationships

The SA Relationship page contains information about a specific relationship between a service provider and a customer.

> **Warning**
>
> The following page looks deceptively simple. It's not. Refer to The Big Picture of SA Relationships and Service Providers **for more information**.

Open **Customer Information, SA Relationship** to maintain this information.

Description of Page

The **SA Relationship Info** is a concatenation of important information about the SA relationship record. The **SA Rel ID** is a system-assigned random number that stays with a SA relationship record for life. This value is assigned when the record is added to the database.

SA Relationship Status defines the state of the SA relationship. The buttons are used to change the state:

- Click the **Activate SA Rel** button to activate a **Pending** SA relationship. This button is only enabled when the status of the SA relationship is **Pending**.
- Click the **Cancel SA Rel** button to cancel an **Active** or **Pending** SA relationship. This button is only enabled when the status of the SA relationship is **Active** or **Pending**.

> **Note**
>
> **Automatic SA Relationship creation and activation.** The system automatically creates and activates required SA Relationships when the Activate SA background process sees that there are required SA relationships for the master SA Type.

Account ID is the account associated with the **Service Agreement ID**. This field is protected after the SA relationship has been added to the database.

Use **Service Agreement** to define the "master" service agreement ID that has a relationship. This field is protected after the SA relationship has been added to the database.

Premise is a display-only field that shows the **Service Agreement**'s characteristic premise (i.e., the primary premise associated with the service agreement).

Use **SA Relationship Type** to define the type of relationship. This field is only enabled if:

- A **Service Agreement ID** is specified, and
- The status of the SA Relationship is not **Canceled**, and
- All sub SAs in the grid at the bottom of the page (if any) are **Canceled**.

Enter the **Effective Date** of the relationship. This field is only enabled if:

- A **SA Relationship Type** is specified, and

Copyright © 2026, Oracle and/or its affiliates.

- The status of the SA Relationship is not **Canceled**.

Usage defines how this relationship is used. The following options are available:

- **Relationship With SPr**. Use this option if this relationship is used to define a relationship between a customer and a service provider.
- **Expire SA Relationship**. Use this option if this relationship expires any prior relationship between a customer and a service provider and there is no relationship with a service provider going forward.

If the **SA Relationship Type**'s Service Provider Model is **Market Service Provider**, the following fields are enabled:

- **Market** identifies the market in which the service provider participates. Selecting a value will enable **Provider Configuration Option**.
- **Provider Configuration Option** specifies a service provider's billing/payment relationship within the specific market. Selecting a value will enable the **Market Relationship Service Provider.**.
- **Market Relationship Service Provider** specifies the 'It's Us' service provider, meaning the utility company who is implementing this system. This is either a distributor or a retailer, depending on the utility's role in the market. The selection list consists of service providers with the selected billing/payment relationship for the selected market.

> **Note**
>
> Refer to Setting Up Market SA Relationship Options for more details on Service Provider Market Options.

The **Service Provider** associated with this relationship is the one who provides the related service, if any. You can only choose service providers who are linked to the "master" service agreement's SA type and the above **SA Relationship Type**. In addition, if the **SA Relationship Type**'s Service Provider Model is **Market Service Provider**, the service provider must also be valid for the selected **Market**, **Provider Configuration Option** and **Market Relationship Service Provider**. The **Service Provider** field is only enabled if:

- **Usage** is **Relationship with SPr**, and
- A **SA Relationship Type** is specified, and
- The **SA Relationship Type**'s Service Provider Model is **Market Service Provider** and **Market**, **Provider Configuration** and **Market Relationship Service Provider** are specified, and
- The status of the SA Relationship is not **Canceled**, and
- All sub SAs in the grid at the bottom of the page (if any) are **Canceled**.

If the **Service Provider** reads the customer's meter (as defined on the service provider's record), the customer's override **Read Cycle** should be defined. This field is only enabled if:

- A **Service Provider** is specified, and
- The **Service Provider** has the ability to override the service point's *service cycle*, and
- The status of the SA Relationship is not **Canceled**

The **Account Number in Provider's CIS** is the customer's ID in the service provider's system. This field is only enabled if:

- A **SA Relationship Type** is specified, and

Copyright © 2026, Oracle and/or its affiliates.

- The status of the SA Relationship is not **Canceled**.

The **Create Sub SA** button is used to create Sub Service Agreements for a service provider for which your organization provides billing services. This button will be gray if:

- You do not provide billing services for the service provider (i.e., the service provider doesn't have a billing relationship of **It's Us** or **We Bill For Them**), or
- If the SA relationship's **Status** is not **Active**, or
- If the status of the master **Service Agreement** is not **Active**, **Pending Start**, or **Pending Stop**.

The grid at the bottom contains the sub service agreements (if any) that exist when your organization provides billing service for the **Service Provider**. You can click the Go To button on a row to transfer to the sub service agreement. Refer to Sub Service Agreements for more information. Note: you can click the **Create Sub SA** button to create sub service agreements for a service provider.

> **Note**
>
> **Automatic sub SA creation.** The system will automatically create sub service agreements for you. It does this when the Analyze SA Relationships background process (known by the batch control ANLYZSAR) sees that a new SA relationship has been activated.

Service Provider SA Relationship

Use this page to manage a service provider's SA relationships and customers. You can view a given service provider's SA relationships of a given type that is in effect on a specified date. You can expire existing relationships or add new relationships. This page is particularly useful for managing negotiated terms.

Open **Menu**, and then **Customer Information**, and then **Service Provider SA Relationship** to view this information.

Description of Page

Select the **SA Relationship Type**.

If the **SA Relationship Type**'s Service Provider Model is **Market Service Provider**, the following additional inputs are required:

- **Market** identifies the market in which the service provider participates. Selecting a value will enable **Provider Configuration Option**.
- **Provider Configuration Option** specifies the service provider's billing/payment relationship within the specific market. Selecting a value will enable the **Market Relationship Service Provider**.
- **Market Relationship Service Provider** specifies the 'It's Us' service provider, meaning the utility company who is implementing this system. This is either a distributor or a retailer, depending on the utility's role in the market. The selection list consists of service providers with the selected billing/payment relationship for the selected market.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Setting Up Market SA Relationship Options for more details on Service Provider Market Options.

Select the **Service Provider**. If the SA Relationship Type's Service Provider Model is **Market Service Provider**, the service providers to choose from are those that are valid for the selected **Market**, **Provider Configuration Option**, **SA Relationship Type** and **Market Relationship Service Provider**.

Enter an **Effective Date**. This date filters the results so that only those SA relationships that are in effect on this date are shown.

Use **SA Filter** to restrict the SA relationships that are shown in the grid based on criteria of the related master service agreement. The following options are available:

- Address. Use this option to restrict SA relationships to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you do not wish to restrict SA relationships based on service agreement attributes.
- Geographic Type. Use this option to restrict SA relationships to those whose service agreements are linked to premises or service points associated with a given **Geo Type** and **Value**.
- SA Type. Use this option to restrict SA relationships to those whose service agreements are linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the filters.

The **Select All** / **Clear All** buttons are used to select / unselect the SA relationships in the results grid. These buttons are enabled if there is at least one SA relationship in the results grid.

The grid that follows contains a line for each SA relationship that has a status of pending or active that meets the search criteria. Canceled SA relationships are not shown. If an SA relationship has more than one sub-SA, additional lines are used to show the additional sub-SA information. The following information appears in the grid:

- Select box. Use this checkbox to select SA relationships that you want to expire using the **Expire SA Relationship(s)** button.
- Click the go to button to transfer to the Account - Main page where you can perform maintenance functions on the account listed under Account Information.
- The **Account ID** column shows the account of the master SA linked to the SA relationship.
- The **Account Information** column shows information about the account identified by the account ID.
- Click the go to button to transfer to the Premise - Main page where you can perform maintenance functions on the premise listed under Premise Information.
- The **Premise Information** column shows information about the premise of the service point linked to the master SA of the SA relationship. If the master SA is linked to multiple service points, one is selected at random.
- The **Premise ID** column shows the ID of the premise whose premise information is shown.

Copyright © 2026, Oracle and/or its affiliates.

- Click the go to button to transfer to the Service Agreement - Main page where you can perform maintenance on the service agreement listed under SA Information.
- The **SA Information** column shows information about the master SA that the SA relationship is linked to.
- The **SA ID** column shows the ID of the master SA that the SA relationship is linked to.
- Click the go to button to transfer to the Service Agreement - Main page where you can perform maintenance on the service agreement listed under Sub SA Information.
- The **Sub SA Information** column shows information about the sub SA linked to the SA relationship.
- The **Sub SA ID** column shows the ID of the sub SA linked to the SA relationship.
- Click the go to button to transfer to the SA relationship page where you can perform maintenance functions on the SA relationship listed under SA Relationship Information.
- The **SA Relationship Information** column shows information about the SA relationship meeting the search criteria.
- The **SA Relationship ID** column shows the ID of the SA relationship meeting the search criteria.

Following the grid is a section with action buttons that allow you to add or expire relationships for the given service provider:

- Press the **Expire SA Relationship(s)** button to expire the SA relationships that you have selected with the checkbox. You will be asked to confirm the date that you want the selected SA relationships to expire. Each selected SA relationship is expired by creating a new SA relationship without a service provider, i.e., its **Usage** is **Expire SA Relationship**. This new SA relationship will be effective the day after the expiration date you specify. This is because an SA relationship that expires on a given day is deemed to be effective until the end of that day.
- Note that you can only expire SA relationships where the SA type and SA relationship type allow gaps in the relationship.
- Press the **Add SA Relationship** button to add a new SA relationship for the service provider and SA relationship type entered above. Enter the effective date for the new SA relationship and the service agreement that the new SA relationship is for. After confirmation, a new SA relationship is added in a pending status. Note that if there is an existing SA relationship with the same effective date whose **Usage** is **Expire SA Relationship**, this relationship is replaced by a new SA relationship with the given service provider.

Maintaining Premises

The premise record contains geographic information about your service addresses.

For more information about premises and their place in the "V", refer to Understanding The "V". For more information about the relationships between premises, service points, meters and items, refer to An Illustration Of A Premise.

The following methods list the various ways in which premises can be created:

- A premise can be added via the premise replicator.
- A premise can be added when an order is completed.
- A premise can be added using this transaction.

Copyright © 2026, Oracle and/or its affiliates.

The Premise portal presents various types of information about a premise. To access the portal, use the **Main Menu**, and then **Customer Information**, and then **Premise**. You are brought to a query portal with options for searching for a specific premise. Once a premise has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page.

- **Premise.** This zone provides high level information about the premise.
  - The premise's unique identifier (i.e., the **Premise ID**) is a system assigned random number that stays with a premise for life.
  - **Formatting is performed by a plug-in.** The basic information about a premise that appears at the top of this page (and on many other pages in the system) is controlled by a plug-in algorithm on the installation record. Refer to the base package's premise format algorithm (PREM-INFO) for an example. If you prefer a different format, your system administrator should configure the system appropriately.
  - Indicate whether the address is a valid **Mailing Address**. Premises that are valid mailing addresses may be specified as the mailing premise on an account. Refer to Account - Person Information for more information.
  - The **Time Zone** in which the premise is located defaults from your postal defaults.
  - **Landlord Agreement.** Use Landlord Agreement to define the landlord agreement that covers the premise, if any.
    - **Landlord reversion.** When you specify a landlord agreement on a premise, you are telling the system that service associated with the premise's service points should be reverted to the landlord's account when a tenant stops service. The system uses this information during the stop service process to automatically create pending start service agreement(s) for the landlord.
    - **Managing a large number of premises for the same landlord** If the premise management module is not turned off, the premise management page helps you to manage the premises linked to the landlord.
  - **Premise Hierarchy**.
    - If the premise management module is not turned off, you may reference a **Parent Premise** to include this premise in a premise hierarchy.
    - When a premise is part of a hierarchy information about the hierarchy is provided in the **Premise Hierarchy** tab.
  - **Alternate Address.**
    - The option to specify an alternate way to define a premise's address only appears if you have enabled alternate addresses on the installation record. Refer to the description of the Alternate Representation field under Installation - Main for more information.
    - This is typically used in countries that use multiple character sets (e.g., the main address is entered in Chinese, the alternate address is entered in English). When a premise has an alternate address, both the main and alternate addresses can be used to search for a premise.
  - **Miscellaneous** details section.
    - You may indicate if the premise has life support or sensitive load equipment.
    - **A person can also have life support / sensitive load information.** If the equipment is personal (i.e., it travels with the person as they move from premise to premise), you should NOT specify the life support information on the premise.

Copyright © 2026, Oracle and/or its affiliates.

Rather, transfer to the Person page and specify life support information on the person.

    * **Alert and C&C.** If the base package's C1-LSSL-PREM algorithm is enabled on the installation record, and life support / sensitive load information is specified, an alert will appear when the premise is displayed on control central. In addition, if a premise has life support or sensitive load equipment, it is possible for a different credit & collection severance process to be kicked off if the account associated with the premise has overdue debt. Refer to Designing Your Severance Procedures for more information.

    *

    ,

  – **Geographic Data** section

    * **Formatting is performed by a plug-in.** The format that is applied to a **Geographic Value** is controlled by the algorithm that is plugged in on the respective Geographic Type. If you prefer a different format, your system administrator should configure this algorithm appropriately. Note, algorithms of this type will NOT convert the input value into the relevant format (i.e., you must enter the information in the exact format dictated by the algorithm).

    * **Default note.** A premise's geographic data defaults from your postal default information. Refer to Setting Up Premise & Service Point Postal Defaults for more information. If you change the premise's postal code, the system will default geographic values based on the new postal code.

  – At the bottom of this zone is a tree that shows the various objects linked to the premise. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

- **Premise Characteristics.** Characteristics may be used to control taxation and other rate options that differ based on geography. This zone shows existing premise characteristics. The collection is maintained using the Edit action for users that are allowed to update this premise.

  – Many charges that appear on bills associated with consumption at this premise are controlled by the premise's characteristic values (e.g., the taxing city defines the city tax percent applied to the premise's consumption). Refer to An Illustration Of A Bill Factor And Its Characteristics for more information.

  – Characteristics also matter if your rates were set up to convert measured consumption into some other unit of measure before charges can be calculated (e.g., converting a cubic foot of gas to its therm value). Refer to All Calculation Rule-Based Rates Share a Common Structure for more information about pre-processing calculation groups.

  – You can only choose characteristic types defined as permissible on the premise record. Refer to Setting Up Characteristic Types & Their Values for more information.

  – The characteristic effective date defaults from the Installation Record when you are adding a new premise. The effective date defaults to the current date when you are changing an existing premise.

  – **Default note.** A premise's characteristics default from your postal default information. Refer to Setting Up Premise & Service Point Postal Defaults for more information. If you change the premise's postal code, the system will default characteristic values based on the new postal code.

- **Premise Attachments.** This zone maintains and displays attachments related to the premise.. You may use the zone header 'Add' action to attach more documents to this record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Customer History** tab page.

- **Premise Customer History.** This zone lists the current and past customers at this premise.

The following zones may appear as part of the portal's **Usage History** tab page.

- **Premise Usage History.** This zone allows you to query and review usage history at the premise.

The following zones may appear as part of the portal's **Activity History** tab page.

- **Premise Activity History.** This zone lists a history of various types of activities at the premise.
- **Premise Activity Overview.** This zone is hidden by default. It provides an overview of field work externally managed by the Service Order Management product. The zone is visible when explicitly enabled via the
  "Cross Product Configuration"

  integration related master configuration.
- **Service Order Management Information .** This zone is hidden by default. It provides further details on a record selected from the 'Premise Activity Overview" zone.

The following zones may appear as part of the portal's **Premise Hierarchy** tab page.

- **Premise Hierarchy.** This zone lists all parents of the current premise and its immediate children.
- **Child Premises.** This zone lists all children of a premise selected in the 'Premise Hierarchy' zone.

The **Additional Information** tab page is hidden by default. It appears only when a custom zone is included as part of its definition. This tab is mainly provided as a way to preserve custom zones that your organization may have configured on the Premise Portal tab of the legacy Premise page.

Actions

The page supports the following special actions:

- **Main Address Update.** Clicking on the icon adjacent to the address launches a drawer that allows you to review and edit the address.
  - The address's constituent fields vary based on the **Country**. Please refer to the Country page for more information.
  - **Default note.** A premise's **state**, **city**, **county**, **division**, **characteristics**, **trend area** and **geographic data** default from your postal default information. Refer to Setting Up Premise & Service Point Postal Defaults for more information. If you change the premise's postal code, the system will default geographic values based on the new postal code.
  - Use **CIS Division** to define the jurisdiction in which the premise is located if the premise type allows CIS Division. This defaults based on the **Country** and the **Postal**, but can be overridden here. If your implementation has set the Installation Option Control By CIS Division flag, the values are restricted to the user's CIS Division(s). See Users and CIS Division for more information. The value is defaulted to user's primary CIS Division, but that can be overridden by the postal defaults
- **Alternate Address Update.** When applicable, clicking on the icon adjacent to the address launches a drawer that allows you to review and edit the address.

Copyright © 2026, Oracle and/or its affiliates.

  - The address's constituent fields vary based on the **Country**. A premise's alternate address must be located in the same country as its main address. Please refer to the Country page for more information about address constituents.

- **Address Validation.** The 'Validate' button is only available if you have set the Allow Address Validation option type to Y on the General System Configuration feature configuration. Once clicked, the Address Validation Script on the General System Configuration feature configuration will be invoked to validate the entered address. Refer to Implementing Address Validation for more information.

> **Note**
>
> If you have enabled the address validation action, adding a new premise is directed to the legacy page for technical reasons. The ability to validate the address using a BPA script at creation time cannot be supported via the portal at this time.

- **Swap Main / Alt Address.** This button becomes enabled when you've entered an alternate address. When clicked, the contents of the main address are swapped with the alternate address.
- **Premise (Legacy).** The Premise portal replaces the legacy Premise page. This action allows you to navigate to the legacy page as needed.

Maintaining Service Points

Refer to Maintaining Service Points in the Device Installation chapter for information about the **Service Point** zone and maintaining service points.

Service point records contain information describing the services supplied to a premise.

The following methods list the various ways in which service points can be created:

- A service point can be added when a premise is created using the premise replicator.
- A service point can be added when an order is completed.
- A service point can be added using this transaction.

> **NOT_SUPPORTED**
>
> For more information about service points and their place in the "V", refer to Understanding The "V". For more information about the relationships between premises, service points, meters and items, refer to An Illustration Of A Premise.

Sidebar Portal

The Sidebar is a portal that always appears on the Oracle Utilities Customer application desktop. Its zones contain tools and data that are useful regardless of the object being displayed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **User configurable.** Refer to Each User Can Customize Which Zones Appear for information about a user can configure which zones appear.
>
> **Global context**. The system automatically refreshes the values saved in global context with information about the object that appears in Object Display Area. Various zones available in the sidebar portal have been designed to display data related to a person id, account id or premise id, which are populated by the base global context algorithm C1-GLBL-CTXT.

The contents of this section describe the zones that are available on this portal.

> **Note**
>
> **Other zones exist.** The zones described below are unique to Oracle Utilities Customer applications. Please see Core Sidebar Zones for a description of other zones that can appear on the sidebar.

- Current Context Zone
- Customer Contact Zone
- Financial Information Sidebar Zone
- Alert Sidebar Zone
- Customer Context Zone
- Premise Context Zone

Current Context Zone

The **Current Context Zone** contains basic information about the customer on which you are working.

A maximum of six rows may appear:

- The first row contains a person context menu and the person's name. You can click on the person's name to transfer to the Person - Main page. Note, the person's name is formatted by a plug-in algorithm on the installation record. Refer to the base package's name format algorithm (PERS-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.
- The second row contains an account context menu and the account's:
  - ID and check digit
  - The primary name of the main customer
  - Customer class
- The third row contains a premise context menu and the premise's address and premise type. Note, the address information is formatted by a plug-in algorithm on the installation record. Refer to the base package's premise format algorithm (PREM-INFO) for an

Copyright © 2026, Oracle and/or its affiliates.

example. If you prefer a different format, your system administrator should configure the system appropriately.

- The fourth row contains a service point context menu and the service point's information; including:
  - Premise address
  - Service point type
  - Measurement cycle and route
  - Status of the service point
- The fifth row contains a device context menu and the device's information; including:
  - Badge number
  - Device type
  - Device installation date/time
  - Status of the device
- The last row contains a measuring component context menu and the measuring component's information; including:
  - The associated device's badge number
  - Measuring component type

> **Note**
>
> If a premise has multiple service points, only one will be displayed in the Current Context Zone.

Customer Contact Zone

The **Customer Contact Zone** has three purposes:

- It shows the age of the **Last** customer contact associated with the person displayed in the Current Context zone. The word Today is shown if the last contact was on the current date. The word Yesterday is shown if the last contact was on the current date - 1 day. If neither is applicable, the number of days old is shown.
- It shows the name of the person who added this contact.
- It greatly simplifies the addition of a new customer contact. To add a new contact, simply select the **Type** of customer contact, enter a brief **Comment**, and press the **Add Contact** button.
- After the contact is added, you can click the hyperlink in the **Last** area to go to the newly added contact where you can set up reminders or indicate the contact should remain open until future resolution.

Financial Information Sidebar Zone

Copyright © 2026, Oracle and/or its affiliates.

The **Financial Zone** that appears on the sidebar contains the same information that is described in the Financial Information Zone on the account information portal.

Alert Sidebar Zone

The **Alert Zone** that appears on the sidebar contains the same information that is described in the Alert Zone on the account information portal.

Customer Context Zone

Along with the Premise Context Zone, the Customer Context Zone is an alternative to the Current Context Zone. It contains basic information about the customer on which you are working. When a person or account is in context, the following information appears:

- Customer Name
- Customer insights (shown as badges).

> **Note**
>
> For more information about customer insights, see Contextual Insights

- Person Contact Information
- Primary Person Identifier
- Account ID. When the person in context is the main customer on multiple accounts, this will appear as a drop down, allowing for selection and switching between these accounts in context.
- Current Balance
- Aged Debt Bar. If the account's Current Balance is greater than zero, a colored bar is displayed adjacent. This bar provides information about the age of the account's debt.
- CIS Division. This appears only when multiple divisions exist.
- Customer Class
- Related Financially Responsible Persons

> **Note**
>
> The base package delivers this zone configured on the Sidebar Portal but with the **Display** turned off by default. To enable the Customer Context zone, navigate to your User - Portal Preferences, open the Sidebar portal settings and turn on the **Display** option for the Customer Context zone.

Premise Context Zone

Along with the Customer Context Zone, the Premise Context Zone is an alternative to the Current Context Zone.

The zone contains a tree that shows the various objects linked to the premise in context as well as any additional premise(s) associated to the account in context. You can use this tree to

Copyright © 2026, Oracle and/or its affiliates.

both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

> **Note**
>
> Refer to the C1-PremiseTree tree definition for more information about the premise tree node definitions and hierarchy.

Moving Service options are used to initiate start, stop, or transfer service requests.

> **Note**
>
> The base package delivers this zone configured on the Sidebar Portal but with the **Display** turned off by default. To enable the Premise Context zone, navigate to your User - Portal Preferences, open the Sidebar portal settings and turn on the **Display** option for the Premise Context zone.

Setting Up Bill Print Groups

Bill print groups allow you to categorize an account's service agreements into groups for bill print purposes.

> **Note**
>
> **Bill print groups are optional.** Typically, only accounts with many SAs will use bill print groups because the standard bill print priority is sufficient for accounts with a limited number of SAs. Refer to SA Type - Billing for more information about the standard bill print priority.

Let's use an example to clarify the bill print group concept. Consider a local government's account. This account would have many service agreements (some for the police department, others for the court system, others for the department of public works, etc.).

If you don't create a bill print group for this customer, their bill segments will be printed in the order dictated by each segment's service agreement's SA type's bill print priority. If you create a bill print group for this customer, you can define the customer's desired categories (e.g., police, courts, DPW, etc.) and then link each of the account's SAs to the appropriate category.

> **Note**
>
> **Nomenclature.** We refer to the categories under a bill print group as "subgroups". A bill print group can have an unlimited number of subgroups. When you create a subgroup, you define its relative bill print priority and the associated verbiage to print on the bill (assuming you print information about the subgroup on the bill).

Copyright © 2026, Oracle and/or its affiliates.

We'd like to highlight the following characteristics of bill print groups:

- A bill print group is associated with a specific account. If multiple customers have the exact categorization preferences, you will have to set up multiple bill print groups.
- Over time, a customer could have many bill print groups. This would happen if a customer's bill categorization preferences change over time. We'd like to stress that while an account can have multiple bill print groups, only one will be used by the bill print process (the one effective on the bill date).
- Bill print groups only affect printed bills. Bill print groups do not affect the order in which service agreements appear on the bill maintenance transaction.
- If new service agreements are added after a bill print group is set up for an account, they will not be linked to a subgroup. If you neglect to link the new service agreements to one of the bill print group's subgroups they will be printed under the "default" subgroup (every bill print group must have one default subgroup to cater for this situation).

> **Warning**
>
> While the system supports the definition of bill print groups and the categorization of an account's service agreements into the various subgroups, the base package's bill print extract does NOT take advantage of this information.

The topics in this section describe how to set up a bill print group and how to link an account's service agreements to its subgroups.

- Bill Print Group - Main
- Bill Print Group - SA Sub Group

Bill Print Group - Main

Open **Menu**, and then **Customer Information**, and then **Bill Print Group**, and then **Search** to maintain an account's bill print group and subgroups. After defining this information, transfer to the **SA Sub Group** page to link the account's service agreements to the subgroups.

Description of Page

The **Bill Print Group ID** is displayed on every page. This value only appears after the bill print group exists on the database. The ID is a system-assigned random number that stays with a bill print group for life.

Enter the **Account ID** associated with the bill print group.

Use **Effective Date** to define when the bill print group is effective for the account. This date is important as it allows a customer to change their preferences over time. For example, if the customer wants to change the number of subgroups on a given date, you would simply add a new bill print group effective on this date and then define the new subgroups (and link each service agreement to one of the subgroups).

Use **Status** to define if the bill print group is Active or Inactive. You would only use the Inactive value if the bill print group is no longer needed (as there is no delete action on this transaction).

Enter a brief **Description** of the bill print group.

Use **Comments** to describe anything special about the bill print group.

Copyright © 2026, Oracle and/or its affiliates.

The grid contains the bill print group's subgroups. The following information should be defined for each subgroup:

- **Bill Print Sub Group.** This is the unique identifier of the bill print subgroup.
- **Sub Group Bill Print Priority.** This is the relative print priority of the subgroup in respect of the other subgroups.
- **Use as Default.** Turn on this switch for the default sub group. A bill print group must have one and only one default subgroup. The default group is used by the bill print process if it detects a service agreement that is not linked to a subgroup (it links this service agreement to the default subgroup).
- **Sub Group Description.** This is a brief description of the subgroup.
- **Description on Bill.** This is the verbiage that will print on the bill (assuming you print something on your bills for the subgroup).

After defining the bill print group's subgroups, navigate to the **SA Sub Group** tab to link the account's service agreements to the subgroups.

Bill Print Group - SA Sub Group

This page is used to link the account's service agreements to one of the bill print group's subgroups.

Open **Menu**, and then **Customer Information**, and then **Bill Print Group**, and then **Search** and navigate to the **SA Sub Group** page to maintain this information.

Description of Page

The **Bill Print Group's ID**, **Account ID** and **Effective Date** are displayed at the top of the page.

The filters control the service agreements that appear in the **SAs in Bill Print Group** grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:
  - Address. Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - All. Use this option if you do not wish to restrict service agreements based on service agreement attributes.
  - Bill Print Sub Group. Use this option to restrict service agreements to those that are linked to a given **Bill Print Sub Group**.
  - Geographic Type. Use this option to restrict service agreements to those linked to service points associated with a given **Geo Type** and **Value**.
  - SA Type. Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.
- Use **Status Filter** to restrict the service agreements based on their status. The following options are available:
  - All. This option shows all service agreements regardless of status.
  - Refer to Service Agreement Lifecycle for the various status values.

Copyright © 2026, Oracle and/or its affiliates.

Don't forget to click the search button after changing the filters.

The **SAs in Bill Print Group** contains an entry for every non-cancelled service agreement linked to the account that is linked to one of the bill print group's subgroups. The following information appears in the grid:

- Use **Bill Print Sub Group** to define the subgroup associated with the service agreement.
- Use **Sequence** when there is more than one service agreement in the subgroup. The sequence controls the order in which the service agreement's financial information appears on the bill.
- The **SA Information** column provides a brief description of the service agreement.
- The **Service Agreement** column contains the unique identifier of the service agreement.
- The **Premise Information** column contains the characteristic premise associated with the bill segment's service agreement.

The next set of filters control the service agreements that appear in the **SAs Not in Bill Print Group** grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:
  - Address. Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - All. Use this option if you do not wish to restrict service agreements based on service agreement attributes.
  - Geographic Type. Use this option to restrict service agreements to those linked to service points associated with a given **Geo Type** and **Value**.
  - SA Type. Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.
- Use **Status Filter** to restrict the service agreements based on their status. The following options are available:
  - All. This option shows all service agreements regardless of status.
  - Refer to Service Agreement Lifecycle for the various status values.

Don't forget to click the search button after changing the filters.

The **SAs Not in Bill Print Group** contains an entry for every non-cancelled service agreement linked to the account that is NOT linked to one of the bill print group's subgroups. The following information appears in the grid:

- Use **Bill Print Sub Group** to define the subgroup associated with the service agreement.
- Use **Sequence** when there is more than one service agreement in the subgroup. The sequence controls the order in which the service agreement's financial information appears on the bill.
- The **SA Information** column provides a brief description of the service agreement.
- The **Service Agreement** column contains the unique identifier of the service agreement.
- The **Premise Information** column contains the characteristic premise associated with the bill segment's service agreement.

Copyright © 2026, Oracle and/or its affiliates.

Landlord Reversion

Landlord reversion occurs when a landlord would like to have service reverted to their name when a tenant moves out. The landlord defines their reversion preferences using a landlord agreement.

- A landlord's preferences can differ for each type of service. For example, you can indicate that a specific landlord always wants electric and gas service reverted to their account when a tenant stops service. However, they do not want water service to revert.
- A landlord's preferences can also be seasonal. For example, you can indicate water service should only revert to the landlord's account during the winter.

The account to use for the reversion of service is defined on the landlord agreement.

Assuming that you have plugged-in a landlord reversion algorithm on SA type, the system uses the information defined on landlord agreement to create a pending start service agreement(s) for the landlord when service for the existing tenant is stopped.

The following examples illustrate how your organization's landlord reversion algorithm might handle various scenarios for a premise that defines a landlord agreement that indicates service should be reverted:

- When the existing customer calls to stop service and no new customer is scheduled to move in, a pending start service agreement is created for the landlord.
- When a new customer calls in to start service
  - If the start date matches the outgoing customer's start date, the pending start service agreement for the landlord is canceled
  - If the start date is earlier than the outgoing customer's stop date, the pending start service agreement for the landlord is canceled and the stop date of the outgoing customer is changed to match the new date
  - If the start date is later than the outgoing customer's stop date, the pending start service agreement for the landlord is updated with a stop date that matches the new customer's start date
- If changes are made to start or stop dates for the tenants' service agreements, a service agreement for the landlord may need to be created or canceled
  - If the change in dates causes a gap in service that did not exist before, a landlord service agreement is created to cover the gap
  - If the change in dates means that the tenants are now starting and stopping on the same date (and there used to be a gap), a landlord service agreement is canceled
- If the new customer cancels their pending stop, a landlord service agreement is created (if one did not exist) or the landlord SA's stop date is removed (if the new customer's start date resulted in a gap that was being covered by the landlord.
- Setting Up Landlord Agreements

Setting Up Landlord Agreements

The landlord agreement page is used to maintain a landlord's service reversion preferences.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **A landlord could have many landlord agreements.** A landlord will have at least one landlord agreement. However, if the landlord has different service reversion preferences for different properties, you can create multiple landlord agreements - one for each set of preferences. After a landlord's reversion preferences have been defined, link the landlord agreement record to the respective premises.

Use **Menu**, and then **Customer Information**, and then **Landlord Agreement**, and then **Search** to maintain a Landlord Agreement.

Description of Page

Landlord and **Landlord ID** only appear after the landlord is added to the database. **Landlord ID** is a system-assigned random number that stays with the landlord agreement for life. **Landlord** is the description of the landlord agreement.

If the premise management module is not turned off, a message is displayed indicating the number of premises linked to this landlord. Clicking the hyperlink message text transfers you to the premise management page, which helps you to manage the premises linked to the landlord.

Enter a **Description** of the landlord agreement.

Use **Account ID** to define the account to use when service is reverted to the landlord. This should be one of the landlord's accounts.

The grid that follows contains one row for each type of service to be reverted to the landlord's account. Any service point whose service type is not referenced in this scroll will NOT revert to the landlord's account when a tenant stops service. For each row define the following:

- **Service Type** defines the type of service point that will revert to the landlord's account when a tenant stops service. Note, a service point inherits its service type from its SP Type.
- Turn on **Seasonal** if the service should only revert to the landlords account during a specific season of the year. When turned on, you must define the **Start Month/Day** and the **End Month/Day** of the effective season.

> **Note**
>
> The month and day must be entered in the format MM/DD. For example, if a season starts on March 1, you must enter the value as 03/01.

Using The Premise Replicator

You use the premise replicator to create many copies of a premise and its service points.

Perform the following steps to replicate a premise:

- Add a premise.
- Add one or more service points to the premise.

Copyright © 2026, Oracle and/or its affiliates.

- Use the premise replicator to create copies of the premise and its service points.

The topics in this section describe how to use the premise replicator.

- Premise Replicator - Main Information
- Premise Replicator - Premise
- Premise Replicator - Premise SPs

Premise Replicator - Main Information

You use this page to create many copies of a given premise and its service points.

> **Warning**
>
> The system warns you if you attempt to create. However, this is just a warning as you may want to generate similar premises and then update them with, for example, a unique apartment number on Premise - Main Information.

Use **Menu**, and then **Customer Information**, and then **Premise Replicator** to open this page.

Description of Page

Choose the **Premise** that serves as the template premise.

> **Note**
>
> **Recommendation.** Carefully verify the template premise using the other pages in this page before you save the replicated premises. After you save the replicated premises, any corrections could prove time consuming.

The two sections at the top of the page determine the number of replica premises to be created and the address information that will be set up for each premise. The left section is used to define how the system will create the part of the address that is *different* for each replica premise. The right section defines the parts of the address that will be the *same* for each replica premise.

In the left section:

Use the **Replicate** option to control how the position of the replicated number in the first address line:

- Choose Street Number if the replicated number should prefix the **Base** value in the first address line of the premise. For example, if you have a **Start #** of 1020 and a **Base** value of Main St and you choose the Street Number option, the first replicated premise will be 1020 Main St.
- Choose Apartment Number if the replicated number should suffix the **Base** value in the first address line of the premise. For example, if you have a **Start #** of 1020 and a **Base** value of 101 Main St, Apt. and you choose the Apartment Number option, the first replicated premise will be 101 Main St, Apt. 1020 .

Indicate the **Number Of Premises** you wish to create.

Copyright © 2026, Oracle and/or its affiliates.

Use **Base** to define the information that will appear at the end of the first address line on each of the replicated premises. Use **Start #** to define the number assigned to the first copy of the premise. Use **Increment** to define the value to increment successive numbers by. The first and last address lines to be created by the replicator are displayed below.

For example, if the first address you want to create is 101 Derby St, and the last address is 199 Derby and you want to do the odd numbered side of the street, you'd enter the following parameters:

Number of Premises would be 50

Address Suffix would be Derby St

Start From would be 101

Increment by would be 2

The right section displays the rest of the address information that will be copied onto each replicated premise. This information is defaulted from the template premise that you selected. The Address 1 field is not shown because you define the information to be entered there in the previous section, as explained above. The remaining fields, however, may be edited if so desired.

After entering the parameters, click the **Replicate** button to generate your new premises. You can use the sections at the bottom of the page to view the premises that will be created when you click save. The left section will show how Address Line 1 will look on the first 10 premises. If you are creating more than 10 premises, the right section will show how Address Line 1 will look on the last 10 premises. If everything looks clean, click save.

Premise Replicator - Premise

The Premise page is a display only page provided to let you confirm that the premise you selected on the add dialog is the one to be used as the template by the replicator. Use **Menu**, and then **Customer Information**, and then **Premise Replicator** and navigate to the **Premise** page.

Description of Page

This is a display only page that displays key information about the premise being replicated.

Premise Replicator - Premise SPs

This is a display only page provided to let you confirm that the premise you selected on the add dialog is the one to be used as the template by the replicator. Use **Menu**, and then **Customer Information**, and then **Premise Replicator** and navigate to the **Premise SPs** page.

Description of Page

This is a display only page that displays the details about the template premise's service points.

Copyright © 2026, Oracle and/or its affiliates.

Start/Stop

The topics in this section describe how to start and stop service.

> **Warning**
>
> Please see Order versus Start/Stop for a discussion of the pros and cons of using the order transaction versus the start/stop transaction to start service.

- The Big Picture Of Starting Service
- The Big Picture Of Stopping Service
- Start/Stop Maintenance

The Big Picture Of Starting Service

Almost all requests to start service involve a conversation between a customer and a customer service representative (CSR). During this conversation, the following is defined:

- WHO will pay for the service (i.e., the account).
- WHERE the service is consumed (i.e., the premise). Note, this is only required for premise-oriented services.
- WHEN the service is requested (i.e., the start date).
- WHAT TYPE of service is desired (i.e., the service agreement and service points).
- HOW MUCH will be paid for the service (i.e., the rate and contract riders).

After the conversation is complete, one or more service agreements in the state of pending start will exist. If the service agreement uses service points, field activities may also be created.

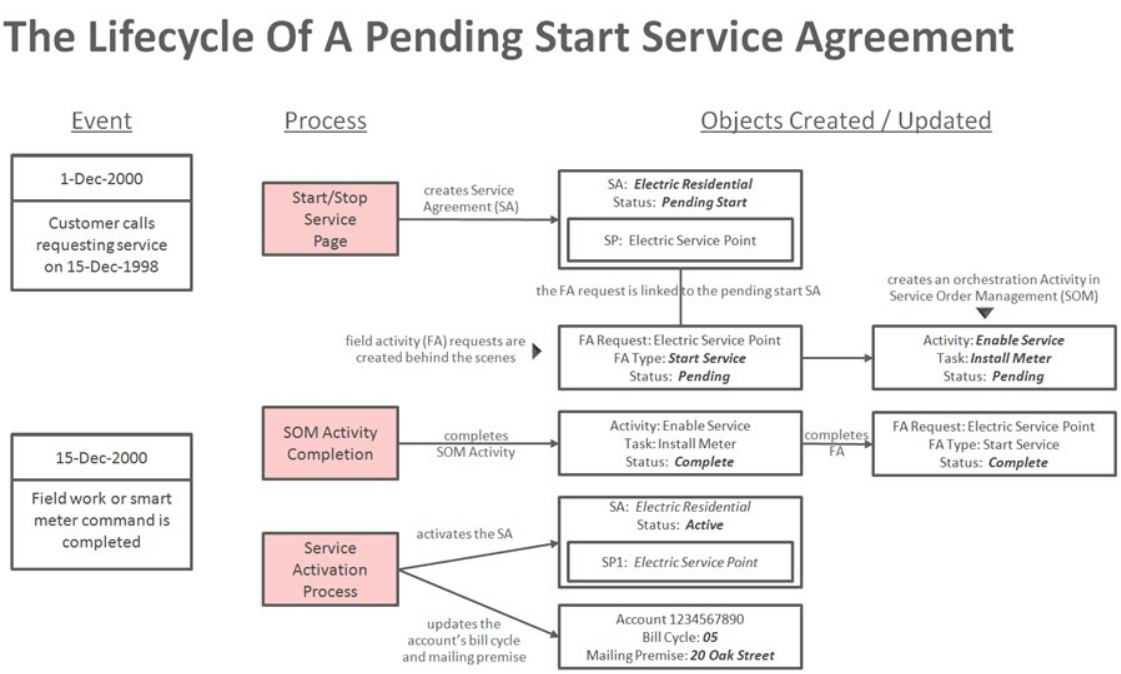
In this section, we provide background information about the start process. For information about the page used to start service, refer to Start Stop Maintenance.

- The Lifecycle Of A Pending Start Service Agreement
- Starting Service and Field Activities
- Activating Pending Starts
- What Happens After A SA Is Started?

The Lifecycle Of A Pending Start Service Agreement

When you request a new service for a customer, a pending start service agreement is created. The following diagram illustrates the lifecycle of a typical pending start service agreement:

Copyright © 2026, Oracle and/or its affiliates.



The above diagram is referenced in the topics that follow.

Starting Service and Field Activities

When you initiate a request to start service, the system creates a service agreement in the state of pending start. If the service agreement is service point-oriented, pending field activity requests necessary to start will be created at some point prior to the service agreement being activated. The question is when? The contents of this section provide information about when they are created and what types of activities are created.

> **NOT_SUPPORTED**
>
> When you enable the Control Central alert algorithm, C1-PEND-STRT, an alert displays when there is a pending start service agreement linked to the account. This algorithm is plugged-in on the installation record.

- What Type Of Start Field Activities Are Created
- When Start Field Activities Are Created

What Type Of Start Field Activities Are Created

The type of customer event (such as, start service, stop service, and cut for non-payment) dictate the type of field activity request generated. If you've setup your system properly, the system creates the appropriate orchestration activity for the field activity request, based on the type of service point, state of the service (such as, disconnected-no meter and connect-meter installed) at the service point, state and capabilities of the device;. You shouldn't have to be aware of field activity requests and orchestration activities.

Copyright © 2026, Oracle and/or its affiliates.

In addition to being linked to the service points, these field activity requests are also linked to the service agreement / service point relationship so that the system can automatically activate the service agreement when the field activity requests are complete.

> **Note**
>
> **Back-to-Back.** We use the term **back-to-back** to describe the situation when a single field activity supports both the stop and start service requests. The fieldwork algorithm delivered with the base package sets up a back-to-back situation by default when it is aware of both the start AND stop customers at a premise.

> **NOT_SUPPORTED**
>
> For more information about the field activities associated with your SP types, refer to Setting Up Field Activity Type Profiles. Refer to The System Activates Most Service Agreements Behind-The-Scenes for information about how the system monitors field activities linked to a service agreement.

When Start Field Activities Are Created

Field activity requests will be created when the pending start service agreement is created if the following conditions are true:

- The Create Field Activity Start Stop switch is enabled on the installation record. Refer to Installation Options - Main for more information.
- The appropriate create fieldwork creation algorithm is defined on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

If the Create Field Activity Start Stop switch is NOT enabled on the installation record, field activity requests will NOT be created when the pending start is saved. Rather, a background process (PSASPM) creates the field activity requests shortly before the start date. The number of days before the start date on which field activity requests will be created is defined on the service agreement's SA type's fieldwork creation algorithm.

> **Note**
>
> **A fail safe.** If any person or process attempts to activate a pending start service agreement and neither of the above events has taken place, field activities will be created when the user attempts to activate the pending start. The creation of the field activities will prevent the service agreement from being activated as described under The System Activates Most Service Agreements Behind The Scenes.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Field activity requests will only be created if you have defined the appropriate field work creation algorithm on the service agreement's SA type. The algorithm that is supplied with the base package checks if the service agreement / service point already has a start field activity request, if so, field activity requests will not be created (the field work has already been initiated).

Activating Pending Starts

When you initiate a request to start service, the system initiates a service agreement in the state of pending start. A pending start service agreement remains in this state until everything necessary to start service is defined in the system. At that time, the service agreement can be activated.

- What Is Necessary To Start Service?
- The System Activates Most Service Agreements Behind-The-Scenes
- What Happens At Activation Time?

What Is Necessary To Start Service?

The service agreement's SA type defines what is necessary to start service:

- For SA types that have service points, all field activity requests linked to the service point must be complete before the service agreement can be activated.
- For SA types that don't have service points (e.g., charitable donations, deposits, one-time charges), everything necessary to activate the service agreement is defined when you initiate the request to start service. These types of service agreements can be activated immediately.

The System Activates Most Service Agreements Behind-The-Scenes

Service agreements are activated by one of the following methods:

- The SA activation background process (SAACT or C1-ACTSA) automatically activates pending start service agreements once it detects that everything necessary to start service is defined in the system. This is the most common method for activating service agreements. This process runs as often as your organization desires (but it should run at least daily).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Sub SAs are not handled by the SA activation background process.** If your organization uses the SA relationship functionality and you have service providers that necessitate sub service agreements, you should be aware that the SA activation background process only activates normal and master SAs; sub SAs are activated by the analyze SA relationships background process (described in the next bullet point).

- If your organization uses the SA relationship functionality and you have service providers that necessitate sub service agreements, the analyze SA relationships background process (ANLYZSAR) automatically activates pending start sub SAs if their master SA is active.
- You can automatically activate a service agreement when it's created if you plug-in an SA creation algorithm of type SACR-AT on the SA type. Because everything necessary to start service must be defined in the system before a service agreement can be activated, this method only works for SA types that don't have service points (e.g., charitable donations, deposits, one-time charges, etc.). If this algorithm is not plugged in on an SA type, service agreements of that type are activated by the SA activation background process or manually.
- You can activate a service agreement manually (assuming everything that's necessary to activate service is in place). Simply open Service Agreement - Main and click the **Activate SA** button. This is only necessary if you do not want to wait until the SA activation background process next runs.

What Happens At Activation Time?

When a service agreement is activated, the following takes place:

- The service agreement's state is changed from pending start to active.
- If the account's bill cycle is not protected and the service agreement has a metered service point, the bill cycle will be changed to reflect the service point's *measurement cycle's* bill cycle.
- If the account's mailing premise is not protected and the service agreement has a service point whose premise is a mailing premise, the mailing premise is changed to reflect the service point's premise.
- If the account's division is not protected, the division is changed to reflect the service agreement's SA type's division.
- If there is a start adjustment defined on the service agreement's start option, an adjustment is levied.
- The main customer's billing address is changed to reflect the value specified on the Start/Stop page (and this may involve changing the main customer's override mailing address on their person record).

What Happens After A SA Is Started?

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Lifecycle Of A Service Agreement for more information about a service agreement's lifecycle.

The Big Picture Of Stopping Service

To stop service you must define:

- WHO pays for the service (i.e., the account).
- WHEN service should stop (i.e., the stop start date).
- WHAT services to stop (i.e., the premise or specific service agreements).

After the above is defined in the system, the state of one or more service agreements will be changed to pending stop. If the service agreement uses service points, field activities may also be created.

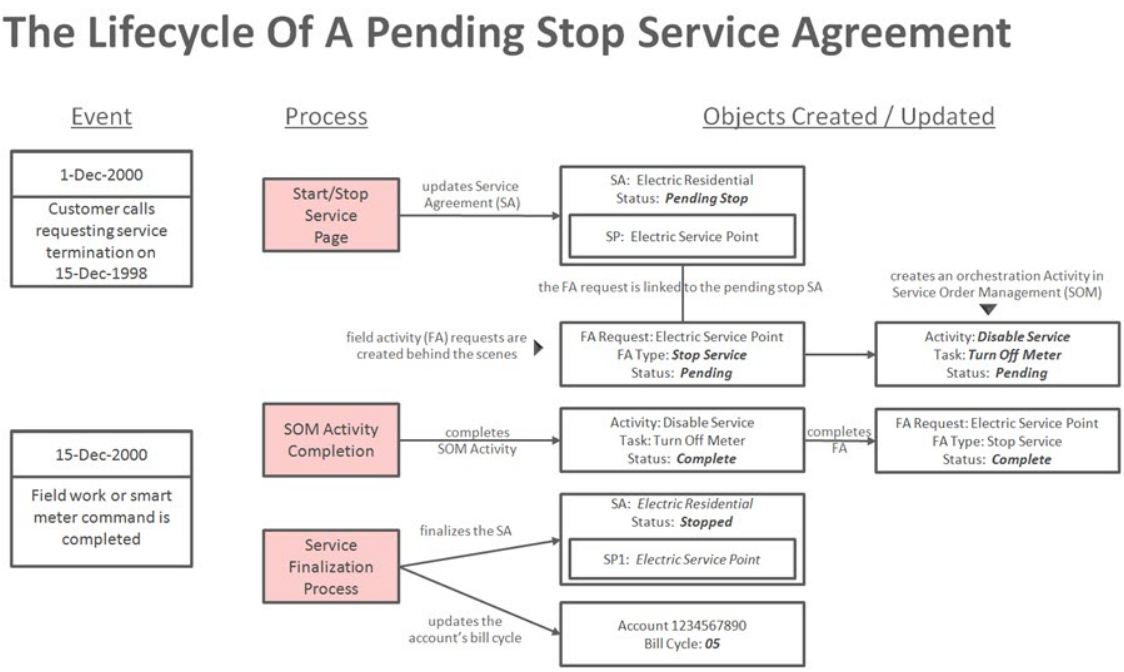
In this section, we provide background information about the stop process. For information about the page used to stop service, refer to Start Stop Maintenance.

- The Lifecycle Of A Pending Stop Service Agreement
- Stopping Service and Field Activities
- Finalizing Pending Stops
- What Happens After A Service Agreement Is Stopped?

The Lifecycle Of A Pending Stop Service Agreement

When you request to stop service for a customer, the status of an existing service agreement becomes pending stop. The following diagram illustrates the lifecycle of a typical pending stop service agreement:

Copyright © 2026, Oracle and/or its affiliates.



The above diagram is referenced in the topics that follow.

Stopping Service and Field Activities

When you initiate a request to stop service, the system changes the service agreement to the state of pending stop. If the service agreement is service point-oriented, pending field activity requests necessary to stop will be created at some point prior to the service agreement being stopped. The question is when? The contents of this section provide information about when they are created and what types of activities are created.

- What Type Of Stop Field Activities Are Created
- When Stop Field Activities Are Created

What Type Of Stop Field Activities Are Created

The type of customer event (such as, start service, stop service and cut for non-payment) dictate the type of field activity request. If you've setup your system properly, the system creates the appropriate orchestration activity for the field activity request, based on the type of service point, state of the service (such as, disconnected-no meter and connect-meter installed) at the service point, state and capabilities of the device. You shouldn't have to be aware of field activity requests and orchestration activities.

In addition to being linked to the service points, these field activity requests are also linked to the service agreement / service point relationship so that the system can automatically stop the service agreement when the field activities are complete.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Back-to-Back.** We use the term **back-to-back** to describe the situation when a single field activity supports both the stop and start service requests. The fieldwork algorithm delivered with the base package sets up a back-to-back situation by default when it is aware of both the start and stop customers at a premise.

> **NOT_SUPPORTED**
>
> For more information about the field activity requests associated with your SP types, refer to Setting Up Field Activity Type Profiles. Refer to The System Finalizes Most Service Agreements Behind-The-Scenes for information about how the system monitors field activity requests linked to a service agreement.

When Stop Field Activities Are Created

Field activity requests will be created when the pending stop service agreement is recorded when the following conditions are true:

- The Create Field Activity Start Stop switch is enabled on the installation record. Refer to Installation Options - System for more information.
- The appropriate create fieldwork creation algorithm is defined on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

If the Create Field Activity Start Stop switch is NOT enabled on the installation record, field activity requests will NOT be created when the pending stop is saved. Rather, a background process (PSASPM) creates the field activity requests shortly before the stop date. The number of days before the stop date on which field activities will be created is defined on the service agreement's SA type's fieldwork creation algorithm.

> **Note**
>
> **More information about the Create Field Activity Start Stop switch on the installation record.** When this switch is on (and the user doesn't override it), pending field activity requests will be created when a start or stop is recorded (as opposed to shortly before the start/stop date).
>
> **A fail safe.** If any person or process attempts to finalize a pending stop service agreement and neither of the above events has taken place, field activity requests will be created when the user attempts to finalize the pending stop. The creation of the field activity requests will prevent the service agreement from being stopped as described under The System Finalizes Most Service Agreements Behind The Scenes.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Field activity requests will only be created if you have defined the appropriate field work creation algorithm on the service agreement's SA type. The algorithm that is supplied with the base package checks if the service agreement / service point already has a stop field activity request, if so,field activity requests will not be created (the field work has already been initiated).

Finalizing Pending Stops

When you initiate a request to stop service, the system changes the service agreement's state to pending stop. A pending stop service agreement remains in this state until everything necessary to stop service is defined in the system. At that time, the system automatically changes the state of the service agreement to stopped**,** and at that time the service agreement can be final billed.

So, how does the system know that "everything necessary" is in the system? In this section we provide the answer and explain the many other things that may take place at pending stop finalization.

- The System Finalizes Most Service Agreements Behind-The-Scenes
- What Happens At Finalization Time?

The System Finalizes Most Service Agreements Behind-The-Scenes

The SA activation background process (SAACT or C1-STPSA) finalizes a pending stop service agreement when everything necessary is in place. The service agreement's SA Type defines what's necessary:

- For SA Types that don't have service points (e.g., charitable donations, deposits, third party billing), everything necessary to finalize the service agreement is defined when you initiate the request to stop service. These types of service agreements can be finalized immediately.
- For SA Types that have service points, all field activity requests linked to the service point must be complete.

When the SA activation background process detects that everything necessary to stop service is in place, it automatically stops the service agreement. This process runs as often as your organization desires (but it should run at least daily).

> **Note**
>
> **Sub SAs are not handled by the SA activation background process.** If your organization uses the SA relationship functionality and you have service providers that necessitate sub service agreements, you should be aware that the SA activation background process only stops normal and master SAs; sub SAs are stopped by the analyze SA relationships background process.

Copyright © 2026, Oracle and/or its affiliates.

You can stop a service agreement real-time (assuming everything that's necessary to stop service is in place). Simply open Service Agreement - Main and click the **Stop** button.

What Happens At Finalization Time?

When a service agreement is finalized, the following takes place:

- If the service agreement's SA type has a SA Stop algorithm, the algorithm is executed.
- If the service agreement is for a non-billed budget (i.e., the SA type has a special role of Non-billed Budget), the system removes all covered SAs from the Non-Billed Budget. Refer to Stopping Non-Billed Budgets for more information.
- If the service agreement is covered by a non-billed budget, the SA is removed from the Non-Billed Budget. Refer to Stopping a Covered SA for more information.
- When the last active service agreement covered by a non-billed budget is stopped, the Non-Billed Budget will be stopped (i.e., its state will be changed from active to pending stop).
- The service agreement's state will be changed from pending stop to stopped.
- If the service agreement has an active severance process, the process and its severance events will be canceled.
- If the service agreement is linked to an active collection process, the service agreement is "removed" from the collection process. If the collection process has no other active service agreements, it is canceled.
- When the last service agreement linked to an account is stopped, the account's bill cycle will be changed so that the account will be final billed when billing next executes. The system selects the latest open bill cycle where the Freeze and Complete switch value matches the value of the Freeze and Complete switch of the account's current bill cycle. In addition, the account's protect bill cycle flag is set to No so that the next time this account starts service, its bill cycle will be set based on the new service point's service cycle.
- The main customer's billing address will be changed to reflect the value specified on the Start/Stop page (and this may involve changing the main customer's override mailing address on their person record).
- If the service agreement is for a deposit (i.e., the SA type has a special role of Cash Deposit), the system calculates interest through the stop date. In addition, the SA's current balance is synchronized with its payoff balance (using the sync adjustment code defined on the service agreement's SA type).
- When the last active service agreement governed by a deposit class is stopped, all active deposit service agreements governed by the same deposit class will be stopped (i.e., their state will be changed from active to pending stop). These pending stop deposit service agreement will then be stopped (as per the above points).
- When a service agreement is stopped, the system automatically cancels the budget. Refer to Canceling A Budget for more information.

What Happens After A Service Agreement Is Stopped?

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Lifecycle Of A Service Agreement for more information about a service agreement's lifecycle.

Start/Stop Maintenance

You use the start/stop page to both start and stop service.

This page handles a variety of scenarios, including:

- Starting some or all services at a premise. Note, the system will automatically stop the existing customer when you indicate a new customer is starting.
- Stopping some or all services at a premise. Note, the system will automatically revert service to a landlord's account when you indicate a tenant is stopping.
- Starting non-premise-oriented services.
- Stopping non-premise-oriented services.
- Changing the mailing address associated with the account's main customer.

NOT_SUPPORTED

Refer to The Big Picture Of Starting Service for more information about starting service. Refer to The Big Picture Of Stopping Service for more information about stopping service.

The topics in this section describe the start/stop page in detail.

- Start/Stop - Main
- Start/Stop - Start Confirmation
- Start/Stop - Stop Confirmation
- Start/Stop - Address/Phone
- Start/Stop - Pending SA
- Start/Stop - Pending Field Activity
- How To

Start/Stop - Main

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

The **Description of Page** section that appears below simply describes the fields on this page. Refer to How To for a description of how to perform common start/stop functions.

Use **Menu**, and then **Customer Information**, and then **Start/Stop** to open this page.

Description of Page

The **Account ID** and the primary name of the **Account**'s main customer are displayed at the top of the page.

If the total number of pending start/stops for the account does not exceed the start stop details threshold defined in Installation Options, the grid contains an entry for every Pending Start and Pending Stop service agreement linked to the **Account ID**.

If information appears on the right portion of the grid, that means that "reciprocal relationships" exist. The following points describe potential reciprocal relationships:

- A reciprocal will be shown for a Pending Start service agreement if the Pending Start service agreement has at least one service point and if another service agreement is found that satisfies the following criteria:
  - The status of the service agreement is not Canceled or Active ,
  - The service agreement is linked to all of the Pending Start service agreement's service points (and all SA/SP records have the same stop date as defined on the qualifying service agreement).
  - The service agreement's stop date is on/before the start date.
  - The service agreement's SA type has a service type that is the same as the Pending Start service agreement's service type.
- Note: if multiple service agreements meet the above criteria, the latest qualifying service agreement is shown as the reciprocal.
- A reciprocal will be shown for a Pending Stop service agreement if the Pending Stop service agreement has at least one service point and if another service agreement is found that satisfies the following criteria:
  - The status of the service agreement is not Canceled
  - The service agreement is linked to all of the Pending Stop service agreement's service points (and they all have the same start date as defined on the qualifying service agreement).
  - The service agreement's start date is on/after the stop date.
  - The service agreement's SA type has a service type that is the same as the Pending Stop service agreement's service type.
- Note: if multiple service agreements meet the above criteria, the earliest qualifying service agreement is shown as the reciprocal.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If multiple reciprocals exist for a given SA, none of the reciprocals will be shown. In this situation, an appropriate message is displayed highlighting this situation. The existence of multiple reciprocals is rare, but it could happen, for example, if multiple customers are paying for a service point's service (where one customer pays for 60% of the metered consumption and the other pays for the remaining 40%).

The following columns appear in the grid:

Drill Down (button) Click this button to drill down into details of the pending start/stop SA. This button transfers you to the **Pending SA** tab.

Action An icon will be displayed to highlight if the associated SA is a pending start or a pending stop. The start icon (green octagon) appears if the SA is a pending start. The stop icon (red octagon) appears if the SA is a pending stop. If there is a reciprocal SA for the pending **start**, the stop icon appears adjacent; clicking it will cause the SA grid to scroll to the left (so that you can see the pending stop). If there is a reciprocal SA for the pending **stop**, the start icon appears adjacent; clicking it will cause the SA grid to scroll to the left (so that you can see the pending start).

Date This is the start date for pending start service agreements and the stop date for pending stop service agreements.

Cancel Turn on this switch if you wish to cancel the pending start/stop.

> **Note**
>
> **Field activity requests will be cancelled too.** If you cancel a pending start/stop that has field activity requests, the field activity requests will also be cancelled. Refer to Starting Service and Field Activities for the details about how the system automatically creates the field activity requests to support start service requests. Also, refer to Stopping Service and Field Activities for the details about how the system creates the field activity requests to support stop service requests.
>
> **System-generated reciprocals will be cancelled too.** If you cancel a pending start/stop that has a SYSTEM-GENERATED reciprocal SA, the reciprocal will also be cancelled.

Premise Information If a characteristic premise is linked to the service agreement, summary information about the pending start/stop SA's characteristic premise is displayed.

SA Information Summary information about the pending start/stop SA is displayed.

Action If this column is blank, then there is no reciprocal start/stop for the service agreement whose details appear to the left. If this column is not blank, then the information that appears to the right describes the reciprocal SA of the SA to the left. If a reciprocal SA exists, two icons will always appear: if a start and stop combination appears, the SA to the left is a pending stop (and clicking the stop icon will cause the pending stop to appear); if a stop and start combination appears, the SA to the left is a pending start (and clicking the start icon will cause the pending start to appear).

Date This is the start date for pending start service agreements and the stop date for pending stop service agreements.

Copyright © 2026, Oracle and/or its affiliates.

Cancel Turn on this switch if you wish to cancel the pending start/stop.

> **Note**
>
> **Field activity requests will be cancelled too.** If you cancel a pending start/stop that has field activity requests, the field activity requests will also be cancelled. Refer to Starting Service and Field Activities for the details about how the system automatically creates the field activity requests to support start service requests. Refer to Stopping Service and Field Activities for the details about how the system creates the field activity requests to support stop service requests.
>
> **System-generated reciprocals will be cancelled too.** If you cancel a pending start/stop that has a SYSTEM-GENERATED reciprocal SA, the reciprocal will also be cancelled.

Account Info The name of the main customer associated with the reciprocal's service agreement's account is displayed.

Premise Information Summary information about the pending start/stop premise is displayed.

SA Information Summary information about the pending start/stop SA is displayed.

If the total number of pending start/stops for the account exceeds the start stop details threshold defined in Installation Options, the **Total** number of **Pending Starts** and **Total** number of **Pending Stops** is displayed together with a grid containing a summary of the pending start and pending stops.

The grid contains an entry for every group of Pending Start / Pending Stop service agreement linked to the **Account ID** with the same pending start / pending stop date, action, premise, and CIS division/SA Type.

The following columns appear in the grid:

Date This is the start date for pending start service agreements and the stop date for pending stop service agreements.

Action An icon will be displayed to highlight if the associated SA(s) are pending start or pending stop. The start icon (green octagon) appears if the SA(s) are a pending start. The stop icon (red octagon) appears if the SA(s) are a pending stop. Click the icon to navigate to the Start Stop - Pending SA tab where the associated pending start / pending stop service agreements will be displayed.

Premise Information Summary information about the pending start/stop premise is displayed.

CIS Division / SA Type This is the CIS division and service agreement type of the pending start / pending stop service agreements.

Total This is the total number of service agreements with the same **Date**, **Action**, **Premise**, and **CIS Division / SA Type**.

Following the Pending Start / Pending Stop grid are two sections; one is used to start service, the other is used to stop service.

The following points describe the fields in the start service section (refer to How To for a description of how to perform common start functions).

- Use **Start Date** to define the date on which you want service to commence.
- Use **Start Method** to define what you want to start. Valid values are:

Copyright © 2026, Oracle and/or its affiliates.

  - Start Premise SPs. Use this method if you want to start selected services at a premise. This method allows you to control the display of the service agreements to be started, providing a manageable view when there are many service agreements to be started at a premise.
  - Start a Premise. Use this method if you want to start services at a premise.
  - Start a SA. Use this method if you want to start a non-premise-oriented service (e.g., a deposit, a charitable contribution, a one-time charge).
- If you have selected the Start Premise SPs or Start a Premise option, use **Address** and **City** to select the premise. Refer to Control Central - Search for suggestions in respect of how to use these fields. The selected address and premise ID appear below the selection.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the premise search is restricted to the premises without a CIS Division and those whose CIS Division matches the account's CIS Division,

- If you have selected the Start a SA option, use **CIS Division** and **SA Type** to define the type of service agreement you wish to start.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the CIS Division is set to the account's CIS Division and cannot be changed.

> **Warning**
>
> Take special care when adding a new service agreement to specify the appropriate CIS Division / SA Type as it will affect how the customer is billed, what rate can be specified, how overdue debt is collected, and much more. After the service agreement is activated, you may not change its SA type.

- Use **Requested by** to define who requested the stop (e.g., Customer, Jane Smith).
- Click the **Start** button to start service. When you click this button, the Start Confirmation page appears.

The following points describe the fields in the stop service section (refer to How To for a description of how to perform common stop functions).

- Use **Stop Date** to define the date on which you want service to stop.
- Use **Stop Method** to define what you want to stop. Valid values are:
  - Stop Selected SAs. Use this method if you want to stop selected service agreements linked to the account.
  - Stop a Premise. Use this method if you want to stop ALL services at a premise.
  - Stop a SA. Use this method if you want to stop a specific SA.

Copyright © 2026, Oracle and/or its affiliates.

  - Stop all Active SAs. Use this method if you want to stop ALL service agreements linked to the account.
- If you have selected the Stop a Premise option, choose the **Premise ID**. You can optionally choose a specific premise if you selected the Stop Selected SAs option.
- If you have selected the Stop a SA option, choose the **SA ID**.
- Use **Requested by** to define who requested the stop (e.g., Customer, Jane Smith).
- Click the **Stop** button to stop service.

> **Note**
>
> Manual date entry is disabled on the start and stop dates and the date picker must be used when the Disable Manual Date Entry **Option Type** on the Customer Information Options Feature Configuration has been enabled.

Start/Stop - Start Confirmation

The Start SA Confirmation page appears after a user clicks the **Start** button on Start/Stop - Main. This page presents a summary of the service agreement(s) that will be created if the **Start** button on this page is clicked. Refer to How To for a description of how to perform common start functions.

Description of Page

The **Account ID** and the primary name of the account's main customer are displayed at the top of the page.

The **Start Date** is the date on which the service is scheduled to start.

Start Requested By contains the name of the person who requested the start.

The **SAs to Start** scroll contains an entry for every service agreement that will be created if you click the **Start** button. If you do not want a service agreement to be created, click the - button (or turn off **Start this SA**). If you'd like to start a non-premise oriented service agreement, click the + button. The following fields appear on this page:

Start this SA Turn this switch on if you want this type of service agreement to be created. Turn this switch off if you do not want this type of service agreement to be created.

If you used a **Start Method** of Start a SA on the Start/Stop page, this switch will be turned on by default.

If you used a **Start Method** of Start a Premise (or Start Premise SPs) and this is the first customer at a premise, this switch will be turned on for all service agreements that appear in the scroll. However, if there was a previous customer at the premise, this switch is only turned on for those types of service agreements used by the previous customer (we assume the new customer only wants the services used by the previous customer). Service agreements that are possible, but were not used by the previous customer, will appear in the scroll with this switch turned off.

CIS Division / SA Type These fields define the type of service agreement that will be created. If you change the CIS Division, you must change the SA Type to reflect one of the services that the division manages.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the CIS Division is set to the account's CIS Division and cannot be changed.

If the SA is linked to a service point, the SA types are restricted based on the type(s) of service points (SP) the system has associated with the service agreement (as indicated in the message box). If there are no SPs, only SA types that don't use SPs are displayed. If there is a single SP, only SA types valid for the related SP Type are displayed. If there are multiple SPs, only SA types valid for all related SP Types are displayed.

Start Option Some SA Types have start options. A start option defines the rate and contract terms defaulted on the service agreement. This field will be gray if the SA type does not use start options.

Total Amount to Bill Use this field if the service agreement is supposed to bill a given amount over its lifetime. This is typically used for deposit service agreements to define the amount of the deposit that should be billed. If the deposit is billed in installments, the installment amount is specified in the next field. This field is gray if the respective SA type doesn't use the Total Amount to Bill option.

 This button is only appears if the SA type has a "special role" of Cash Deposit and references a deposit class. When clicked, the system attempts to calculate a recommended deposit amount by calling the recommended deposit amount algorithm that is plugged in on the SA type's deposit class.

Customer Read Use this value to define if the customer reads their own meter. Valid values are Yes and No. This field does not appear for some SA types.

Allow Estimates Turn on this switch if it's OK for the system to use estimated usage from Oracle Utilities Meter Data Management for this service agreement's agreement consumption if a real meter read is missing at billing time. If the SA type does not allow estimates, this does not appear.

Recurring Charge Amount This field's label is dynamic based on the content of the Recurring Charge Amount Label on the respective SA Type. This field is gray if the respective SA type doesn't use the Recurring Charge option. This field is typically used only for the following types of service agreements:

• Service agreements on a levelized payment plan. This field contains the amount charged each month for the service regardless of the actual amount owed.

• Merchandise service agreements on an installment plan. This field contains the installment amount.

• Deposit service agreements on an installment plan. This field contains the installment amount.

• Charitable contribution service agreement. This field contains the contribution amount.

> **Note**
>
> **Default note.** The recurring charge amount defaults based on the **Start Option**.

Copyright © 2026, Oracle and/or its affiliates.

SIC Select the SIC code associated with the customer. This is typically used only for commercial and industrial customers.

Business Activity Enter a description of the business activity at the location where this service is being started.

SA Information This is a display-only area that provides information about the specific pending start SA. For example, information about the service points linked to the SA will be displayed for SP-oriented service agreements.

Messages This is a display-only area that that provides information about the account's start/stop situation in general. For example, if there are multiple accounts paying for service at the premise an appropriate message will be displayed.

When you click the **Start** button, the system performs the following for every service:

- A service agreement is created effective on the start date.
- If a start option is specified, the service agreement's rate and contract terms are populated using information on the option. If a start option is not specified, the service agreement's rate is populated with the SA type's default rate.
- Service points are linked to the service agreement (if the service agreement is service point-oriented). The SA/SP start date for each service point is the same as the SA's start date.

If you use a **Start Method** of Start Premise SPs on Start/Stop - Main, the **SAs to Start** are displayed in a grid; you can control the service agreements that appear using filter options. You may want to use the filter if you have a premise with a large number of service agreements to start.

Premise Info displays summary information about the premise at which services are to be started.

The section beneath **Premise Info** provides you with options that control which service agreements appear in the grid. The following points describe the various options:

- Use the **SP Type Filter** to restrict the service agreements that appear in the grid based on the SP type of the service points that will be linked to the service agreement. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on service point type.
  - Specific SP Type. Use this option to restrict service agreements to those that will be linked to service points of a specific **SP Type**.
- Use the **Geographic Type Filter** to restrict the service agreements that appear in the grid based on a geographic type and value of the service points that will be linked to the service agreement. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on a geographic type and value.
  - <Geographic Type>. Use this option to restrict service agreements to those that will be linked to service points associated with the selected geographic type and **Geographic Value**.
- Select whether to Include or Exclude service agreements that are linked to service points where **Service Exists Already**. These service agreements are those that would be not be started by default. Refer to the description of the **Start this SA** checkbox for more information on service agreements that would be not be started by default.

Don't forget to click the search button after changing the filters.

Copyright © 2026, Oracle and/or its affiliates.

The **Select All** / **Clear All** buttons are used to select service agreements to start. These buttons are enabled if at least one service agreement is displayed in the grid.

> **Note**
>
> **30 service agreements at a time.** Clicking **Select All** selects the first 30 service agreements in the grid. If more than 30 service agreements exist, you must select and start them in batches.

The grid that displays contains the **SAs to Start** that match your filter criteria. Click the drill down button to open the **Start SA Confirmation Details** page.

The fields in the grid on the Start SA Confirmation page and on the Start SA Confirmation Details page are as described above for the **SAs to Start** scroll. Differences are highlighted here:

SP Info This is summary information about the service point at which service is to be started.

SP Identifier The service point's SP type's identifying geographic type and value. Refer to Understanding Service Point Types. Up to three geo types and values are displayed in the SAs to Start grid.

Click the **Start** button on the Start SA Confirmation page to start the selected SAs. Click on the **Close** button to return to the main page - any changes made on this page since the last time you clicked **Start** are discarded.

Start/Stop - Stop Confirmation

If you choose the stop method of Stop Selected SAs, the Stop SA Confirmation page appears after a user clicks the **Stop** button on Start/Stop - Main. This page presents a summary of the service agreement(s) that will be stopped if the **Stop** button on this page is clicked. Refer to How To for a description of how to perform common start stop functions.

Description of Page

The **Account ID** and the primary name of the account's main customer are displayed at the top of the page.

The **Stop Date** is the date on which the service is scheduled to stop.

Requested By contains the name of the person who requested the stop.

The section beneath **Requested By** provides you with options that control which service agreements appear in the grid. The following points describe the various options:

- Use the **Premise Filter** to restrict the service agreements to those that are linked to a specific characteristic premise. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on premise.
  - No Premise. Use this option to restrict service agreements to those that do not have a characteristic premise.
  - Specific Premise. Use this option to restrict service agreements to those with a characteristic premise of **Premise ID**.
- Use the **SA Type Filter** to restrict service agreements to those of a given SA Type.

Copyright © 2026, Oracle and/or its affiliates.

  - All. Use this option if you do not wish to restrict service agreements based on SA Type.
  - Specific SA Type. Use this option to restrict service agreements to those of a given **CIS Division** and **SA Type**.
- Use the **Geographic Type Filter** to restrict the service agreements that appear in the grid based on the geographic type and value of the service points that are linked to the service agreement. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on a geographic type and value.
  - <Geographic Type>. Use this option to restrict service agreements to those that will be linked to service points / premises associated with the selected geographic type and **Geographic Value**.

Don't forget to click the search button after changing the filters.

The **Select All** / **Clear All** buttons are used to select service agreements to stop. These buttons are enabled if at least one service agreement is displayed in the grid.

> **Note**
>
> **30 service agreements at a time.** Clicking **Select All** selects the first 30 service agreements in the grid. If more than 30 service agreements exist, you must select and stop them in batches.

The grid that follows contains the service agreements to stop that match your filter criteria. The following information appears in the grid:

Select Box Use this checkbox to select service agreements that you want to stop.

Premise Information Summary information about the pending stop premise is displayed.

SA Information Summary information about the pending stop SA is displayed.

SP Identifier The service point's SP type's identifying geographic type and value. Refer to Setting Up Service Point Types. Up to three geo types and values are displayed.

Click the **Stop** button on the Stop SA Confirmation page to stop the selected SAs. Click on the **Close** button to return to the main page - any changes made on this page since the last time you clicked **Stop** are discarded.

Start/Stop - Address/Phone

Open **Menu**, and then **Customer Information**, and then **Start/Stop** and navigate to the **Address/Phone** page to change basic information about a customer while you are starting or stopping service.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **A subset of person and account information.** This page features a subset of person and account fields that typically change when a customer moves. You can always use the Person or Account transactions to change this information.
>
> **This information is held in limbo.** The information on this page will only be updated on the respective person / account when the first service agreement linked to the account is activated or stopped (i.e., this information is held in limbo until the pending start / pending stop is activated / stopped).
>
> **Main customer information.** The information on this page is associated with the account's main customer (as defined on Account - Person Information). If you need to change information about any other person linked to the account, you must transfer to Account - Person Information and make the changes there.

Use **Bill Route Type** to indicate how the bill is sent to the customer. If the **Bill Route Type** you select indicates that bills are routed via fax, make sure you specify a fax number in the **Telephone Numbers** below (note, the system knows which of a person's phone numbers is a fax number by the phone type). If the **Bill Route Type** you select indicates that bills are routed via Email, make sure to specify the person's **Email Address** below. If the **Bill Route Type** you selected indicates that bills are routed via the postal service, select the appropriate **Address Source**. You have the following options:

- Choose Mailing Premise on Account if bills should be sent to the account's mailing premise. This address is displayed below.
- Choose Person if bills should be sent to the person's mailing address . This address is displayed below.
- Choose Account Override if bills should be sent to the account person's override address. This address is displayed below. Typically, you would only choose this option if the person has multiple accounts and each account's bills should be sent to a different address.

If the **Address Source** is Person or Account Override , the respective address fields are displayed and may be changed. If the **Address Source** is Mailing Premise on Account, the account's mailing premise is displayed and cannot be modified. Refer to Premise - Main Information for information about these fields.

> **Note**
>
> **Default note.** If you switch the **Bill Route Type** from a non-postal to postal routed, the **Address Source** defaults to the bill address source defined on the account (if one exists) or Mailing Premise on Account if bill address source is not specified for the main customer.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> By default, the **Address Source** field is protected on Start/Stop– Address/Phone page if the **Bill Route Type** is set to a value other than Postal. To enable the **Address Source** field for all **Bill Route Types** in most circumstances, set the Always Enable Address Source option type on the General System Configuration feature configuration to Y.
>
> Regardless of the **Always Enable Address Source** option value, **Address Source** and **Address Information** are displayed but both are disabled when **Bill Routing Method** is ***Not Postal*** and **Quote Routing Method** (on Account – Persons page) is ***Postal***. A message "*The Main Person receives Quotes at this address. To edit it, go to the 'Persons' tab on the Account.*" is displayed next to the **Address Source.**

> **NOT_SUPPORTED**
>
> Refer to Setting Up Bill Route Types for more information about bill route types.

The person contacts, from the Person - Main page are displayed so that you can review and update them as necessary.

> **Note**
>
> The Email Address field appears when the Legacy Person Phone and Email Option Type on the Customer Information Options Feature Configuration has been enabled.

Specify the customer's **Email Address**. This can be used to route bills to the customer via email (if your bill routing software supports this function).

Telephone Numbers are used by Control Central to look for accounts and persons. If the person has more than 20 phone numbers, the first 20 will be shown as display only with a message indicating that more phone numbers exist until you save your start or stop. Once the start or stop has been saved, you are able to modify the phone number collection.

> **Note**
>
> The Person Phone grid appears when the Legacy Person Phone and Email **Option Type** on the Customer Information Options Feature Configuration has been enabled.

The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business, ...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

Start/Stop - Pending SA

If you need to change additional information about the pending start or stop SAs, navigate to the **Pending SA** tab. Alternatively, you can click the Drill Down button on the SA grid of the **Main** tab.

Description of Page

If the total number of pending start/stops for the account does not exceed the start stop details threshold defined in Installation Options, an entry appears in the scroll for each entry in the SA grid on the main page. Use the **Pending Start/Stop SA** navigation arrow buttons to select the pending SA you want to review or modify. This page is split between the primary SA on the top, and the reciprocal SA on the bottom. The following fields are available.

Primary SA:

SA Info Summary information about the pending start/stop SA is displayed.

Cancel Turn on this switch if you wish to cancel the pending start/stop.

Date This is the start date for pending start service agreements and the stop date for pending stop service agreements.

Reason This column indicates if the pending start/stop is System Generated or Customer Requested. System-generated starts are created when a tenant stops service at a premise covered by a landlord agreement. System-generated stops are created when a customer requests a start at a premise where the existing customer hadn't already requested a stop.

CIS Division / SA Type These fields define the type of service agreement that will be created. These fields will be gray if the service agreement is pending stop.

If you change the CIS Division, you must change the SA Type to reflect one of the services that the division manages.

If the SA is linked to a service point, the SA types are restricted based on the type(s) of service points (SP) the system has associated with the service agreement (as indicated in the message box). If there are no SPs, only SA types that don't use SPs are displayed. If there is a single SP, only SA types valid for the related SP Type are displayed. If there are multiple SPs, only SA types valid for all related SP Types are displayed.

Start Option Some SA Types have start options. A start option defines the rate and contract terms defaulted on the service agreement. This field will be gray if the SA type does not use start options or if the service agreement is pending stop.

Premise Summary information about the pending start / stop premise is displayed.

Reciprocal SA:

Copyright © 2026, Oracle and/or its affiliates.

SA Info Summary information about the pending start/stop SA is displayed.

Cancel Turn on this switch if you wish to cancel the pending start/stop.

Date This is the start date for pending start service agreements and the stop date for pending stop service agreements.

Reason This column indicates if the pending start/stop is System Generated or Customer Requested. System-generated starts are created when a tenant stops service at a premise covered by a landlord agreement. System-generated stops are created when a customer requests a start at a premise where the existing customer hadn't already requested a stop.

CIS Division / SA Type These fields define the type of service agreement that will be created or stopped. This field will be gray if the service agreement is pending stop.

If you change the CIS Division, you must change the SA Type to reflect one of the services that the division manages.

If the SA is linked to a service point, the SA types are restricted based on the type(s) of service points (SP) the system has associated with the service agreement (as indicated in the message box). If there are no SPs, only SA types that don't use SP's are displayed. If there is a single SP, only SA types valid for the related SP Type are displayed. If there are multiple SPs, only SA types valid for all related SP Types are displayed.

Start Option Some SA Types have start options. A start option defines the rate and contract terms defaulted on the service agreement. This field will be gray if the SA type does not use start options or if the service agreement is pending stop.

Premise Summary information about the pending start / stop premise is displayed.

Account The name of the main customer associated with the reciprocal's service agreement's account is displayed.

If the total number of pending start/stops for the account exceeds the start stop details threshold defined in Installation Options, the pending start/stop SAs and their reciprocal SAs are displayed in a grid; you can control the service agreements that appear using filter options.

The section beneath **Account** provides you with options that control which service agreements appear in the grid. The following points describe the various options:

- Use the **SA Action Filter** to restrict the service agreements that appear in the grid to pending starts or pending stops. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on SA Action.
  - Start. Use this option to restrict service agreements to pending starts.
  - Stop. Use this option to restrict service agreements to pending stops.
- Use the date filter to restrict service agreements to those that will start or stop between the **From Date** and the **To Date**.
- Use the **Premise Filter** to restrict the service agreements to those that are linked to a specific characteristic premise. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on premise.
  - No Premise. Use this option to restrict service agreements to those that do not have a characteristic premise.
  - Specific Premise. Use this option to restrict service agreements to those with a characteristic premise of **Premise ID**.
- Use the **SA Type Filter** to restrict service agreements to those of a given SA Type.

Copyright © 2026, Oracle and/or its affiliates.

  - All. Use this option if you do not wish to restrict service agreements based on SA Type.
  - Specific SA Type. Use this option to restrict service agreements to those of a given **CIS Division** and **SA Type**.
- Use the **Geographic Type Filter** to restrict the service agreements that appear in the grid based on the geographic type and value of the service points that are linked to the service agreement. The following options are available:
  - All. Use this option if you do not wish to restrict service agreements based on a geographic type and value.
  - <Geographic Type>. Use this option to restrict service agreements to those that will be linked to service points / premises associated with the selected geographic type and **Geographic Value**.

Don't forget to click the search button after changing the filters.

The **Select All** / **Clear All** buttons are used to select service agreements if you plan on issuing any of the mass update actions at the bottom of the page. These buttons are enabled if at least one service agreement is displayed in the grid.

> **Note**
>
> **30 service agreements at a time.** Clicking **Select All** selects the first 30 service agreements in the grid. If more than 30 service agreements exist, you must select and start them in batches.

The grid that displays contains the service agreements that match your search criteria. Click on the drill down button to open the **Pending SA Details** page.

Please refer to the **Pending Start/Stop SA** scroll for a description of the fields in the pending start/stop SA grid on the Pending SA tab and on the Pending SA Details page. Additional fields are described here:

Pending SAs Grid:

- **Select box**. Use this checkbox to select service agreements for mass update.
- The **Action** icons highlight if the associated SA is a pending start or a pending stop and if there are reciprocal SAs. Refer to the action icon on Start Stop - Main.

Mass Update Buttons:

Use the **Cancel Selected** button to cancel the selected pending start/stops together with any associated reciprocal pending start/stops. This button is enabled if you have selected one or more pending SAs.

Use the **Change Start/Stop Date** button to change the start/stop date for the selected pending start/stop service agreements. If a selected start/stop has a reciprocal SA, its start/stop date is also changed. This button is enabled if you have selected one or more pending SAs.

Start/Stop - Pending Field Activity

When you start/stop service point-oriented service, field activity requests are typically created. You can view every non-cancelled, non-complete field activity linked to the account's service agreements on the **Pending Field Activity** page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The field activity requests are NOT limited to those associated with starting / stopping service. So, if a customer has a pending "cut for non payment" field activity request, it would also appear on this page.

Description of Page

An entry appears in the scroll for every Pending field activity request linked to the account's service agreements. The information in the **Field Activity** column contains a summary of the field activity request and its service point. The information in the **How Used** column describes the purpose of the field activity request (such as, Start Activity, Stop Activity, and Severance Activity).

If the total number of pending start/stops for the account exceeds the start stop details threshold defined in Installation Options, you can control the field activity requests that appear using filter options.

The section beneath **Account** provides you with options that control which field activity requests appear in the grid. The following points describe the various options:

- Use the **Schedule Date** filter to restrict the field activity requests that appear in the grid to those that are scheduled between. The following options are available:
  - All. Use this option if you do not wish to restrict field activities to a specific schedule date.
  - Specific Date. Use this option to restrict the field activities to those with a schedule date between the **From Date** and the **To Date**.
- Use the **Premise Filter** to restrict the field activity requests to those that are linked to service points at a specific premise. The following options are available:
  - All. Use this option if you do not wish to restrict field activity requests based on premise.
  - Specific Premise. Use this option to restrict field activity requests to those linked to a service point at the given **Premise ID**.
- Use the **SP Type Filter** to restrict field activity requests to those that are linked to service points of a given SP Type.
  - All. Use this option if you do not wish to restrict field activity requests based on SP Type.
  - Specific SA Type. Use this option to restrict field activity requests to those of a given **SP Type**.
- Use the **Geographic Type Filter** to restrict the field activity requests that appear in the grid based on the geographic type and value of the service points that are linked to the service agreement. The following options are available:
  - All. Use this option if you do not wish to restrict field activity requests based on a geographic type and value.
  - <Geographic Type>. Use this option to restrict field activity requests to those that are linked to service points / premises associated with the selected geographic type and **Geographic Value**.

Don't forget to click the search button after changing the filters.

Copyright © 2026, Oracle and/or its affiliates.

How To

The topics in this section describe how to perform common start/stop functions.

- How To Start Service At A Premise
- How To Stop Service At A Premise
- How To Start A Deposit
- How To Start A Charitable Contribution
- How To Start A One-Time Charge Service Agreement
- How To Start Service When Multiple Service Points Measure Consumption

How To Start Service At A Premise

The following instructions specifically reference the *Start a Premise* Start Method. You can also use the *Start a Premise's SPs* Start Method and its corresponding user interface.

You navigate through the pages as follows:

- Use Control Central - Search to look for the customer. If the customer exists, click the Start Service button to transfer to Start/Stop. If the customer doesn't exist, add a new person/account by following the instructions under How To Add A New Customer From Control Central.
- Specify the **Start Date**, **Start Method** (Start a Premise**)**, and select a premise (by entering the premise **Address** and clicking the search button).
- Click the **Start** button to review the various service agreements that will be created.

> **Note**
>
> **Service agreements are created.** Each row in the scroll is associated with a service agreement. When you close the page, the system creates new service agreements in the state of pending start.

- If you're starting service at a premise, the system automatically populates the Start Confirmation page with information about every service agreement to be created. To do this, we make a basic assumption - the customer wants every service at the premise. If this assumption is incorrect, simply remove the service agreement from the scroll (by turning off the **Start this SA** switch) before closing the page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Automatically stop the existing customer.** If there's a customer currently paying for service at the premise, the system automatically sets up the stop information for the existing customer. The stop date of the existing customer is the start date of the new customer. If you don't want to stop the existing customer, simply indicate such in the scroll.

> **Warning**
>
> Take special care when adding a new service agreement to specify the appropriate CIS Division and SA Type as it will affect how the customer is billed, what rate can be specified, how overdue debt is collected, and much more. After the service agreement is activated, you may not change its CIS Division / SA type.

> **NOT_SUPPORTED**
>
> Refer to Starting Service and Field Activities for the details about how the system automatically creates the field activity requests to support start service requests.

How To Stop Service At A Premise

The following instructions specifically reference the *Stop a Premise* Stop Method. You can also use the *Start Selected SAs* Stop Method and its corresponding user interface.

You navigate through the pages as follows:

- Use Control Central - Search to find the customer. Click the Stop Service button to transfer to Start/Stop.
- Specify the **Stop Date**, **Stop Method (**Stop a Premise**)**, and select a **Premise** (the premise will have been pre-selected for you if you drilled down from Control Central).
- Click the **Stop** button to stop service at the premise.
- When you click the stop button, the system changes the status of each service agreement to pending stop and sets the stop date accordingly. If we stopped a service agreement that we shouldn't have, simply cancel the stop by checking the **Cancel** column in the grid (and save the page)

> **Note**
>
> **Landlord reversion.** If you stop service at a premise that references a landlord agreement, the system will automatically create pending start service agreements for the landlord for those service types covered by the landlord agreement.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Stopping Service and Field Activities for the details about how the system creates the field activity requests to support stop service requests.

How To Start A Deposit

Charging a customer for a deposit involves creating a service agreement. Deposit service agreements are just like other service agreements in many ways:

- The system bills the customer for the deposit using information on the service agreement.
- The customer pays for the billed amount.
- If the customer doesn't pay, credit & collections starts a collection process to encourage the customer to pay.

Deposit service agreements differ from regular service agreements in the following ways:

- Deposit service agreements typically don't cause a receivable account to be affected when the deposit is billed. It is only when the deposit is paid that a financial transaction occurs (debit cash, credit deposit payable).
- Deposit service agreements have a credit balance when the billed amount is paid (the deposit paid by the customer is a liability of your company - it owes the customer this money).
- Deposit service agreements may be billed in installments or in a lump sum.

Creating a deposit service agreement involves the following steps:

- Select the account on Control Central and open Start/Stop.
- Enter a **Start Date** and **Start Method** of Start a Premise.
- Indicate the premise that the deposit service agreement will cover using the Address and click the **Start** button.
- After you click the **Start** button, the Start Confirmation page will appear:
  - Choose an appropriate **CIS Division / SA Type** for the deposit service agreement.
  - If **Start Options** have been created for the CIS Division / SA Type, you need only select the appropriate option and close the page. The system will use the information on the start option to populate the service agreement's billing parameters (e.g., the total deposit due, the installment amount, etc.).
  - If **Start Options** have not been created for the CIS Division / SA Type, you must define the total amount of the deposit and the installment amount, if any. You define the total amount of the deposit in **Total Amount To Bill**. You can have the system recommend a deposit amount by pushing the  that appears adjacent to this field.
  - If you allow the customer to pay the deposit in installments, define the Installment Amount (the prompt for this field is defined on the SA type and therefore will differ depending on how you've set up the system). After defining this information, close the page. The system will use the information you entered to create a deposit service agreement.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The above information related to starting a deposit assumes that the deposit SA Type is set up as a premise-based service. If your company does not associate a deposit with a specific premise, you will need to set up your SA Type for the deposit accordingly. Then, you will need to use a Start Method of Start a SA.

How To Start A Charitable Contribution

If a customer elects to contribute money to a charitable organization, you must create a service agreement. Charitable contribution service agreements are just like other service agreements in many ways:

- The system bills the customer for the elected contribution amount using information on the service agreement.
- The customer pays the billed amount.
- If the customer doesn't pay, credit & collections may start a collection process to encourage the customer to pay (depending upon how the charitable contributions debt class has been set up on your system).

Charitable contribution service agreements differ from regular service agreements in the following ways:

- Charitable contribution service agreements typically don't cause a receivable account to be affected when the charitable contribution is billed. It is only when the charitable contribution is paid that a financial transaction occurs (debit cash, credit charitable contribution payable).
- Charitable contribution service agreements have no balance when the billed amount is paid because the money is being forwarded to a charity.

Creating a charitable contribution service agreement involves the following steps:

- Select the account on Control Central and open Start/Stop.
- Enter a **Start Date** and **Start Method** of Start a SA.
- Indicate the **CIS Division** / **SA Type** of the charitable contribution agreement and click the **Start** button.
- After you click the **Start** button, the Start Confirmation page will appear:
  - If **Start Options** have been created for the CIS Division / SA Type, you need only select the appropriate option and close the page. The system will use the information on the start option to populate the service agreement's billing parameters (i.e., the charitable contribution amount to be periodically billed).
  - If **Start Options** have not been created for the CIS Division / SA Type, you must define the charitable contribution amount (the prompt for this field is defined on the SA type and therefore will differ depending on how you've set up the system). After defining this information, close the page. The system will use the information you entered to create a charitable contribution service agreement.

Copyright © 2026, Oracle and/or its affiliates.

How To Start A One-Time Charge Service Agreement

You create a one-time charge (i.e., billable charge) whenever a customer should be charged for a service that occurs outside the normal course of business. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line.

A billable charge must reference a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each unbilled billable charges.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.

Therefore, in order to levy a one-time charge, the customer must have a billable charge service agreement. If one already exists (because of a previous billable charge), you don't have to create a new service agreement. Rather, transfer to the Billable Charge page and add a billable charge for the service agreement. If the customer doesn't already have billable charge service agreement, create one by performing the following steps:

- Select the account on Control Central and open Start/Stop.
- Enter a **Start Date** and **Start Method** of Start a SA.
- Indicate the **CIS Division** / **SA Type** of the billable charge service agreement and click the **Start** button.
- After you click the **Start** button, the Start Confirmation page will appear. Simply close the confirmation page to save the agreement.
- After the billable charge service agreement has been created, add a billable charge to it (the easiest way is to use the SA context menu). Refer to Maintaining Billable Charges for more information.

How To Start Service When Multiple Service Points Measure Consumption

When you start service at a premise, the system automatically links service points to service agreements in the same manner as existed for the previous customer. So, if the previous customer amalgamated consumption of three service points and billed this under a single service agreement, the system will create the same configuration for the new customer.

This works fine when the unusual service point configurations have already been set up for the previous customer. However, it doesn't work when the premise is new. So, if you have a new premise that has multiple service points whose consumption needs to be amalgamated under a single service agreement, do the following:

- Start service for only one of the service points to be linked to the service agreement. When you start service, the system creates a service agreement, links the service point to it, and links a field activity to the service point.
- For every service point that wasn't linked to the service agreement:

Copyright © 2026, Oracle and/or its affiliates.

– Link the service point to the pending service agreement This task is done on Service Agreement - SA / SP.

  * • After linking all the service points to the pending service agreement, manually activate the service agreement. This task is done on Service Agreement – Main. While the service agreement might not be truly activated at this time, this step ensures necessary field activity requests are created behind the scenes.

field activity requests

> **Note**
>
> **That's it.** The activation process will automatically activate the service agreement when the fieldwork is complete on all of its service points.

Customer Contacts

Customer contacts are used to record when and why a customer contacted your company. They can also be associated with person, accounts, and premises. This information can be used for both audit and statistical purposes. The topics in this section provide details about how customer contacts are created and used in the system.

- Person-Based Versus Premise-Based Customer Contacts
- How Customer Contacts Are Created
- Customer Contacts Are Used To Trigger Letters
- A Customer Contact May Trigger Reminders
- Customer Contacts Are Used To Record Notifications
- Customer Contacts Can Be Used As Case Files
- Maintaining Customer Contacts
- Creating Customer Contacts Using Call Summary and Call Tagging

Person-Based Versus Premise-Based Customer Contacts

Customer Contacts can be linked to the person, account, and premise.

- If a person is associated with the customer contact, it is considered to be a person-based contact regardless of whether an account and/or premise are also associated with the contact.
- If only a premise is associated with the customer contact, the system will designate the contact as premise-based.
- Contacts that have an account associated with them must also have a person associated and the contact will become person-based. There are no account-based customer contacts.

Customer Contacts that are associated with a person, account, and/or premise can be accessed through each entity's context menu.

Copyright © 2026, Oracle and/or its affiliates.

How Customer Contacts Are Created

You can add a Customer Contact at any time using either of the following methods:

- Use the customer contact maintenance page.
- Use the Customer Contact Zone on the sidebar.

When customer contacts are created through the Customer Contact sidebar zone, the system populates person, account, and/or premise based on context.

In addition to manually adding customer contacts, your implementation team can set up plug-in algorithms that create customer contacts when certain events transpire. For example, you can plug-in an algorithm on the customer class object that creates a customer contact whenever an order is completed. Because the number and type of plug-ins can be customized for your implementation, we cannot provide a concise list of all such algorithms.

In addition to the above, a variety of events can trigger the creation of a customer contact. Refer to Customer Contacts Are Used To Trigger Letters for the details.

Customer Contacts Are Used To Trigger Letters

In order to send a letter to a customer, a customer contact must be created for the customer. These types of customer contacts reference a customer contact type that, in turn, references a letter template. The letter template controls the type of information that is merged into the "form letter" and how the letter is physically produced. Refer to Printing Letters for more information about letter templates and how letters are physically produced.

Because these letters are being sent to a customer, the customer contact that triggers a letter must be person-based. See Person-Based Versus Premise-Based Customer Contacts for more information.

Customer contacts that trigger letters can be produced by any of the method described under How Customer Contacts Are Created. While a user can create this type of customer contact (e.g., if a customer requests a form to sign-up for automatic payment), the primary sources of customer contacts that trigger letters are via system events and algorithms. Besides algorithms, the following system events may generate customer contacts automatically:

- **Collection events.** When a customer falls in arrears, the system creates a collection process. A collection process is a series of events meant to encourage the customer to pay their outstanding debt. A classic collection event is one that creates a customer contact that, in turn, causes a letter to be sent to the customer requesting payment. These customer contacts are created with person and account populated. Refer to The Big Picture Of Collection Events for more information.
- **Severance events.** The last event in a collection process typically creates a severance process for each service agreement that's in arrears. A severance process is a series of events that leads to service being severed. A classic severance event is one that creates a customer contact that, in turn, causes a letter to be sent to the customer warning of impending cutoff (or telling them they've been cutoff and the terms necessary to reinstate service). These customer contacts are created with person and account populated. Premise is also populated if the service agreement being severed is associated with a premise. Refer to The Big Picture Of Severance Events for more information.

Copyright © 2026, Oracle and/or its affiliates.

- **Write-off events.** After a customer has been final billed, a write-off process will be created. A write-off process is a series of events that leads to the debt being written off. A classic write-off event is one that creates a customer contact that, in turn, causes a letter to be sent to the customer informing them of that their debt has been referred to a collection agency. These customer contacts are created with person and account populated. Refer to The Big Picture Of Write Off Events for more information.
- **Overdue events.** Overdue processes can be used to collection on past due debt. An overdue process is a series of events that are used to collect on debt. Some overdue events may create a customer contact that, in turn, causes a letter to be sent to the customer. These customer contacts are created with person and account populated. Refer to The Big Picture Of Overdue Events for more information.

Your implementation team can introduce plug-in algorithms to create a customer contact when certain events take place. If the created customer contact references a letter template, a letter will be triggered. Because the number and type of plug-ins can be customized for your implementation, we cannot provide a concise list of all such algorithms.

Algorithms that are delivered with the product populate person, account, and/or premise based on the information available when the customer contact is created. For example, the field activity remark algorithm populates the premise and also the person and account if the field activity's service point is linked to a service agreement. From the service agreement the account and account's main person are obtained and used on the customer contact.

A Customer Contact May Trigger Reminders

If you want the system to remind you (via a To Do entry) about a customer-related issue, you can set up a "reminder" on a customer contact. To do this, follow these steps:

- Determine if you need a new customer contact or if you can reuse one that already exists. If you need a new customer contact, use any of the "online methods" described under How Customer Contacts Are Created.
- Refer to the embedded help on the Customer Contact Reminders zone on the Customer Contact portal for more information about using reminders.

Customer Contacts Are Used To Record Notifications

Notification types may reference a customer contact type that is used to create a record of notifications of that type. This is supported for both message-based and template-based notifications. Refer to Template-Based vs. Message-Based Notification Types for more information.

Customer contacts create from template-based notifications contain more notification details and support an integration that updates the customer contact with information from an integration system such as *Oracle Integration Cloud* and also from the system responsible for delivering the notification. Customer contacts may also be added and updated by an integration system to record notifications initiated from an external system such as outage notifications that originate from a network management system.

Customer Contacts Can Be Used As Case Files

Copyright © 2026, Oracle and/or its affiliates.

If a customer calls with a complaint (e.g., they suspect their bill is too high), you can start a "case file" for the complaint by creating a customer contact. This customer contact should be marked as "open".

> **Note**
>
> **Alerts.** Customer contacts that are marked as being open cause an alert to appear when the customer is displayed on control central.

Over time, as you work on the case, you can add entries to the customer contact's log to describe your progress.

You can use the customer contact's reminder functionality to remind you to check out the case on a future date (or dates).

When the case is resolved, you simply turn off the customer contact's open indicator and add an appropriate entry to the log describing the resolution.

Maintaining Customer Contacts

Customer contacts are used to record when and why a customer contacted your company. This information is used for both audit and statistical purposes. The topics in this section describe how to maintain customer contacts. Refer to The Big Picture Of Customer Contacts for background information about customer contacts.

To view and maintain customer service requests, open **Menu**, and then **Customer Information**, and then **Customer Contact**, and then **Search**. You are brought to a query portal with options for searching for customer contacts.

Once a customer contact has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are displayed:

- **Customer Contact**
- **Customer Contact Reminders**

Refer to the zones' embedded help for more information.

Creating Customer Contacts Using Call Summary and Call Tagging

Customer Service Representatives (CSRs) can use Call Summary and Call Tagging to create Customer Contacts that include an automatically generated summary of the call as well as tags (based on Characteristic Types) to categorize the call. Call Summary and Call Tagging uses Generative Artificial Intelligence and integration with a third-party Computer Telephony Integration (CTI) vendor to automatically summarize and categorize call transcripts, providing customer service agents or customer service representatives (CSR) a summary of the call. Agents can review and edit the summary and tags before saving them directly to the customer contact.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Call Summary and Call Tagging is only available with Oracle Utilities Customer Cloud Service and Oracle Utilities Customer Care and Billing Cloud Service.

> **Note**
>
> Call Summary and Call Tagging relies on generative AI models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Call Summary and Call Tagging.

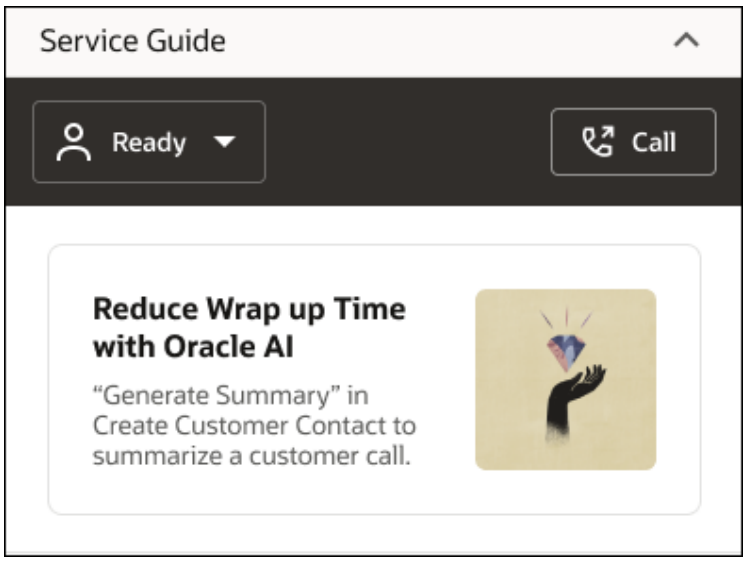
> **Note**
>
> Call summaries are prepared by generative AI. Carefully review and verify them against trusted sources, as they may include out-of-date, inaccurate, or incomplete information.

Customer Service Representatives access Call Summary and Call Tagging via the **Service Guide** zone in the **Assistant** area, located above the **Sidebar**. After accepting a call the agent can enter call notes, and then create a customer contact record to store information about the call when the call is completed.

To make themselves available to receive customer calls, the CSR should select "Ready" from the **Agent Status** dropdown list.

To make an outbound call, click the "Call" icon (

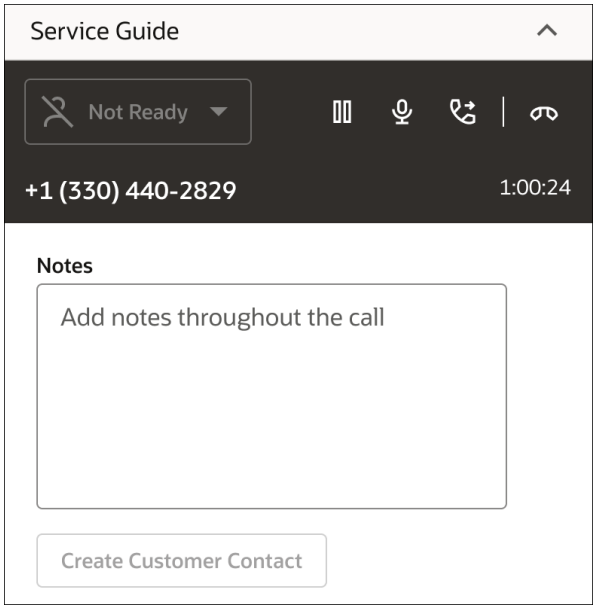
). You can either dial a complete phone number using the dialpad or select from pre-configured speed dial numbers.



When a customer service representative receives a call from a customer, the call appears in the **Service Guide** zone.

When a CSR answers a call displayed in the **Service Guide** zone, the **Agent Status** is set to "Not Ready" and the phone number and call duration appear in the zone. During the call the CSR can enter a brief description of the call in the **Notes** field in the zone.

Copyright © 2026, Oracle and/or its affiliates.

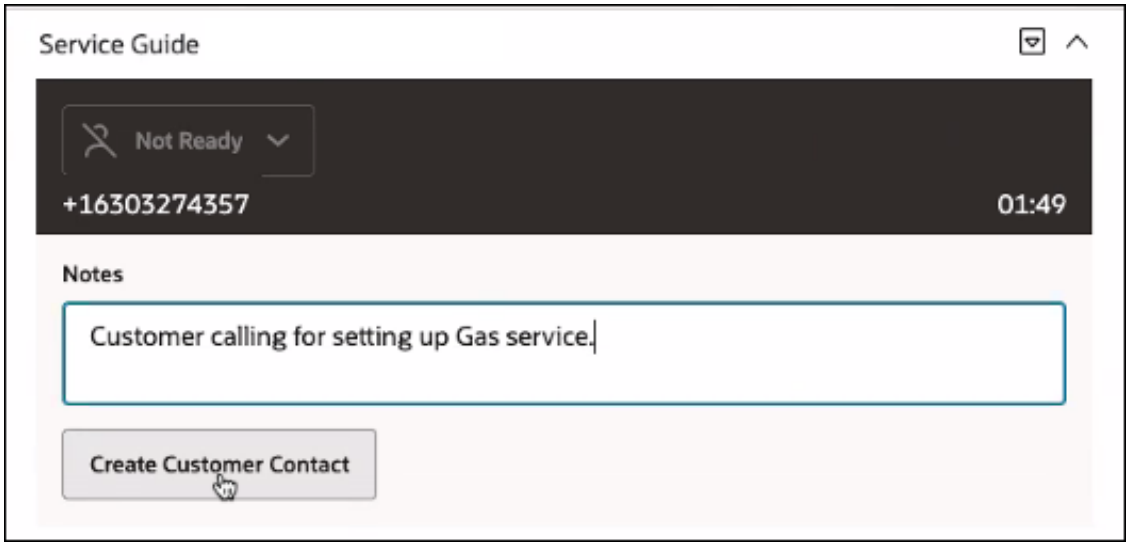


Other call options include:

- **Hold** ( ): Puts the call on hold. Click the icon again to return to the call
- **Mute** ( ): Mutes the CSR. Click the icon again to unmute.
- **Transfer Call** ( ): Click the icon to transfer a caller to another person or company during an active call. A "cold" transfer involves dialing the destination number using the dialpad and releasing the caller directly into the new line. Select Consult for a "warm" transfer that involves first connecting with the destination, introducing the customer, and then leaving the call once the handoff is complete.
- **End Call** ( ): Ends the call.

When the call is ended, the CSR should click the **End Call** icon (the call is also ended when the customer hangs up). The phone number and call duration remain displayed in the **Service Guide** zone, and the CSR will create a customer contact where they can enter specifics about the customer contact.

Copyright © 2026, Oracle and/or its affiliates.



To create a customer contact:

- Click **Create Customer Contact** in the **Service Guide** zone. The **Select Type** page opens.
- Select the **Contact Class** from the dropdown list.
- Select the **Customer Contact Type** from the dropdown list.
- Click **OK**. The **Customer Contact** page opens.
- Other details, including the **Person**, **Account**, and **Premise** associated with the customer are populated automatically if the customer is in context (based on the Customer Contact Type). Notes previously entered in the **Service Guide** zone are populated in the **Comments** field.
- Click **Generate Summary** to generate a summary and list of tags for the call. A summary of the call appears in the **Summary** field, and tags appear in the **Tags** field (tags are based on Characteristic Types defined on the Customer Contact Type). Both can be reviewed and edited as appropriate.
  - Edit the text in the **Summary** field as appropriate. To revert to the original summary text, click **Generate Summary** again. Note that you can only revert to the original summary and tags list before saving the customer contact.
  - Use the **Tags** dropdown list to edit the tags associated with the customer contact.
- Add additional **Characteristics** to the call as appropriate, selecting a **Characteristic Type** and enter a **Value** for each.
- Click **Save** to save the contact, including the **Summary** and **Tags**.

See Maintaining Customer Contacts for more information about maintaining customer contacts.

> **Note**
>
> By default the **Agent Status** is set to "Not Ready" when a call is ended. To configure your system to automatically set the **Agent Status** to "Ready" when a customer contact is created, contact your Oracle Customer Success Manager.

Copyright © 2026, Oracle and/or its affiliates.

Enabling Call Summary and Call Tagging

Enabling Call Summary and Call Tagging involves the following:

1. Create an integration with a CTI application
2. Configure Characteristic Types and Customer Contact Types with Characteristic Tags
3. Enable the Service Guide Zone

See Call Summary and Call Tagging Integration in the *Administrative User Guide* for more information about these steps.

> **Note**
>
> Call Summary and Call Tagging relies on generative AI models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Call Summary and Call Tagging.

Printing Letters

As described under Customer Contacts Are Used To Trigger Letters, you must create a customer contact whenever you want to send a letter to a customer. The contents of this section describe the technical implementation of letter production.

- The Letter Extract Processes
- BI Publisher Versus Document Composition Applications
- Where Does The Address Come From?
- Technical Implementation Of Online Letter Display
- Reproducing The Letter Print Flat File
- How To Reprint A Specific Letter
- Letter Data Area Organization
- Letter Extract File Structure
- Extending Letter Record Types

The Letter Extract Processes

Customer contacts that trigger letters reference a customer contact type that, in turn, references a letter template. Two background processes are provided with system that control the information extracted into a file.

C1-LTPEX – Letter Extract

This process produces a letter extract file based on letter extract records that are configured with each letter template. Each letter template can be configured with a different set of record type. Some record types have configuration options such as which characteristic types to extract or how to identify the object to extract. This is a highly configurable extract that should reduce the need for a custom solution.

Copyright © 2026, Oracle and/or its affiliates.

The record layouts can be viewed in the application on the Letter Template Extract Record List zone on the letter template portal.

> **Note**
>
> This batch does not utilize the extract algorithm plug-in spot on letter template. The sample Online Letter Image algorithm provided with the system calls the letter template extract algorithm plug-in spot and is not compatible with this solution.

LTRPRT – Letter Extract

This process uses the extract algorithm on the letter template and is responsible for extracting the information merged onto your letters. These algorithms create the "flat file records" that are passed to your letter print software.

Additionally, the extract algorithm can be called if your letter print software has the ability to construct a real-time image of a letter (in a PDF). You can plug-in an Online Letter Image algorithm on the Installation Record. This type of algorithm calls the letter template's extract algorithm to extract the information that is merged onto the letter. Refer to Technical Implementation of Online Letter Production for the details.

> **Note**
>
> **Alternate print / routing methods.** If you require different print formats, you must create new letter extract algorithms and plug them in on your letter template.

> **Note**
>
> This process is not intended for new implementations. For information related to the record layouts produced by the sample algorithms provide for use with this process, please refer to the documentation supplied with CC&B 2.7.0.1 and earlier.

BI Publisher Versus Document Composition Applications

Using BI Publisher as your external reporting tool to display an online letter image differs from using document composition applications as follows:

- An extract file is not required when using BI Publisher. The sample reports provided to use with these external report tools do not have to extract information from another object to merge into a letter.
- Generating letters in batch is not supported yet.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Letter Print - BI Publisher Sample Welcome Letters - CI_LTRGN_ENG REPORT for more information.

Where Does The Address Come From?

The following points describes the source of a letter's address:

- For letters generated as a result of collection events, severance events, write-off events and overdue events, the address is extracted as follows:
  - The respective Account - Person record is retrieved for the letter recipient (note, the account ID is retrieved from the related collection / severance / write-off / overdue process).
  - If the person has a seasonal address effective on the business date, the letter's address is the respective seasonal address.
  - If the address source is Mailing Premise on Account, the letter's address is the account's mailing premise.
  - If the address source is Account Override, the letter's address is the override address on the respective Account - Person record.
  - If the address source is Person, the letter's address is the person's mailing address .
  - If the address source is not defined, the person's mailing address is used.
  - If an address was not populated as a result of the above, the letter is not produced and a warning is issued by the background process responsible for creating the letter extract file.
- For all other types of letters, the address is extracted as follows:
  - If the letter's person has a seasonal address effective on the business date, the letter's address is the respective seasonal address.
  - If the letter's person has a mailing address , it is used.
  - Otherwise, the system checks if the letter's person has at least one account where the person is the main customer and the account has at least one active service agreement. If so, the logic described above is executed for the first such account found.
  - If an address was not populated as a result of the above, the letter is not produced and a warning is issued by the background process responsible for creating the letter extract file.

Technical Implementation Of Online Letter Display

Users can view an image of any letter that is sent to a customer if you plug plug-in an Online Letter Display algorithm on the Installation Record

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Some display algorithms rely on the extract algorithm on each letter template. Algorithms of this type format the records that contain the customer information that is passed to your printing software.

The system can make a call to an external system to retrieve a letter image. Many letter print vendors support this functionality.

The system provides an Installation – Online Letter Display algorithm type (C1GENLTTRURL) that can generate a dynamic URL.

- The algorithm of this algorithm type supports dynamic variables such as the customer contact ID and static variable such a vendor or implementation supplied security key.
- The algorithm constructs a URL including generating HASH or security keys.
- The system displays the retrieved image in a separate browser session.
- Refer to the algorithm type description for information about this algorithm type.

The system can interact with either document composition applications, or a reporting engine (such as BI Publisher).

The system provides an Installation – Online Letter Display algorithm type (C1-LT-DISP) that interacts with Documaker to render an online image of your letter. The following points describe what takes place when clicking **Display Letter**.

- The algorithm based on this algorithm type calls the letter extract algorithm on each letter template.
- The letter extract algorithm will construct the information that appears on the letter
- The constructed information is passed to document composition software that will reconstruct the letter's image in a PDF format.
- The system displays the PDF in a separate browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

The system provides an algorithm that interacts with BI Publisher to render an online image of a letter. The following points describe what takes place when clicking Display Letter when these sample algorithms are used.

- The sample Online Letter Display algorithm C1-ONLTR-RPT is executed.
- This algorithm calls the letter extract algorithm. The sample letter extract algorithm C1-LTREX-RPT is supplied for displaying letters via BI Publisher. This is not actually extracting data because when you use BI Publisher, the report itself is responsible for extracting and formatting the data. The base product algorithm supplied returns the appropriate report code to the Online Letter Display algorithm, which in turn returns the report code to the customer contact page.
- A user exit on the customer contact page invokes BI Publisher and displays the letter in a separate browser session.

> **NOT_SUPPORTED**
>
> Refer to Letter Print - BI Publisher Sample Welcome Letters - CI_LTRGN_ENG REPORT for more information.

Copyright © 2026, Oracle and/or its affiliates.

Reproducing The Letter Print Flat File

You can reproduce the flat file containing the information sent to your printing software at any time. Simply request the respective extract batch process and specify the run number associated with the historic run.

How To Reprint A Specific Letter

If you need to reprint a specific letter, navigate to Customer Contact - Main and turn on the **Reprint** switch (and save the customer contact). Alternatively, if your implementation has enabled the online creation of letter images, you can also click the **Display Letter** button on this page and then print the resultant PDF on your local printer.

Letter Data Area Organization

The following refers to logic supported by letter extract process C1-LTPEX – Letter Extract. Refer to The Letter Extract Processes for more information.

The letter extract records are defined using data areas. There is an organizing data area that includes a detail data area and optionally a characteristic data area. The detail data area contains the individual elements. Using the account record as an example, the organizing data area is “Account Letter Extract Record” and includes two other data areas, “Account Details Letter Extract” and “Account Characteristics Letter Extract”. This was done to:

- Ensure that characteristic records are extracted directly after their associated record.
- Properly sequence parent records followed by their characteristic then followed by a child record and its characteristic.
- Allow custom extensions to the detail record. See Extending Letter Record Types for more information.

The characteristic data area is only included on objects with a characteristic collection. For consistency, objects without a characteristic record still follow same pattern with an organizing and detail data area. There are two special records, the end of letter record and the summary record that do not follow this pattern.

Letter Extract File Structure

The following refers to logic supported by letter extract process C1-LTPEX – Letter Extract. Refer to The Letter Extract Processes for more information.

The letter extract file structure is impacted by the extract format. The extract batch, contains a File Format parameter that supports XML, CSV, and fixed position extract formats.

When the format is set to XML, the organizing data area as described in Letter Data Area Organization, is treated as a single schema. The included data areas are nested within the organizing data area. For example, an account record configured with two characteristics to extract will be nested in a single schema with a detail group and two occurrences of the characteristic list group.

Copyright © 2026, Oracle and/or its affiliates.

When the format is set to CSV or fixed position, the included data areas are treated as an individual schema so that each record is extracted on a separate row. For example, an account record configured with two characteristics to extract will produce three rows in the extract file.

Extending Letter Record Types

The following refers to logic supported by letter extract process C1-LTPEX – Letter Extract. Refer to The Letter Extract Processes for more information.

Each record type has a business object that defines the format and the logic to extract records of that type. The BO Extract Record plug-in spot contains the logic to populate the record's data area. This plug-in spot supports multiple algorithms. Each business object is delivered with one algorithm configured for this plug-in spot that contains the logic to populate the record type's elements.

As described in Letter Data Area Organization, there is a detail data area that contains the individual elements for a record type. If your implementation wants to extend a delivered record type, define a custom data area that extends the detail data area. The organizing data area should not be extended. To populate the custom data area, create an additional algorithm for the Extract Record plug-in spot. The record's schema is passed into the plug-in spot so subsequent algorithms have access to values defined by previous algorithms.

Maintaining Declarations

The declaration page allows you to define a declaration percentage for a combination of account, premise, service type, declaration type and date range. The declaration may be used by algorithms to adjust certain calculations in the system. For example you may give certain customers a reduced VAT rate. The declaration is used to indicate the reduced rate for the qualifying customer.

Open **Menu**, and then **Customer Information**, and then **Declaration**, and then **Add** to create a new declaration.

> **Note**
>
> **Applicable for UK market**. This page is intended for the United Kingdom market. Other markets may disregard this transaction.

Description of Page

The **Declaration Info** is a concatenation of important information about the declaration record. The **Declaration ID** is a system-assigned random number that uniquely identifies this record.

Use **Account ID** to indicate the account for this declaration.

Use **Premise ID** to indicate the premise for this declaration.

The **Service Type** controls the service agreements covered by this declaration. All service agreements for this account and premise whose service point is for this service type are covered.

Copyright © 2026, Oracle and/or its affiliates.

Use the **Declaration Type** to indicate the type of declaration. The system includes the declaration types CCL Relief and Reduced Rate VAT.

> **Note**
>
> The values for this field are able to be customized using the Lookup table. This field name is DCL_TYPE_FLG.

Enter the **Start Date** and **End Date** to indicate the effective period of this declaration. The start date defaults to the current date.

Enter the **Declaration Percentage**.

If this declaration is related to an external document, indicate the **Document Reference**.

This bottom of this page contains a tree that shows information about all the declarations linked to this account. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Service Tasks

The topics in this section describes service tasks.

- The Big Picture of Service Tasks
- Searching and Viewing Service Tasks

The Big Picture of Service Tasks

Service tasks are records that can be used to perform a variety of tasks. Examples include:

- Task-related records performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service.
- Task-related records performed by Oracle Utilities Customer Care and Billing to manage specific processing, such as net energy metering true-ups, or prepaid billing to create bill segments for smart meter prepaid service agreements.

For additional information about Service Tasks, see Defining Service Tasks Options

Searching and Viewing Service Tasks

Use this procedure to search for and view service tasks using the Service Task Query portal.

1. Select **Menu** , then **Service Task.**
2. Enter your search criteria.
   Base package search options include customer name, external ID and service task ID.

   Self-service task search criteria include service task type.
3. Click **Refresh.**

Copyright © 2026, Oracle and/or its affiliates.

4. In the search results list, click the link for the service task you want to view or edit. The selected service task opens in the Service Task portal.

Registration Points

- Understanding Registration Points
- Maintaining Registration Points

Understanding Registration Points

Deregulated markets operate by passing transactions between market participants to manage market processes. Markets use a unique market identifier for market participants to identify the customer or premise that the transaction applies to.

Some markets will identify a customer at a premise and the unique identifier will change with each new customer at the same premise. Other markets will identify the premise as the unique identifier and all customers at that premise will use the same identifier.

A registration point can manage a unique identifier at the premise level, service point level or the service agreement level. Therefore, providing the ability to operate in all markets.

Registration points can be associated directly with market transactions and/or market processes. Transactions can be implemented to trigger from changes to a registration point, or any object associated with the registration point. Transactions from Market Participants are received and validated against the registration point and can update the registration point directory or associated objects.

A registration point search portal provides a central location for all transactions associated with a customer and/or premise to be viewed in a single location.

For implementations operating in deregulated markets, refer to your implementation documentation for further details relating to registration point(s) constructed for your market(s).

Refer to Understanding Market for more details on markets.

Maintaining Registration Points

This portal is used to display and maintain a Registration Point.

Refer to Understanding Registration Points for more details.

To view and maintain registration points, open **Menu**, and then **Customer Information**, and then **Registration Point**, and then **Search**. You are brought to a query portal with options for searching for registration points.

Once a registration point record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are displayed:

- **Registration Point**
- **Registration Point Attributes**
- **Registration Point Characteristics**

Copyright © 2026, Oracle and/or its affiliates.

- **Market Messages and Market Processes**
- **Registration Point Relationships**

Refer to the zones' embedded help for more information.

Consumer Contracts

- Maintaining Consumer Contracts

Maintaining Consumer Contracts

This portal is used to display and maintain a Consumer Contract.

Refer to Understanding Consumer Contracts for more details.

To view and maintain consumer contracts, open **Menu**, and then **Customer Information**, and then **Consumer Contract**, and then **Search**. You are brought to a query portal with options for searching for consumer contracts.

Once a consumer contract record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are displayed:

- **Consumer Contract** shows the details of the consumer contract.
- **Consumer Contract Milestone** shows the significant dates on which specific actions will take place for the consumer contract.
- **Consumer Contract Product** shows the related contract product versions
- **Consumer Contract Change Request** shows the related consumer contract change request(s) that created and/or updated the consumer contract.

Refer the zones' embedded help for more information.

Customer Service Requests

- Understanding Customer Service Requests
- The Start Service Process and Request
- The Stop Service Process and Request
- The Transfer Service Process and Request
- Additional Information About the Start, Stop, and Transfer Processes
- Displaying Insights on Start and Transfer Service Processes
- Maintaining Customer Service Requests

Understanding Customer Service Requests

Copyright © 2026, Oracle and/or its affiliates.

Customer service requests can be used to orchestrate complex customer related processes that can create or update multiple objects.

- Customer service request person
- Customer service request account
- Customer service request premise
- Customer service request service agreement
- Customer service request consumer contracts
- Customer service request contract product
- Customer service request service location

An implementation can add custom components to use the customer service request objects to add or update persons, accounts, premises and contracts.

To see the available customer service request types in your implementation, refer to the Customer Service Request business objects in the application.

- About Starting, Stopping and Transferring Service Using Customer Service Requests

About Starting, Stopping and Transferring Service Using Customer Service Requests

The system is delivered with step-by-step processes to facilitate start, stopping and transferring service. A few different objects are used to comprise these processes. These objects are designed to work with each other – like a matched set.

A **Process Flow Type** controls the user experience by combining a series of panels that create the steps of the flow. The process flow guides the user to gather information based on configured business rule. It also orchestrates performing business logic contained in plug-in spots to progress the process flow. For example, it uses a chosen premise to then show the user the available services to start at that premise. Along the way, the process flow creates and interacts with an underlying transaction.

A **Customer Service Request** is a lifecycle-based transaction that is created by the process flow. The customer service request is comprised of a parent request and child objects to record, update, and provide processing for persons, accounts, service agreements, and premises. The customer service request lifecycle continues after the process flow is complete and until all associated services are activated, stopped, or canceled.

A **Customer Service Request Type** contains the business rules that the process flow and customer service request jointly interpret, apply, and enforce. The customer service request type's business rules settings are called controls. They along with the plug-in spots, define the type's behavior. Some configuration is used by the process flow and some by the customer service request. For example, the customer service request type has controls that impact the fields used to search for a person and a plug-in spot used to create pending start service agreements.

The Start Service Process and Request

When starting service, the system creates a start service process flow and assigns it a customer service request type. The process flow type (start service process) and customer service request type (start service request type or request type) are determined from the action method with Action Method Role of **Start Service**. See Action Method for more information

Copyright © 2026, Oracle and/or its affiliates.

about setting up action methods. The customer service (start service) request parent and child business objects are defined on the request type.

The system is delivered with the following objects that are used to start service

- Process Flow Type **C1-STARTSERVICEREQUEST**.
- Customer Service Request Type business object **C1-StartServiceRequestType**
- Customer Service Request business object **C1-StartServiceRequest**
- Customer Service Request Person business object **C1-CustSvcRequestPerson**
- Customer Service Request Account business object **C1-CustSvcRequestAccount**
- Customer Service Request Premise business object **C1-CustSvcRequestPremise**
- Customer Service Request Start SA business object **C1-StartServiceRequestSA**

The start service process can be initiated via BPA script (refer to script C1StaActMtdR).

Start Service Process

The start service process contains five steps or panels. The logic that is enabled is defined in the request type. The following overview of the four panels assumes the request type has enabled the described functionality. This is meant to illustrate the full features provided with the start service process. The request type can be configured to accommodate your business rules.

If your organization operates in multiple jurisdictions, it is likely that the **Control By CIS Division** flag is set on the Installation Record. When set, the premise search is presented before the process flow is started. This is done to get the CIS Division from the premise and pass it to the action method. This is useful when action method is used to associate customer service request types and/or process flow types to specific CIS Divisions.

Move-to Premise. This panel is used to search for the starting premise. This is the first step to ensure that the new or existing customer's Move-to Premise exists and is in your service territory. In some instances, such as when **Control By CIS Division** flag is set on the Installation Record, the Move-to Premise is known and is passed into the process flow. In this case, premise is already selected and can't be changed.

Customer Identification. This panel is used to create a new person and/or account or ensure a person's existing account is eligible for service. If the start service process began without an existing person in context, a person search is provided to ensure the person is in fact not in your system to help prevent creating duplicate person records. If no person is found, the information used to search for the person such as their name, phone number, or ID is used to create a new person. If an existing person is found, the start process then continues as if the person was provided when the process began.

Regardless if an existing person is being used or a new person is being added, the start process determines if a credit or ID check needs to be performed and if it does, performs it. It then checks to see if the person has accounts that are eligible for service. For example, accounts used for bankruptcy or accounts that have been dormant for too long might not be eligible for service. A control determines if a new account can be created even if an existing account is eligible for service. This is useful to prevent duplicate accounts for the same person, unless your business rules allow it. The start process then uses the previous information to determine if a deposit is required. For more information, please see Verifying Identity, Checking Credit and Requiring Deposits when Starting Service.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When specifying **First Name** and **Last Name**, the person's name is formatted as Last Name + <Person Name Separator Character> + First Name, if name separator is being used. Otherwise, person's name is formatted as First Name + <space> + Last Name.

Customer Class must be defined for a new account. The values are restricted to the customer classes defined on the customer service request type. If the **Control By CIS Division** flag is set on the Installation Record, **CIS Division** must also be specified. The values are restricted to the user's CIS Division(s). See Users and CIS Division for more information

> **Note**
>
> **Default note.CIS Division** is defaulted from the premise. If there is no CIS Division defined on the premise, the CIS Division on the customer service request type is used, if it has one. Otherwise, **CIS Division** is not defaulted. If the CIS Division is configured to ensure the same CIS Division is used on related accounts, premises and service agreements, it can't be changed. **Customer Class** is defaulted from the Installation Record based on the **Person Type** (Person or Business). If **CIS Division** is specified, the **Customer Class** defaults first from the CIS Division and then from the Installation Record, each based on the **Person Type** (Person or Business).

Before transitioning to the next panel, this panel interacts with the start service request to create person and account records. When a new person or account is added, their record will exist in the system after this point, even if this process is not completed.

Services to Start. On this panel the services to start are selected. After inputting a start date, the panel displays services to start in three different groupings.

If a deposit is required, the first sections is shown. The start process defaults the deposit service agreement request type, calls the deposit calculation plug-in spot on deposit class, and defaults installments information amounts if configured on the request type. The start process enforces rules pertaining to installment amounts. While not common, additional deposit SAs can be added.

The next section shows the eligible services to start at the premise. This is based on the premise's SPs. This section is built by calling the algorithm in configured in the **Determine Services to Start** plug-in spot defined on the request type. It returns the default SA Type(s) for each service point based on the existing or prior service or the initial services defined on the SP Type if this is the first service at this SP. The start SA box is usually checked by default and can be unchecked to start only some of the eligible services at the premise. In rare cases, the start SA box is unchecked by default if some services are on and some services are off at the premise. This is done to replicate the current service at the premise by default. Entries cannot be removed or added to this collection of eligible services to start. Field service notes can be entered for the premise or service point.

The third service section is where additional services such as charity can be started. Additional services can default from the request type or be added. The search is designed to exclude deposit and SP-based services shown in the prior sections.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the CIS Division is configured to ensure the same CIS Division is used on related accounts, premises and service agreements, only service agreement types for the current CIS Division are defaulted to the panel. Additionally, when manually adding service agreement types, the CIS Division is set to the current CIS Division and can't be changed. The current CIS Division is determined based on the following hierarchy:
>
> - Account
> - Premise
> - Customer Service Request Type

At the end of the panel, premise attributes such as indicating if life support equipment is at the premise or if there are specific field warnings relevant to the premise.

Person and Account Details. This panel is where all other person and account information is gathered. This includes all person contact information, other names, other person identifies, alternate mailing address, paperless billing enrollment, autopay enrollment and updates, adding other persons on the account, and more.

Review and Submit. This panel provides an opportunity to review the information before submission.

When the start process is finished, a **Summary** is shown. The start service information is copied to the start service parent and child requests. The start service process is completed, and the start service request takes over.

Start Service Request

The start service request lifecycle is comprised of this logic:

- The start service request first checks to see if there are pre-requites to starting service. See Introducing Your Business Rule Before Starting or Transferring Service.
- Once any prerequisites have been completed, the request applies any person and account updates gathered during the start service flow. It then creates the pending start SAs by calling the algorithm plugged into the **Process Services to Start** plug-in spot on the request type.
- The request then monitors the pending SAs for activation.
- Once all services are active, premise updates are applied.
- The start request is then complete.

> **Note**
>
> For more details, see the start service request business object detailed description.

The Stop Service Process and Request

When stopping service, the system creates a stop service process flow and assigns it a customer service request type. The process flow type (stop service process) and customer service request type (stop service request type or request type) are determined from the action

Copyright © 2026, Oracle and/or its affiliates.

method with Action Method Role of **Stop Service**. See Action Method for more information about setting up action methods. The customer service (stop service) request parent and child business objects are defined on the request type..

The system is delivered with the following objects that are used to stop service

- Process Flow Type **C1-STOPSERVICEREQUEST**.
- Customer Service Request Type business object **C1-StopServiceRequestType**
- Customer Service Request business object **C1-StopServiceRequest**
- Customer Service Request Person business object **C1-CustSvcRequestPerson**
- Customer Service Request Account business object **C1-CustSvcRequestAccount**
- Customer Service Request Premise business object **C1-CustSvcRequestPremise**
- Customer Service Request Start SA business object **C1-StopServiceRequestSA**

The stop service process can be initiated via BPA script (refer to script C1StpActMtdR).

Stop Service Process

The stop service process contains three steps or panels. The logic that is enabled is defined in the request type. The following overview of the two panels assumes the request type has enabled the described functionality. This is meant to illustrate the full features provided with the stop service process. The request type can be configured to accommodate your business rules.

Services to Stop. This panel is used to choose the services to stop for the account. After entering a stop date, the algorithm in configured in the **Determine Services to Stop** plug-in spot defined on the request type is called. It returns the stoppable services for the account. The services at the premise in context are selected to stop by default. Other premises and non-premise-based service can also be selected to stopped but are not selected by default. Non-premise-based services with a characteristic premise matching the premise in context are defaulted to stop.

Field service notes can be entered for the premise or service point where service is being stopped.

Additionally, as customers stop service at a premise, some premise attributes such as indicating if life support equipment is at the premise or if there are specific field warnings relevant to the premise may need to be updated. Premise updates can be made for any premise where service is being stopped.

Person and Account Details. This panel has the same features as the same panel in the Start Service Process. The options presented are controlled by the stop request type

Review and Submit. This panel provides an opportunity to review the information before submission.

When the stop service process is finished, a **Summary** is shown. The stop service information is copied to the stop service parent and child requests. The stop service process is completed, and the stop service request takes over.

Stop Service Request

The stop service request lifecycle is comprised of this logic:

- The stop service request applies any person and account updates gathered during the stop service flow. Services that are chosen to stop are transitioned to pending stop by the algorithm plugged into the **Process Services to Stop** plug-in spot on the request type.

Copyright © 2026, Oracle and/or its affiliates.

- The request then monitors the pending stop SAs to check when they have stopped.
- Once all services are stopped, the request applies any premise updates.
- The stop request is then complete.

> **Note**
>
> For more details, see the stop service request business object detailed description.

The Transfer Service Process and Request

When transferring service (starting and stopping service for the same customer), the system creates a transfer service process flow and assigns it a customer service request type. The process flow type (transfer service process) and customer service request type (transfer service request type or request type) are determined from the action method with Action Method Role of **Transfer Service**. See Action Method for more information about setting up action methods. The customer service (transfer service) request parent and child business objects are defined on the request type.

The system is delivered with the following objects that are used to transfer service:

- Process Flow Type **C1-TRANSFERSERVICEREQUEST**.
- Customer Service Request Type business object **C1-TransferServiceRequestType**
- Customer Service Request business object **C1-TransferServiceRequest**
- Customer Service Request Person business object **C1-CustSvcRequestPerson**
- Customer Service Request Account business object **C1-CustSvcRequestAccount**
- Customer Service Request Premise business object **C1-CustSvcRequestPremise**
- Customer Service Request Start SA business object **C1-StartServiceRequestSA**
- Customer Service Request Stop SA business object**C1-StopServiceRequestSA**

The transfer service process can be initiated via BPA script (refer to script C1TrnActMtdR).

Transfer Service Process

The transfer service process contains five steps or panels. The logic that is enabled is defined in the request type. The transfer process combines panels from the start and stop processes into one process.

Move-to Premise. This panel is similar to the Move-To Premise panel used in the Start Service Process. Since an account is known at the start of the transfer service process, the search result is limited to the account's CIS Division if the CIS Division is configured to ensure the same CIS Division is used on related accounts, premises and service agreements.

Services to Start. This panel is the same as the same panel used in the Start Service Process. The options presented are controlled by the transfer request type.

Services to Stop. This panel is the same as the same panel used in the Stop Service Process. The options presented are controlled by the transfer request type.

Person and Account Details. This panel is similar to the same panel in the Start and Stop Service Processes. The only notable difference is that this panel allows for the creation of a customer contact for the starting and stopping service. The options presented are controlled by the transfer request type.

Copyright © 2026, Oracle and/or its affiliates.

Review and Submit. This panel provides an opportunity to review the information before submission.

When the transfer service process is finished, a **Summary** is shown. The transfer service information is copied to the transfer service parent and child requests. The transfer service process is completed, and the transfer service request takes over.

Transfer Service Request

The transfer service request lifecycle is comprised of this logic:

- The transfer service request first checks to see if there are pre-requites to starting service. See Introducing Your Business Rule Before Starting or Transferring Service.
- Once any prerequisites have been completed, the request applies any person and account updates gathered during the transfer service flow. It then creates the pending start SAs by calling the algorithm plugged into the **Process Services to Start** plug-in spot on the request type. Next it creates the pending stop SAs by calling the algorithm plugged into the **Process Services to Stop** plug-in spot on the request type.
- The request then monitors the pending start SAs for activation and the pending stop SAs for stopping.
- The move-to premise is updated as soon as one of its SAs are Active. Each move-from premise is updated as soon as one of its SAs are Stopped.
- The transfer request is completed when all of the following are true:
  - All premises are updated
  - All SAs to start are Active
  - All SAs to stop are Stopped.

> **Note**
>
> For more details, see the transfer service request business object detailed description.

Additional Information About the Start, Stop, and Transfer Processes

The following explains some of the more intricate details performed during the start, stop, and transfer processes.

Override Mailing Address

By default, the main person on an account's mailing address source is the mailing premise on the account. Other person's on the account do not default the address source and one must be chosen. Aside from the mailing premise on the account, the system stores a mailing address information on the person and directly on the account person record, which is called Account Override. The account/person record indicates the address source for that account/person.

Your business rules may define where alternate mailing addresses are stored or when each is used. The request type defines which alternate location is used or if both are used. When adding an alternate mailing address, the address is stored in the preferred location or if both are enabled, Person or Account Override must be selected.

Existing data might not reflect your current business rules. For example, the request type is set to Account Override, but the address source is already set to Person. In this case, the process behaves as if both sources are enabled and a message is shown to indicate that the current

Copyright © 2026, Oracle and/or its affiliates.

address is stored in the source not supported by the request type. The process does not require the source to be changed.

Default Bill Route Type

When a new person and account are added or when a new person is linked to an account as an other person, the bill route type is defaulted from the installation record.

Paperless Billing

The processes allow customers to enroll in paperless billing if not already enrolled. If the current bill route type's routing method is email or fax, the customer is considered already enrolled. When enrolling, the bill route type defined on the request type is used.

Autopay Enrollment

The processes allow customers to enroll, check, or stop automatic payments. If enrolled, the details of the current record are shown. Changing the enrollment expires the currently effective and creates a new record effective the following day. To stop automatic payments, uncheck the box next to the enrollment details and the current record is expired.

Displaying Insights on Start and Transfer Service Processes

The Start and Transfer Service Processes can display insights about the premise, account and the services at the premise.

The insights are shown when a premise is selected from the search results for the Move-to Premise and on the header of subsequent panels – i.e. Customer Identification, Services To Start and Account Details.

The base product provides Insight Classes for each category of insights (premise-based, account-based and service-based), as well as Insight Types that can be linked to the Insight Groups for these Insight Classes.

The following are the steps to enable insights on Start and Transfer Service processes:

- Add an Insight Group for the **Start/Stop/Transfer Insights - Account-Based** Insight Class. Link the Highlight Account Info - Inline (C1-ACCTINFO-INLINE) Insight Type to the Insight Group.
- Add an Insight Group for the **Start/Stop/Transfer Insights - Premise-Based** Insight Class. Link one or more of the following insight types:
  - Highlight Unique Accounts Had Service At Premise - Inline (C1-UNACCTSP-INLINE)
  - Highlight Life Support/Sensitive Load on Premise - Inline (C1-PREMLSSL-INLINE)
  - Highlight Open Cases - Inline (C1-CASEOPEN-INLINE)
  - Highlight Open Business Flags - Inline (C1-OPBUSFLG-INLINE)
  - Highlight Field Activity Type - Inline (C1-FLDACTTY-INLINE)
  - Highlight Pending Field Activity - Inline (C1-PENDFA-INLINE)
  - Highlight Premise Link To Parent Premise - Inline (C1-PRMPNT-INLINE)
  - Highlight Child Premises - Inline (C1-PRMCHILD-INLINE)
  - Highlight Premise Linked to Multiple Accounts - Inline (C1-PRMACCTS-INLINE)
  - Highlight Landlord - Inline (C1-LANDLORD-INLINE)

Copyright © 2026, Oracle and/or its affiliates.

- Add an Insight Group for the **Start/Stop/Transfer Insights - Service-Based** Insight Class. Link the Highlight Premise Service Info - Inline (C1-PREMSVC-INLINE)

Maintaining Customer Service Requests

This portal is used to display and maintain a Customer Service Request.

Refer to Understanding Customer Service Requests for more details.

To view and maintain customer service requests, open **Menu**, and then **Customer Information**, and then **Customer Service Request**, and then **Search**. You are brought to a query portal with options for searching for customer service requests.

Once a customer service request record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are displayed:`

- **Customer Service Request** zone shows the details of the customer service request.
- **Customer Service Request Person** shows the related person(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Account** shows the related account(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Premise**

  shows the related premise(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request SA** shows the related service agreement(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Contract** shows the related contract(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Contract Product** shows the related contract product(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Service Location** shows the related service location(s). Select a record to go to the specific portal where the details can be viewed and maintained.
- **Customer Service Request Related Object** shows other related entities.

Refer to the zones' embedded help for more information.

- Maintaining Customer Service Request Person
- Maintaining Customer Service Request Account
- Maintaining Customer Service Request Premise
- Maintaining Customer Service Request SA
- Maintaining Customer Service Request Contract
- Maintaining Customer Service Request Contract Product
- Maintaining Customer Service Request Service Location

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Customer Service Request Person

This portal is used to display and maintain a Customer Service Request Person.

This portal is displayed when a record is selected from the **Customer Service Request Person** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Person** zone shows the details of the related person. Refer to the zone's embedded help for more information.

Maintaining Customer Service Request Account

This portal is used to display and maintain a Customer Service Request Account.

This portal is displayed when a record is selected from the **Customer Service Request Account** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Account** zone shows the details of the related account. Refer to the zone's embedded help for more information.

Maintaining Customer Service Request Premise

This portal is used to display and maintain a Customer Service Request Premise.

This portal is displayed when a record is selected from the **Customer Service Request Premise** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Premise** zone shows the details of the related premise. Refer to the zone's embedded help for more information.

Maintaining Customer Service Request SA

This portal is used to display and maintain a Customer Service Request SA.

This portal is displayed when a record is selected from the **Customer Service Request SA** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request SA** zone shows the details of the related SA. Refer to the zone's embedded help for more information.

Maintaining Customer Service Request Contract

This portal is used to display and maintain a Customer Service Request Contract.

This portal is displayed when a record is selected from the **Customer Service Request Contract** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Contract** zone shows the details of the related contract. Refer to the zone's embedded help for more information.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Customer Service Request Contract Product

This portal is used to display and maintain a Customer Service Request Contract Product.

This portal is displayed when a record is selected from the **Customer Service Request Contract Product** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Contract Product** zone shows the details of the related contract product. Refer to the zone's embedded help for more information.

Maintaining Customer Service Request Service Location

This portal is used to display and maintain a Customer Service Request Service Location.

This portal is displayed when a record is selected from the **Customer Service Request Service Location** zone on the Customer Service Request portal. Refer to Maintaining Customer Service Requests for more details.

The **Customer Service Request Service Location** zone shows the details of the related service location. Refer to the zone's embedded help for more information.

Customer Profiles

A Customer Profile stores insights that help build a deeper understanding of each customer, enabling more personalized and engaging experiences.

For example, based on information received from an external system, a profile may indicate that a customer is in a low-to-moderate income bracket. The system can surface this insight in the customer insights view along with relevant next steps, such as completing an affordability survey.

- Maintaining Customer Profiles

Maintaining Customer Profiles

This portal is used to display and maintain a Customer Profile.

Refer to Understanding Customer Profiles for more information.

To access this portal, open **Main Menu**, then **Customer Information**, then **Customer Profile**. You are brought to a query portal with options for searching for customer profiles.

Once a consumer profile has been selected, you are brought to the maintenance portal.

The **Customer Profile** zone shows the customer profile details and attributes.

Refer to the embedded help for more information.

Asset Management

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The asset management functionality described in this section is available with the following products:
>
> - Oracle Utilities Advanced Meter Solution (C2M-AMS)
> - Oracle Utilities Customer to Meter (C2M)
> - Oracle Utilities Customer Cloud Service (CCS)
> - Oracle Utilities Meter Solution Cloud Service (MSCS)

- Assets
- Asset Activities
- Asset Activity Generators
- Asset Completion Events
- Asset Inbound Communications
- Asset Locations
- Asset Replication
- Manufacturers
- Mass Asset Disposition Update
- Service History
- Shipping Memos
- Specifications

Assets

- About Assets
- Maintaining Assets

About Assets

An asset describes such objects as meters, poles, pipes, transformers, components, or any other material item owned or managed by an organization. Stock items that are "Tracked" can be linked to the Asset record when the tracked stock item defined on the Specification record is associated with the Asset record. A tracked stock item defined as hazardous material will also display within the **Maintenance** tab on the asset record.

Key aspects of an asset are as follows:

- Have monetary or functional value
- Able to be inspected
- Able to be serviced and maintained
- Includes a disposition history
- Could include one or more components

Copyright © 2026, Oracle and/or its affiliates.

- Includes a unique asset identifier

Both the Asset Number and Badge Number identified for a particular asset must be unique within the system. These numbers can be entered into the system manually or an administrator can configure individual asset types to automatically generate the asset number and/or badge number for an asset created from that asset type. Please refer to *Understanding Asset Types* for information on how to configure the system for automatic generation of asset and badge numbers.

Assets are always associated with a location. Refer to About Asset Locations for more information.

Assets typically exist within a structured hierarchy based on the relationship between the asset and its locations and organizations.

Fixed Assets

Fixed assets (or Capital assets) are generally company-owned long-term assets; having a record in the asset table with an asset ID. These assets are initially capitalized but then depreciate over time. All fixed assets must be associated with a cost since cost is part of the value of the fixed assets for depreciation purposes. Deprecation is currently not managed within **Oracle Utilities Work and Asset Management** however, **Oracle Utilities Work and Asset Management** can interface fixed assets, for example, property unit, quantity, in service date, value, and so on, to an external fixed asset management system. Fixed asset information is most often obtained during the receipt and accept process or through the construction work activity process however, there is an exception for plant based capital work. Plant based capital work is unique because it usually involves standalone assets that are very large and expensive and obtaining information on these assets fall outside of these processes, so a secured manual process can be used. This manual process enables users to populate specific fields on the **Asset Performance** tab, such as *Member Asset*, *Fixed Asset Number*, *Additional Cost*, *ERP Project Number*, and *ERP Subproject Task Number*, and send this information to the external fixed asset system using a secured **Send Asset Information Update** button.

To use this feature, implement the following:

- Define your Property Units (Groupings of fixed assets).
- Identify which **Work and Asset Management** assets are considered "Member Assets" (These are fixed assets that are not individually depreciated such as poles, conductors, or pipes).
- Ensure the ERP Project and Task Numbers are known (This is needed if there is a Project Integration. Sends specific fixed Asset information to ERP Project instead of Fixed Assets system).
- Add two new Access Modes to the appropriate User Groups (Update Fixed Asset Information (W1UF) and Send Fixed Asset Information (W1SF).
- Ensure the application services for Fleet Asset, General Asset, Infrastructure Asset, and Smart Meter have these access modes (Users that need access to this feature must have these application services and access modes).

Linear Assets

Setting up your assets so that they are recognized as a “system” can help you to manage maintenance. If your organization uses GIS, this functionality can also help to position your assets on the mapping system that you have in use. Please consider the following when configuring linear assets:

- The Asset Location Type and Asset Location determine where the linear asset is connected.

Copyright © 2026, Oracle and/or its affiliates.

- Linear assets include a “Connected to” field which allows you to define the connections between assets

  For example, a pipe system would consist of a pipe, connected to a manhole, connected to the next pipe, and so on.

Maintaining Assets

You use the **Asset** portal is used to display and maintain assets.

Refer to About Assets for additional functional information.

You can access the portal from **Menu > Asset Management > Asset**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset**: This zone displays the basic attributes of the asset.
- **Attached Components**: This zone displays a list of any components that are installed on the asset.
- **Disposition History**: This zone displays a log of the status and location changes for the life of the asset.

The following zones may appear as part of the portal's **Device** tab page:

- **Device**: This zone defines the basic attributes of the device
- **Device Configuration List**: This zone lists the device configurations for the device
- **Device Activities**: This zone lists activities related to the device in date descending order
- **Device Events**: This zone lists the most recent device events associated to the device in descending order
- **Device Exceptions**: This zone ists the first 50 VEE exceptions for all measuring components for the device, sorted by the exception creation date in descending order

The following zones may appear as part of the portal's **Maintenance** tab page:

- **Asset Service History**: This zone displays a log of the service history that has been recorded for the life of the asset.
- **Activity History**: This zones displays a log of the activities that have been performed on the asset over the life of the asset.

The following zones may appear as part of the portal's **Tree View** tab page:

- **Asset Hierarchy**: This zone displays the service point at which the asset is currently installed (if applicable). Clicking the broadcast icon opens the **Installed Assets** zone, which displays a tree view of all of the assets installed at the service point.
- **Currently Installed Assets**: This zone displays a list of all assets currently installed in this location.
- **Installed Assets**: This zone displays a tree view of all of the assets installed at the service point.

The following zones may appear as part of the portal's **Log** tab page:

- **Asset Log**: This zone displays the server logs for the asset.

Copyright © 2026, Oracle and/or its affiliates.

- **Asset Sync Requests**: This zone displays a list of synchronization requests related to the asset. Synchronization requests are used to synchronize the asset with an associated device.

> **Note**
>
> The **Asset** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

Working with Assets

A number of actions are available for an asset depending on its status:

Action	Allowed Statuses	Details
Store	In Receipt Installed In Repair Retired	Moves the asset to a storage (“out of service”) location and sets the asset’s status to “In Store”..
Reject	In Receipt In Store	Sets the status of the asset to “Rejected” (used if an asset fails an inspection)
Install	In Store	Installs the asset at an asset location (service point) and sets the asset’s status to “Installed”.
Repair	In Store Retired	Sets the status of the asset to “In Repair”.
Retire	In Store In Repair	Sets the status of the asset to “Retired”.
Pending Disposition	In Store Installed In Repair Retired	Sets the status of the asset to “Pending Disposition”.
Change Location	In Store	Changes the storage location of the asset.

> **Note**
>
> If an asset is deleted, its related device is also deleted.

Asset Activities

- About Asset Activities
- Maintaining Asset Activities

About Asset Activities

Copyright © 2026, Oracle and/or its affiliates.

Asset activities are used for creating field work tasks related to an asset or asset location, such as installation or removal of a meter to and from its service point. It may also be used for creating field work tasks for a specific location unrelated to an asset such as trimming trees near power lines, fixing a street light pole, repairing fire hydrant leaks, and so on.

The base package includes an asset activity type business object (W1-SOMActivityType) and a corresponding asset activity business object (W1-SOMActivity), used in the integration or consolidation of Operational Device Management activities with and Service Order Management activities related to field work.

When an asset activity (based on the asset activity type and asset activity business objects referenced above) is created and enters the “Communication in Progress” state, the activity is synchronized with Service Order Management, creating a service order field activity with a field task type that corresponds to the asset activity’s type.

Once the field activity has been created, Service Order Management is responsible for dispatching the request to the field work system. When the task is completed, Service Order Management receives the completion information from the field work system, completes the service order field activity and sends the completion information to Operational Device Management to complete the corresponding asset activity.

Asset activities (based on the (W1-SOMActivity business object) are defined by:

- **Status**: The current status of the asset activity
- **Emergency**: A flag that indicates if the asset activity is an emergency or nor
- **Originating Activity**: The activity that triggered the creation of the asset activity (if applicable)
- **Location**: The location where the asset activity is to be performed
- **Asset**: The asset associated with the asset activity
- **Activity Generator**: The asset activity generator that created the asset activity (if applicable)
- **Service Date/Time**: The date and time at which the field work associated with the asset activity is to be performed
- **Address Information**: The address where the field work associated with the asset activity is to be performed
- **Contact Information**: Contact information (customer name and phone number) related to the asset activity

Maintaining Asset Activities

You use the **Field Activity Query** and **Field Activity** portals to display and maintain asset activities.

Refer to About Asset Activities for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Activity**. You are brought to the **Field Activity Query** portal with options for searching.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity**: This zone displays the basic attributes of the asset activity.
- **Completion Information Entry**: This zone displays a list of completion events that must be processing to finalize the asset activity

Copyright © 2026, Oracle and/or its affiliates.

- **Attachments**: This zone displays a list of attachments (if any) related to the asset activity.

The following zones may appear as part of the portal's **Log** tab page:

- **Activity Log**: This zone displays the server logs for the asset activity.

> **Note**
>
> The **Field Activity** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

> **Note**
>
> For asset activities that have been synchronized with Service Order Management, you are brought to the **Activity** portal to view and maintain the selected record. See About Service Order Field Activities for more information.

Asset Activity Generators

- About Asset Activity Generators
- Maintaining Asset Activity Generators

About Asset Activity Generators

Asset activity generators are used for mass generation of activities. An Asset Generator Criteria business object is selected when creating asset activity generators, and is used for entering the criteria values for searching assets and asset locations that will be linked to the activities to be created. Such criteria includes location type, asset identifiers, location identifiers, address components, asset disposition, indication if asset is installed or not, asset type, specification and among others. Required information also includes, the activity type, disposition of the created activities, maximum number of activities that will be created, and an indication if the activities will be created via batch processing.

The base package the following set of activity generator business objects containing the various criteria:

- **Asset Criteria** (W1–AssetCriteriaActGenerator): used to define asset-related criteria used to select the assets for which activities will be created (for example, activities for all assets for a certain asset type, a specified current disposition or condition, or whose identifiers contain specific characters, and so on)
- **Asset List** (W1–AssetListActGenerator): used to define a list of assets for which activities will be created
- **Component Criteria** (W1–CompnCriteriaActGenerator): used to define component-related criteria to select the components for which activities will be created (for example, asset inspection activities for all assets for a certain component type, a specified current disposition or condition, or whose identifiers contain specific characters, and so on)
- **Component List** (W1–CompnAssetListActGenerator): used to define a list of components for which activities will be created

Copyright © 2026, Oracle and/or its affiliates.

- **Location Criteria** (W1–NodeCriteriaActGenerator): used to define location-related criteria to select the asset locations for which activities will be created (for example, activities for all locations where assets are current installed, locations of a specific type, or whose identifiers contain specific characters, and so on)
-

Asset activity generators are defined by:

- **Status**: The current status of the asset activity generator
- **Batch Generation Requested**: A flag that indicates if the activities created by the asset activity generator should be generated via batch processing
- **Generator Information**: Details related to the specific activity generator business object (Asset List, Asset, Location, etc.) used by the asset activity generator
- **Activity Information**: Details for the types of activities created by the asset activity generator, including activity type, default disposition, activation and work lead days (defaulted from the asset activity type), and the maximum number of activities to be created

Specific asset activity generators contain additional information based on their business object. For instance, asset activity generators based on the Location Criteria business object includes **Location Information** and **Address Information** sections.

Maintaining Asset Activity Generators

You use the **Activity Generator** portal is used to display and maintain asset activity generators.

Refer to About Asset Activity Generators for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Activity Generator**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Generator**: This zone displays the basic attributes of the asset activity generator.
- **Activity Information**: This zone displays summary information about the activities created by the activity generator.

The following zones may appear as part of the portal's **Activities** tab page:

- **Activities**: This zones displays a list of the activities created by the activity generator.

The following zones may appear as part of the portal's **Log** tab page:

- **Activity Generator Log**: This zone displays the server logs for the asset activity generator.

> **Note**
>
> The **Activity Generator** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

Copyright © 2026, Oracle and/or its affiliates.

Asset Completion Events

- About Asset Completion Events
- Maintaining Asset Completion Events

About Asset Completion Events

Asset completion events are used to execute business events within the application. For example, completion events may be configured for several types of asset processing such as asset update, state transition, replication, service history and activity creation.

These are created automatically by the inbound communication through the completion event types configured on the activity type of the activity that the inbound communication is attempting to complete.

> **Note**
>
> Since these records are most often created by the system, most users will only rarely be manually creating or updating asset completion events.

Asset completion events are defined by:

- **Status**: The current status of the asset completion event
- **Execution Sequence**: The sequence number for the asset completion event
- **Inbound Communication**: The asset inbound communication that created the asset completion event
- **Relationship Information**: The asset, component, location, or service history associated with the asset inbound
- **Event Information**: Information about the asset completion event, including date and time of completion and any comments

Maintaining Asset Completion Events

You use the **Completion Event** portal is used to display and maintain asset completion events.

Refer to About Asset Completion Events for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Completion Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Completion Event**: This zone displays the basic attributes of the asset completion event.
- **Completion Event Related Objects**: This zone displays objects related to the asset completion event, such as a device, service point, or initial measurement.

Copyright © 2026, Oracle and/or its affiliates.

- **Attachments**: This zone displays attachments, if any, related to the asset completion event.

The following zones may appear as part of the portal's **Log** tab page:

- **Completion Event Log**: This zone displays the server logs for the asset completion event.

> **Note**
>
> The **Completion Event** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

Asset Inbound Communications

- About Asset Inbound Communications
- Maintaining Asset Inbound Communications

About Asset Inbound Communications

Asset inbound communications are used to stage the completion of an asset activity. This is typically created automatically by the system following the completion of the corresponding field activity in the field work system (such as Oracle Field Service Cloud).

> **Note**
>
> Since these records are most often created by the system, most users will only rarely be manually creating or updating asset inbound communications.

Asset inbound communications are defined by:

- **Status**: The current status of the asset inbound communication
- **Initiating Activity**: The asset activity, typically a field activity, that triggered the creation of the asset inbound communication
- **Event Information**: Information about the completion event associated with the asset inbound communication, including date and time of completion and any comments
- **Relationship Information**: The asset, component, location, or service history associated with the asset inbound communication
- **Asset Identifier Information**: Identifier types and values used to establish a connection between an asset and a corresponding entity in an external system
- **Attached to Asset Identifier Information**: Identifier types and values used to establish a connection between a component and related asset, and a corresponding entity in an external system
- **Location Identifier Information**: Identifier types and values used to establish a connection between a location and a corresponding entity in an external system
- **Activity Type Identifier Information**: Identifier types and values used to establish a connection between an asset activity type and a corresponding entity in an external system

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Asset Inbound Communications

You use the **Inbound Communication** portal is used to display and maintain asset inbound communications.

Refer to About Asset Inbound Communications for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Inbound Communication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Inbound Communication**: This zone displays information for the asset inbound communication.
- **Communication Tree**: This zone displays a tree of objects related to the asset inbound communication, including its parent activity.

The following zones may appear as part of the portal's **Log** tab page:

- **Inbound Communication Log**: This zone displays the server logs for the asset completion event.

> **Note**
>
> The **Inbound Communication** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

Asset Locations

- About Asset Locations
- Maintaining Asset Locations

About Asset Locations

Asset locations define the physical location where an asset is installed. In Oracle Utilities Customer To Meter, asset locations are service points, and specify the address where a device is installed.

In Oracle Utilities Customer To Meter, asset locations are created when a service point is created, and are maintained using the **Service Point** portal. If the service point has a contact associated with it, a corresponding contact will be created for the asset location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with asset locations.

Note that an asset location is the physical location where assets or components can be installed, whereas Out of Service Locations are for receipt, storage, repair, retirement, transport (on vehicles) and other dispositions.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Asset Locations

You use the **Asset Location** portal is used to search for asset locations.

Refer to About Asset Locations for additional information about asset locations.

You can access the **Asset Location** portal from **Menu > Asset Management > Asset Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the **Service Point** portal to view and maintain the selected record.

> **Note**
>
> In Oracle Utilities Customer To Meter, asset locations are created and/or updated when a service point is created and/or updated. If the service point has a contact associated with it, a corresponding contact will be created for the asset location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with asset locations.

Asset Replication

- About Asset Replication
- Maintaining Asset Replication

About Asset Replication

As devices and other assets are received in large quantities, the **Asset Replication** portal allows for the receipt, inspection, storage and other processing of these assets in batches.

Using asset replication functionality your organization can complete processes such as:

- Replicating assets in the same lot/pallet
- Selecting assets for testing or inspection
- Mass updating asset disposition such as rejecting or accepting a lot or pallet

This processing creates new asset IDs, badge numbers and serial numbers for assets based on general information that is entered related to each batch. When a new replication event is created it is tied to a replication activity type to determine default settings and attributes for the replication. The actual asset information can be entered manually or uploaded via a CSV file.

Assets can then be moved between locations, or dispositioned using Mass Asset Disposition Update.

Maintaining Asset Replication

You use the **Asset Replication** portal is used to display and maintain asset replication activities.

Refer to About Asset Replication for additional functional information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal from **Menu > Asset Management > Asset Replication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Replication**: This zone displays the basic attributes of the asset replication activity.
- **Completion Event**: This zone displays a list of completion events related to the asset replication activity.

Manufacturers

- About Manufacturers
- Maintaining Manufacturers

About Manufacturers

Manufacturers are the companies that makes assets (devices, communication components, transformers, etc.).

An asset is related to a manufacturer through the asset's specification.

Each manufacturer can have zero or more models defined. Models for a single manufacturer can have diverse service types.

Maintaining Manufacturers

You use the **Manufacturer** portal is used to display and maintain manufacturers.

Refer to About Manufacturers for additional functional information.

You can access the portal from **Menu**, then **Asset Management**, then **Manufacturer**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Manufacturer**: This zone displays the basic attributes of the manufacturer.
- **Manufacturer Models**: This zone displays a list of models associated with the manufacture. To add or edit models available for the manufacturer, click **Edit** in the zone title bar.
- **Manufacturer Specifications**: This zone displays a list of specifications associated with the manufacturer. Use the **Specification** portal to add or edit the specifications associated with the manufacturer.

Mass Asset Disposition Update

You use the **Mass Asset Disposition Update** portal is used to update the disposition of a group of assets.

Copyright © 2026, Oracle and/or its affiliates.

More specifically, Mass Asset Disposition is used for moving many assets of the same type from one location to another location. In a mass disposition task, the target is always a single asset location. For example, if a pallet of meters has been received in the receiving asset location, mass disposition can be used to move this batch of assets to a storage asset location or an out of service location.

Refer to About Assets for additional functional information.

To use the **Mass Asset Disposition Update** portal:

1. Select **Menu > Asset Management > Mass Asset Disposition Update**.
   You are brought to a query portal with options for searching assets
2. Enter search criteria appropriate to the asset you wish to update and click **Search**.
3. From the list of search results, select the assets you wish to update and click **Change Disposition**.The **Mass Asset Disposition Update** screen opens.
4. Enter the date and time at which the update should take effect in the **Effective Date/Time** field.
5. Select the new disposition for the assets from the **Asset Disposition** drop-down list.
6. Select the new location for the assets from the **Location** drop-down list or field.
7. Select or enter other data as appropriate (based on the new disposition). For instance, if you select “Rejected” you must also select a reason from the **Reason** drop-down list.

Service History

- About Service History
- Maintaining Service History

About Service History

Service history is used to capture and record relevant information regarding service or maintenance on assets. Typically it is used to record inspection feedback, pass/fail details, downtime, parts failure information, maintenance or service logs, or other information regarding service on the asset.

Types of service history might include:

- Asset downtime
- Failure details
- Asset maintenance
- Visual inspection
- Leak detection
- Smart meter test
- Battery replacement
- ..., and so on.

Service history records can be manually created or automatically generated as the result of completion events being processed for activities. If the service history type has any schedule defined, a set schedule may also be established to create regular service history for repeated

Copyright © 2026, Oracle and/or its affiliates.

tasks such as inspection or regular maintenance. Service history types linked to asset types control which service history can be created for assets.

Maintaining Service History

You use the **Service History** portal is used to display and maintain service history records.

Refer to About Service History for additional functional information.

You can access the portal from **Menu > Asset Management > Service History**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service History**: This zone displays the basic attributes of the service history record.

Shipping Memos

- About Shipping Memos
- Maintaining Shipping Memos

About Shipping Memos

Shipping memos are records of stock items being shipped from an internal storeroom, whether from storeroom to storeroom or from storeroom to vendor.

Shipping memos can be created for both standard and tracked stock items and enables storeroom managers and shipping clerks a way to manage and track stock items being shipped between locations.

Some uses for shipping memo include the following:

- Tracking stock items returned to vendors for credit, replacement, or repair
- Helping to manage the return of repaired or replacement stock items back into inventory

Oracle Utilities Customer To Meter uses the following types of shipping memos:

- **General**: the shipping memo category that should be used when manually creating shipping memos

Oracle Utilities Work and Asset Management and Oracle Utilities Operational Device Management include additional types of shipping memos.

Maintaining Shipping Memos

You use the **Shipping Memo** portal is used to display and maintain shipping memos.

Refer to About Shipping Memos for additional functional information.

You can access the portal from **Menu > Inventory > Shipping Memo**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Shipping Memo**: This zone provides information about the shipping memo, such as shipping category, shipping memo number, ship to details, and tracking information.
- **Shipping Lines**: This zone provides links to the shipping line and stock items associated with the shipping memo.

Specifications

- About Specifications
- Maintaining Specifications

About Specifications

Specifications are used to define the manufacturer, model, and other information about a set of assets.

- Specifications can include attachments which support the specification definition or provide the actual specification from the manufacturer.
- Specifications may also include default in service and out of service maintenance schedules for assets or components linked to the specification. When the specification is associated with the asset or component, the related maintenance schedule becomes in effect for the asset or component.

Specifications are defined by the following:

- **Asset Type**: The asset type associated with the specification. This links the specification to assets (assets of the specified asset type reference the current specification).
- **Manufacturer**: The manufacturer associated with the specification.
- **Detail**: Additional details pertaining to the specification. Details can include
  - **Model**: The model number for the specification. Used with item specifications.
  - **Configuration Mapping**: Details of the template device used when creating devices (meters, items, or communication components) based on assets that reference this specification. This includes:
    - **Configuration**: The type of configuration
    - **Device Identifier Type**: The device identifier type (Asset ID, Badge Number, Serial Number, etc.) used to identify the template device
    - **Identifier Value**: The value for the specified device identifier type used to identify the template device (for example, the Badge Number of the template device)
- **Firmware Information**: Details concerning the firmware specifications associated with the specification (not used with analog meter specifications)

The base package includes the following specifications, each based on a different business object. The table below lists the base package specifications and the business objects for each.

Specification	Business Object
Analog Meter Specification	W2-ManualMeterSpecification

Copyright © 2026, Oracle and/or its affiliates.

Specification	Business Object
Item Specification	W1ItemSpecification
Communication Component Specification	W2-CommunicationComponentSpec
Communication Relay Specification	W2-CommunicationRelaySpec
Firmware Specification	W2-FirmwareSpecification
Analog Meter Specification	W2–ManualMeterSpecification
Smart Meter Specification	W2-SmartMeterSpecification

Maintaining Specifications

You use the **Specification** portal is used to display and maintain specifications.

Refer to About Specifications for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Specification**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Specification**: This zone displays the basic attributes of the specification.
- **Specification Assets**: This zone displays a list of assets that reference the specification.

Device Management

- Devices
- Device Configurations
- Measuring Components
- Measuring Component Comparison Periods

Devices

- About Meters
- About Items
- About Communication Components
- Maintaining Devices

About Meters

Copyright © 2026, Oracle and/or its affiliates.



Meters are a type of device, which can be physical or virtual objects, that can produce data to be handled by the system.

A meter can have one or more device configurations over time. A meter's measuring components (if applicable) are associated with the device via device configurations.

A meter's device category (defined as an option on the device's business object) defines the broad category to which a particular device belongs. This option is used by smart meter commands and other processes when determining how the system should behave when executing processes based on the category of device. For example, when viewing items via the Device portal, certain item-specific zones appear that are not displayed when viewing meters or communication components. The base package supports the following device categories for meters:

- **Smart Meter**: these meters are associated to a head end system and support smart meter commands. The measuring components associated to these devices are expected to be read very frequently. These meters support periodic estimation.
  Refer to About Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.
- **AMR Meter**: these meters may also be associated to a head end system but they do not support smart meter commands. The measuring components are read frequently.
- **Manual Meter**: these meters are seldom read and do not support smart meter commands.

Attributes used to define devices include the following:

- Basic information about the Device including type of device (Smart Meter, Manual Meter, and so on), Manufacturer, and Model
- One or more device identifiers such as serial number, badge number, meter number, etc.
- For smart meters the device can also specify a head end system and an incoming data shift. Note: these fields are only necessary when there is no fallback defined on the device type.

Refer to Initial Measurement Data for additional functional information about how the device's head end system and incoming data shift can impact processing.

About Communication Modules

Communication modules enable automatic meter reading for manual meters, and handle electronic transmission of measurement data and events for meters that are not capable or not enabled to handle this data. Devices of this sort are attached to manual meters as equipment, and in the case of modules that measure consumption or other data, are associated to the physical meter’s scalar measuring components.

Communication modules associated with a manual meter are defined in the **Device Equipment** list on the Manual Meter business object (D1-ManualMeter). See the embedded help for more information.

Creating Measuring Components for Communication Modules

Copyright © 2026, Oracle and/or its affiliates.

Measuring components used with communication modules must be scalar register measuring components (based on the D1-Register business object) and must be defined on the manual meter's device configuration.

If the communication modules will also record interval consumption, create an interval channel measuring component (based on the D1-IntervalChannel business object) add it to the manual meter's device configuration, and add it as the register measuring component's **Consumption Reference Measuring Component**.

Measurement and Device Event Processing for Communication Modules

The IMD Seeder and Device Event Seeder support payloads that reference the communication module identifiers currently attached to a physical meter, instead of the physical meter's identifier.

Based on the physical meter and the communication module currently attached to it, the seeders can identify the physical meter's register (or related interval) measuring component and use it for instantiating the initial measurement or device event.

Devices and Characteristics

Characteristics can be defined for devices using the **Device Characteristics** zone on the **Device** portal. Only characteristics of types defined for the device's type in the **Valid Device Characteristic Types** zone on the Device Type portal can be defined for the device.

About Items



While the most common type of devices are meters, there are many other devices that can be involved with a customer's service. These other devices are referred to simply as "items", and are used to represent lamps, poles, current transformers, backflow devices, pulse initiators, etc.

Items can be either "badged" or "unbadged". "Badged" items are those that have unique identifiers, and are represented by device records. Device records are not created for "unbadged" items. "Badged" items are installed at service points in much the same way as meters are installed (though they use an item-specific install event). For "unbadged" items, "multi-item" service points are used to define the number of each type of item that is installed at the service point.

These "badged" items are not directly measured and therefore do not have a measuring component. Instead there is configuration on the device type that will identify an effective dated average daily usage. Using the average daily usage the item's consumption will be calculated. The total consumption for a given period will always be equal to the average daily usage multiplied by the duration of the period. However, when the consumption for a period is turned into an interval curve there are two ways this can be done:

- **Use Straight Line**: the consumption will be evenly divided amongst the intervals (thus producing a straight line when graphed)

Copyright © 2026, Oracle and/or its affiliates.

- **Use Profile**: a profile measuring component will be used to align the consumption to a curve. This is useful because it can more accurately identify the hours where consumption has occurred.

Refer to About Service Points for additional functional information about "unbadged" items.

About Communication Components

Communication components are devices attached to meters. A communication component may handle electronic transmission of measurement data, events, and commands in situations in which the underlying meter is not capable or not enabled to handle this data. Devices of this sort are sometimes referred to as ERT (Electronic Receiver/Transmitter) meters, or communication modules (for example, the term "gas module" may refer to a communication module attached to a gas meter).

Maintaining Devices

You use the **Device** portal to display and maintain devices.

Refer to About Meters, About Items, and About Communication Components for additional information about specific types of devices.

You can access the portal by selecting **Menu**, then **Device**, then **Device**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Device** portal includes the following zones:

- **Device**: defines the basic attributes of the device
- **Device Configuration List**: lists the device configurations for the device
- **Device Attachments**: lists attachments related to the device
- **Device Activities**: lists activities related to the device in date descending order
- **Device Exceptions**: lists the first 50 VEE exceptions for all measuring components for the device, sorted by the exception creation date in descending order
- **Device Events**: lists the most recent device events associated to the device in descending order
- **Device Characteristics**: displays characteristics defined for the device. To add device characteristics, click the **Edit** link in the zone title bar.

> **Note**
>
> Only characteristics of types defined for the device's type in the **Valid Device Characteristic Types** zone on the **Device Type** portal can be defined for the device.

- **Device Entity Groups**: This zone lists the Entity Groups to assigned to the device. To add an entity group to the device, click **Add Entity Group to Device**. A dialog opens where you can select the entity group.

Copyright © 2026, Oracle and/or its affiliates.

Assets and Devices

In Oracle Utilities Meter To Customer, devices are created when an asset is created (or replicated) The type of device created is based on the template device identified in the **Configuration Mapping** section of the specification referenced on the asset or associated with the asset type.

Device identifiers for devices created via an asset can not be changed in the **Device** portal. These identifiers must be changed in the Asset portal.

Devices created via an asset can not be deleted in the **Device** portal. If a device's related asset is deleted in the Asset portal., the device is deleted as well.

Device Configurations

- About Device Configurations
- Maintaining Device Configurations

About Device Configurations

Device configurations represent specific configurations of a device as of a certain time.

Over time, a device can have many configurations. Device configurations are effective-dated allowing a device to retain its identifiers (such as serial number, badge number, etc.) even while the quantities it measures change over time.

Each device configuration can have one or more measuring components associated with it.

Attributes used to define device configurations include the following:

- The device configuration type
- The parent device
- The effective date and time of the device configuration
- The time zone in which the device operates

Refer to Multi Time Zone Support for additional information on the importance of the Device Configuration time zone.

Maintaining Device Configurations

You use the Device Configuration portal to maintain device configurations.

Refer to About Device Configuration for additional functional information.

You can access the portal from **Menu**, then **Device**, then, **Device Configuration**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Device Configuration**: Defines the basic attributes of the device configuration
- **Measuring Components**: Lists the measuring components for the device configuration

Copyright © 2026, Oracle and/or its affiliates.

- **Scalar Readings**: Displays a list of scalar readings for scalar measuring components associated with the current device configuration. To create a new reading for scalar measuring components associated with the device configuration, click **New Reading**. To edit an existing reading, click the edit icon for the reading you wish to edit.

NOTE: The Scalar Readings zone is available only with Oracle Utilities Meter Data Management.

Measuring Components

- About Measuring Components
- Physical Measuring Components
- Scratchpad Measuring Components
- External Measuring Components
- Maintaining Measuring Components
- Uploading Measurement Data
- Requesting Measurement Data

About Measuring Components

Measuring components are single points for which data will be received and stored in the system.

A measuring component can be associated to a device, which can have one or more measuring components, or it can be "virtual" or "stand-alone," meaning that it is not associated to a device.

The base package supports the following types of measuring components:

- **Physical**: Physical measuring components are those that physically exist, and that are linked to a device that can be configured differently over time. Interval channels and scalar registers are examples of physical measuring components.
  Note: The terms register and channel are synonyms for measuring component.
- **Standalone**: A standalone measuring component is used to record measurements for something that does not have a physical presence or is not associated to a particular device or service point. For example, you might create a standalone measuring component to record the average daily temperature supplied by a weather station. Standalone measuring components can be identified through **Standalone Identifier**.
- **Scratchpad**: Users create scratchpad measuring components based on physical measuring components to experiment with measurement manipulation functions.
- **External**: External measuring components are used to model measuring components for data that is not stored in the system. For example, some meters may have channels (measuring components) that capture Amps or other data not used my Meter Data Management or Customer to Meter, but that should be passed on to other external systems such as Oracle Utilities Data Intelligence. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist.
- **Aggregator**: An aggregator measuring component holds summarized usage from other measuring components. For example, aggregator measuring components could be configured to hold total consumption for each postal code within a service territory. Head

Copyright © 2026, Oracle and/or its affiliates.

end system processing statistics used by Oracle Utilities Smart Grid Gateway are stored as aggregated measurements for aggregator measuring components.
Refer to About Aggregations for additional functional information about aggregation measuring components.

Attributes used to define measuring components can include the following:

- The measuring component type
- The device configuration to which the measuring component is associated
- Details concerning how the measuring component is read, such as the number of digits, the type of read out (dials or digital), the meter multiplier, etc.
- **Initial Load IMD Ordering**: A flag that indicates if incoming initial measurements for the measuring component should be processed using Measuring Component Ordering. This is used most often in the case of daily scalar measuring components for which data is occasionally sent to the system out of order, but can also be used to ensure that "check channels" are processed before related channels. Applicable only to measuring component types used with Direct Measurement Processing (see Direct Measurement Processing and Measuring Component Ordering for more information).
- The VEE groups used for validation and estimation of measurement data.
Refer to About VEE for details about the types of VEE groups supported.
- **Enable ML Based Validation**: A flag that indicate (Yes or No) if initial measurements for the measuring component will be processed through a machine learning scoring model. IF set to "Yes", VEE processing for initial load data will only be used when data fails machine learning normalcy checks. See Machine Learning Anomaly Scoring for more information.
- **Anomaly Score Threshold**: The anomaly score threshold for the measuring components. This overrides the value set for the measuring component's type and/or in the MDM Master Configuration. This field is only visible when the **Enable ML based Validation** flag is set to "Yes" or **Enable ML Based Validation (Fallback)** flag on the measuring component type is set to "Yes". See Machine Learning Anomaly Scoring for more information.

Scalar vs. Interval

Beyond the four types described above, measuring components generally fall into one of two primary classes of: scalar measuring components, and interval measuring components.

- **Scalar measuring components** are measured at unpredictable intervals. For example, "once a month" is not a predictable interval as the amount of time between reads in unpredictable and inconsistent.
- **Interval measuring components** are measured at predictable intervals, such as every 15 minutes, every 30 minutes, every hour, etc.
- The terms "interval size" and "seconds-per-interval" (SPI) are used to define the size of an interval measuring component's intervals.
- Note: A device may have any combination of interval and/or scalar measuring components

Measuring Component Measurements

Measuring components are configured to measure specific types of quantities. These include:

- **Unit of Measure:** The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc.
- **Time of Use:** Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the

Copyright © 2026, Oracle and/or its affiliates.

greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

- **Service Quantity Identifiers:** Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components measure the same thing, but in different ways. A meter that measures both generation KWH and consumption KWH could use SQIs to differentiate between the two.

The combination of UOM, TOU and SQI define what a measuring component measures. TOU and SQI are optional, but UOM must be defined for all measuring components. For example, consider a meter (as illustrated in the image below) with two measuring components, both measuring the same unit of measure (kWh), but each measuring component measures consumption in different time of use (TOU) periods (peak and off-peak).

Subtractive vs. Consumptive Measurements

Another attribute that defines how measuring components measure quantities is the distinction between subtractive and consumptive measuring components.

A subtractive measuring component's usage is equal to the current reading (also known as the Stop or End Measurement or Reading) minus the previous reading (also known as the Start Measurement or Reading). To put this more simply:

Usage = End Reading - Start Reading

Most residential scalar meters are subtractive and in some cases interval meters can be subtractive as well . The table below lists a series of measurements for a subtractive measuring component.

Date	kWh (Reading)	Usage
01/01/2010	0	
01/31/2010	1000	1000 kWh
03/02/2010	1780	780 kWh
04/01/2010	2700	920 kWh

A consumptive measuring component's usage is equal to its current measurement. Most residential interval meters are subtractive. For scalar, consumptive measuring components are often used to measure demand, such as KW. The table below lists a series of measurements for a consumptive scalar measuring component.

Date	KW (Reading)	Usage
01/01/2010	0	0 KW
01/31/2010	10	10 KW
03/02/2010	10	10 KW
04/01/2010	13	13 KW

Physical Measuring Components

Physical measuring components are associated to the meters installed at service points. Through a device configuration there can be one-to-many measuring components in place at a given point in time. Much like other measuring components these are either scalar or interval and either consumptive or subtractive.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to About Measuring Components for additional information about scalar versus interval and consumptive versus subtractive measuring components.

The base product supports a core set of physical measuring components, as described in the following sections.

Interval Channel

These measuring components represent consumptive data that is recorded on a regular and consistent frequency as defined by the interval size. By definition interval data must fall on a specific interval boundary. For example, a meter that receives data once per hour cannot have a measurement at 1:30. Rather, the data must fall on either 1:00 or 2:00 not in between.

Data for interval channels is expected to be contiguous. However, in situations where it is not, there is no impact to the calculation of those intervals subsequent to the missing intervals. This is because the calculation of each interval is independent to all other intervals.

If there is a gap in the intervals it can be estimated through the periodic estimation process.

Interval Channel - Subtractive

These measuring components have behavior characteristics of both interval channels and registers. Like the interval channel they receive data on a regular and consistent frequency that is expected to be contiguous. Like a register they use a subtractive consumption calculation and provide rollover validation.

Unlike the interval channel when there are missing intervals there is an impact on the calculation of those intervals subsequent to the missing intervals. This is because the calculation of a given interval is always computed by subtracting its reading from the last known prior reading. The following table describes an example of that scenario:

Date/Time	Reading Condition	Reading	Measurement Condition	Usage
01/01/2010 12:00AM	Regular	0	Regular	0 kWh
01/01/2010 01:00AM	Regular	10	Regular	10 kWh
01/01/2010 02:00AM	Regular	20	Regular	10 kWh
01/01/2010 03:00AM	Regular	30	Regular	10 kWh
Missing measurements				
1/01/2010 12:00PM	Regular	120	Combined Quantity	90 kWh

Notice that the consumption for the 01/01/2010 12:00PM interval has become overloaded and covers the entire period for which no intervals have had been received. Also take note that the measurement condition is assigned a unique condition of Combined Quantity to signify that the consumption covers more than one interval.

A similar situation can arise when there has been an underestimation of consumption for a particular meter:

Copyright © 2026, Oracle and/or its affiliates.

Date/Time	Reading Condition	Reading	Measurement Condition	Usage
01/01/2010 12:00AM	Regular	0	Regular	0 kWh
01/01/2010 01:00AM	System Estimate	10	System Estimate	10 kWh
01/01/2010 02:00AM	System Estimate	13	System Estimate	3 kWh
01/01/2010 03:00AM	System Estimate	16	System Estimate	3 kWh
1/01/2010 04:00AM	Regular	40	Estimated	24 kWh

To help avoid these situations where an interval becomes overloaded with usage due to missing or estimated intervals the subtractive interval channel has the ability to initiate the calculation of missing intervals or re-calculation of estimated intervals when processing newly received intervals.

> **Note**
>
> Refer to Understanding Measuring Component Types link for additional information about how to configure subtractive interval measuring component types to perform prior and subsequent estimation.

Register

A register, often referred to as scalar, measuring component is used to represent infrequently received data from a meter. Typically these types of measuring components will only be read once per month aligning to the billing period for the customer.

The physical meter these measuring components represent often have dial readings where there are a finite number of digits that can register the consumption. When the maximum amount has been measured the meter will rollover and start counting once more from zero. To appropriately handle these scenarios this type of measuring component can be defined with a rollover tolerance that allows the system to accurately identify an invalid reading from a standard rollover. For example when a meter has 6 dials (a maximum of 999,999):

- Going from a reading of 999,990 to 20 would likely be valid and represent consumption of 30.
- Whereas going from a reading of 999,990 to 800,000 would be an invalid reading since it would represent consumption of 810,000.

Given the infrequent nature of these measuring components they are not eligible for the periodic estimation process.

Register - Auto-Read

An automatically read register processes nearly identical to a register. The key difference between these two types of measuring components is that the automatically read variant is expected to be read on a frequency of at least once per day.

These measuring components are eligible for periodic estimation since the information about their reading schedule allows for missing readings to be detected and estimated.

Copyright © 2026, Oracle and/or its affiliates.

Given the frequency these meters are read and the nature of subtractive consumption calculation they can be configured to re-estimate any past or future estimates when actual data has been received. The actual readings allow these estimates to be refined since the total consumption for the estimated period can be calculated (current reading - last reading prior to estimation).

> **Note**
>
> Refer to Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.

Scratchpad Measuring Components

Scratchpad measuring components provide users with a means to manipulate "scratchpad" measurement data without affecting existing "live" measurement data. Examples of measurement manipulation might include experimenting with the impact of executing the "spike smooth" function on an initial measurement, or adding or removing intervals to or from the measurement.

Scratchpad measuring components are defined by the same attributes as "regular" measuring components, with the following additional attributes:

- **User:** The name of the user who created the scratchpad. All scratchpad measuring components "belong" to a user.
- **Source Measuring Component:** The measuring component upon which the scratchpad is based.

Like "regular" measuring components, scratchpad measuring components can be related to a device through a device configuration, but scratchpad measuring components are not considered "active" for purposes of install events, and are not used in usage calculations for usage subscriptions.

Like "regular" measuring components, scratchpad measuring components and viewed and maintained using the Measuring Component portal, and measurement data for scratchpad measuring components can be viewed using the Initial Measurement portal or the 360 Degree View.

External Measuring Components

External measuring components are used to model measuring components for data that is not stored or used in the system. For example, some meters may have channels (measuring components) that capture data not used my Meter Data Management or Customer to Meter, but that should be passed on to other external systems such as Oracle Utilities Data Intelligence.

For example, if your implementation has interval meters that record 2 channels of data (kWh and Amps), but only the kWh channel is used for usage calculations, you would set up the kWh channel as a Physical interval channel and the Amps channel as an External channel. Data for the kWh channel would be processed as usual, but the data for the Amps could either be filtered entirely, or passed through to an external system.

Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring

Copyright © 2026, Oracle and/or its affiliates.

components that don't exist. See **Filtering Data for External Measuring Components** below for more information.

External measuring components are defined by the following;

- Device Configuration
- Channel ID
- External ID
- MDM External ID
- Number of Digits Left (Used for consumption calculations when passing data through to an external system)
- Number of Digits Right (Used for consumption calculations when passing data through to an external system)
- Channel Multiplier (Used for consumption calculations when passing data through to an external system)

The base product supports the following external measuring components:

- Interval Channel - External (D1-IntervalChannelExternal)
- Scalar Channel - External (D1-ScalarChannelExternal)

Like "regular" measuring components, external measuring components are related to a device through a device configuration, and can be considered "active" for purposes of install events, but are not used in usage calculations for usage subscriptions.

Like "regular" measuring components, external measuring components and viewed and maintained using the Measuring Component portal.

Filtering Data for External Measuring Components

Incoming data for external measuring components can be filtered (deleted) or passed through to an external system via object storage. Note that payload statistics track the number of initial measurements filtered and passed through.

Filtering data for external measuring components is supported with Direct Measurement Processing ONLY. See Direct Measurement Processing and Measuring Component Ordering for more information.

Maintaining Measuring Components

You use the **Measuring Component** portal to maintain measuring components.

Refer to About Measuring Components for additional functional information.

You can access the portal by select **Menu**, then **Device**, then **Measuring Component**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Measuring Component**: defines the basic attributes of the measuring component
- **Initial Measurement Data History:** lists the last 50 initial measurements for the measuring component in reverse chronological order.
- **Measuring Component Profile Use:** lists all profile factors on which the measuring component is referenced as a factor value.

Copyright © 2026, Oracle and/or its affiliates.

- **Measurements**: lists final measurements for the measuring component, including any derived values for each measurement. Note: depending on the type of measuring component this zone may be the Interval, Scalar, or Subtractive Interval variant.
- **Device Configuration Overview**: Displays an overview of the measuring component's parent device's configurations over time, including the measuring components associated with each configuration, and service points (along with the relevant install events) at which the device's configurations have been installed over time, if any exist.
- **Data Requests**: Lists data requests for the measuring component. Applicable only for Standalone Data Request measuring components (based on the Standalone Data Request MC business object)
- **Measuring Component Comparison Period Details**: Lists comparison periods for the measuring component (created when the current measuring component is compared to another measuring component via the Interval Proxy Day Estimation VEE rule)

Uploading Measurement Data

Measurement data for a measuring component can be manually uploaded in comma separated values (CSV) format. This can used to import data from external sources, such as weather systems or other market participants such as system operators in a settlement market.

This feature is accessed via the **Load IMDs (CSV)** option on the measuring component's context menu. When this option is selected, the **Online IMD CSV Upload** screen opens. From this screen users can select a CSV file containing measurement data to upload from a local desktop.

Import files can contain one or more initial measurements For interval data, breaks between initial measurements will be based on a gap greater than one interval size (SPI) between intervals. For scalar measurements, each row will create a separate initial measurement.

Refer to the embedded help on this page for details concerning the format of CSV files used for importing measurement data.

When processed, each initial measurement in the import file results in the creating of an initial measurement for the measuring component.

Requesting Measurement Data

The base package provides a mechanism by which standalone measuring components can request data from an external application. Examples of this include requesting data weather data from a weather station or zonal load data from an independent system operator (ISO) when used with Oracle Utilities Market Settlements Management. Measuring components of this type are referred to as "standalone measuring components".

Standalone Data Request Measuring Components

Requesting measurement data for standalone measuring components is supported by the "Standalone Data Request MC" (D1-StandaloneDataRequestMC) and "Standalone Data Request MC Type (D1-StandaloneDataRequestMCType) business objects.

Standalone data request measuring components use many of the same attributes as device-based measuring components, but also use the following unique parameters:

- **Standalone Identifier**: An identifier for the measuring component. This ID is used when importing data for standalone measuring components via the standard initial measurement loading process (see About Initial Measurement Data for more information).

Copyright © 2026, Oracle and/or its affiliates.

- **Processing Timetable Type**: A processing timetable type used when scheduling data requests. If not specified, a fallback processing timetable type can be defined on the measuring component's type.
- **Service Provider**: The external application to which data requests are sent. If not specified, a fallback external application can be defined on the measuring component's type.
- **Request Information**: Details about the last data request processed for the measuring component

Measurement Data Requests

Requests for measurement data can be scheduled or made manually by a user.

- Scheduled requests are triggered by the "Standalone MC Data Request - Monitor" batch process.
- Manual requests are triggered by clicking the **Request Data** button in the **Record Actions** section of the **Measuring Component** zone.

  This opens the Request dialog where te user enters a start and end date/time for the request.

Each of these invokes the "Calculate Schedule and Create Outbound Communication" algorithm which creates an outbound communication based on the processing method defined for the external application (Service Provider). When the external application sends back the requested measurement data, it can be received via the standard method for importing measurement data.

Configuring External Applications for Measurement Data Requests

To support measurement data requests, the following processing method must be defined for the external applications to which data requests are sent:

Processing Role	Processing Method
Standalone MC Data Request	How to Send MC Related Information Used to send the outbound communication to the external application. **Default Communication Out Business Object**: D1-StandaloneMCDataRequest (Standalone MC Data Request Communication Out The **Request Mapping Algorithm** and **Response Mapping Algorithm** options can be used to map the format of data both from the standard format to an external system format and from an external system format to the standard format. For example, when requesting for zonal load from an ISO the process would be: • Take the input of an outbound communication that has the measuring component's standalone ID and the date/times and put it into the zonal request format for the ISO • Take the input of an outbound message that has the response from the ISO with the intervals and convert it into the IMD format

In addition, the following objects must be configured to send data requests.:

- Communication Type for "Standalone MC Data Request Communication Type"
- External System
- Outbound Message Type

Copyright © 2026, Oracle and/or its affiliates.

- Message Sender

Measuring Component Comparison Periods

- About Measuring Component Comparison Periods
- Maintaining Measuring Component Comparison Periods

About Measuring Component Comparison Periods

Measuring component comparison periods are records of measuring component comparison. These are used by the Interval Proxy Day Estimation VEE rule to compare weather data to identify historical dates that most closely match current and/or future dates.

Measuring component comparison periods are created by the Interval Proxy Day Estimation rule, and are defined the following:

- **Primary Measuring Component**: The measuring component against which the comparison will be run
- **End Date** / **Start Date**: The date range that is compared to the period for the **Examined Measuring Component** (defined by the **Examined End Date** and **Examined Start Date**)
- **Rank**: The rank of this data point relative to the other comparison period stores for the same Measuring Component, Comparison Type, and End Date. The lower the rank, the better.
- **Examined Measuring Component**: The measuring component that will be compared to the **Primary Measuring Component**.
- **Examined End Date** / **Examined Start Date**: The date range that is compared to the period for the Primary Measuring Component (defined by the **End Date** and **Start Date**)
- **Lowest Quality Condition** : The lowest quality of condition code that was found for the measurements in the related period.
- **Comparison Metric**: The specific metric used to calculate the Comparison Result from the related Comparison Type)
- **Comparison Result**: The value result of the comparison.

Maintaining Measuring Component Comparison Periods

You use the **Measuring Component Comparison Period** portal to maintain measuring component comparison periods.

Refer to About Measuring Component Comparison Periods for additional functional information.

You can access the portal from **Menu > Device > Measuring Component Comparison Period**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Comparison Period**: This zone displays the basic attributes of the measuring component comparison period.

Copyright © 2026, Oracle and/or its affiliates.

Device Installations

- Service Points
- Service Point Quantities
- Install Events
- Correcting Cross Installed Meters
- Route Management
- Facilities
- Network Locations

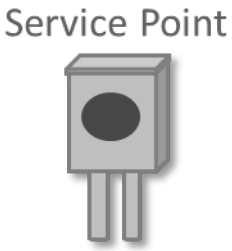
Service Points

- About Service Points
- Maintaining Service Points

About Service Points

Service points are physical locations (usually a meter socket) at which a company supplies service and are used in Oracle Utilities Meter Data Management to store information describing the type of service supplied and how it is measured. Devices are installed at service points, and the relationship between individual service points and devices can change over time. For example, over time different devices may be installed at a service point, and a device may be installed at different service points.

A service point may only have one device. If multiple devices at a service point are required, define a parent and child service point relationship to create a hierarchy of service points. On each service point, users will be able to specify a parent service point. An example of this feature would be to link each condominium in a building to a parent service point that represents the common areas of the building. Another example would be a deregulated service point that allows multiple devices per service point. In this example, the parent service point would be the market service point ID and the children service points would relate to the individual devices.



Attributes used to define Service Points can include the following:

- Basic information about the Service Point, including address, time zone, market, parent Service Point (if applicable), status, and main contact

Copyright © 2026, Oracle and/or its affiliates.

- Specifics related to whether or not the current Service Point supplies service to life support equipment, or if the load supplied to the Service Point is considered sensitive
- Information related to field work performed at the current Service Point, including any applicable warnings or instructions
- The measurement cycle, route, and route sequence for the Service Point

Service Point Access via Access Groups and Divisions

Access to service points (and objects and data related to the service point, see below) can be restricted based on Access Groups and Divisions.

Restriction via Access Groups

Service point data access restriction based on Access Group is enabled by default. Users can only access service points (and related objects and data) with an Access Group associated with the user via Access Role on the **Access Security** tab of the **User Preferences** portal. See The Big Picture of Row Security in the *Administrative User Guide* for more information about Access Groups and Access Roles. Restriction via Access Groups applies to all service point queries, including use of the **Service Point Search** zone, the **Route Management** and **360 Search** portals, and service point queries on other zones and portals.

Restriction via Division

To enable service point data access restriction based on Division, select "Yes" from the **Restrict Division** drop-down list in the **Control By Division** section of MDM Master Configuration.

When this restriction is in effect, users can only access service points (and related objects and data) with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. This applies to all service point queries, including use of the **Service Point Search** zone, the **Route Management** and **360 Search** portals, and service point queries on other zones and portals.

Unbadged Items

Items that do not have any unique identification are referred to as "unbadged." These types of items, unlike "badged" items, do not have a device to represent them. Instead they are associated to service points with the service point category of "multi-item." For these types of service points a list of "unbadged" items can be configured to represent the items that require usage to be calculated. This is an effective dated list that allows for changes over time to the type and count of items over time. The consumption of the list of items is either calculated using the service quantity associated to the item types listed or by using the installation override quantity that can be specified on the service point.

Refer to Multi Time Zone Support for additional information on the importance of the Service Point time zone.

Maintaining Service Points

You use the **Service Point** portal to view and maintain service points.

Refer to About Service Points for additional functional information.

You can access the portal by selecting **Menu**, then **Device Installation**, then **Service Point**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When searching for services points or creating new service points, if the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal will be available. In addition, the **Division** drop-down list will only list divisions associated with the user.
> See **Restriction via Division** in About Service Points for more information.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point**: defines the basic attributes of the service point
- **Service Point Attachments**: lists attachments related to the current service point
- **Device History**: lists all device configurations that have been installed for the service point, in reverse chronological order
- **Usage Subscriptions**: lists all usage subscriptions linked to the service point, sorted in descending order by Start Date/Time of Usage Subscription/Service Point
- **Service Point Entity Groups**: This zone lists the Entity Groups to assigned to the service point. To add an entity group to the service point, click **Add Entity Group to Service Point**. A dialog opens where you can select the entity group.
- **Child Service Points**: lists the first 50 children service points of the current service point (displayed only if the current service point has child service points)

The following zones may appear as part of the portal's **Geographic Information** tab page:

- **SP/Measurement Cycle Schedule Route List**: lists all the measurement cycle schedule routes linked to the service point.
- **SP Network Locations**: lists all network locations linked to any facility associated with the service point, ordered by start date/time.

The following zones may appear as part of the portal's **Multi-Item** tab page:

- **Multi-Items**: defines "multi-items" installed at the service point. "Multi-items" are used to define configurations in which multiple unbadged items of a set of one or more item types are installed at the service point (for example, 3 items of one type and 4 items of another type).
- **Item Details**: displays details for a broadcasted "multi-item" in the **Multi-Items** zone. This zone only appears if a "multi-item" is defined in the **Multi-Items** zone.

The following zones may appear as part of the portal's **SP Activity** tab page:

- **SP Activity History**: shows all activities linked to the service point. The results can be filtered based on activity type category and creation date/time.
- **SP Service Issue Monitors**: displays a list of service issue monitors and related service investigative orders related to the current service point, in reverse chronological order.
- **SP Business Flags**: displays a list of business flags related to the service point in reverse chronological order.
- **SP Suppression History**: displays a list of suppression activities related to the service point in reverse chronological order.

The following zones may appear as part of the portal's **Market Settlement** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **SP Market Participants**: lists all market participants linked to the current service point
- **SP Quantities**: lists service point quantities for the current service point

The following zones may appear as part of the portal's **Log** tab page:

- **Service Point Log**: This zone displays the server logs for the service point.

Service Point Quantities

- About Service Point Quantities
- Maintaining Service Point Quantities

About Service Point Quantities

Service point quantities are quantities that are stored at the service point level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given service point to the market wide coincidence peaks.

Service point quantities are defined by the following:

- **Service Point**: The service point for which the quantities were calculated
- **Service Point Quantity Type**: The Service Point Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Service Point Quantities**: Calculated quantities based on **Quantity Identifiers** configured on the Service Point Quantity Type (see Understanding Service Point Quantity Types for more information)

> **Note**
>
> There cannot be more than one Service Point Quantity for the same Service Point, Service Point Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

Maintaining Service Point Quantities

You use the **Service Point Quantity** portal to view and maintain service point quantities.

Refer to About Service Point Quantities for additional functional information.

You can access the portal from the **Main > Device Installation > Service Point Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Quantity**: This zone displays the basic attributes of the service point quantity

Copyright © 2026, Oracle and/or its affiliates.

Install Events

- About Install Events
- Maintaining Install Events

About Install Events

Whenever a device is installed at a Service Point, an installation event is created. Installation events capture the history of the devices that have been installed at a Service Point. This allows consumption for a Service Point to be calculated over time. In technical terms, installation events (or Install Events) link a specific Device Configuration to a Service Point.

There are specific Install Event business objects for different types of devices, including smart meters, items, manual meters, and communication components.

When creating Install Events for items, only "badged" items (those that are uniquely identified in some way) use item Install Events. "Unbadged" items do not use Install Events. Instead, the number of each type of item is specified in the "Multi-Item Detail" section of the Service Point.

While a device is installed at a Service Point, it may be turned off (and back on again). The installation event that records the original installation date and time also records the dates and times when the device has been turned on and off. When a device is removed, the original installation event is updated with the removal date and time.

In Oracle Utilities Customer To Meter, when a device with a related asset is installed at a service point, the install event updates the asset's disposition, changing its **Status** and **Asset Disposition** to "Installed" and its **Location** to the service point with an effective date matching the install event's **Installation Date/Time**.

Maintaining Install Events

This portal is used to display and maintain install events.

Refer to About Install Events for additional functional information.

You can access the portal from **Menu > Device Installation > Install Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Install Event**: defines the basic attributes of the Install Event, the on/off history for the Install Event, and removal information.

When creating a new record, the following options are available:

Name	Details	Business Object
Communication Component Install Event	This Install Event is intended for use with a Communication Component that is installed on top of a traditional Meter.	D1-CommComponentInstallEvent

Copyright © 2026, Oracle and/or its affiliates.

Item Installation Event	This Install Event records the installation of an Item (i.e. streetlight).	D1-ItemInstallEvent
Manual Meter Installation Event	This Install Event is used for manually read meters.	D1-ManualMeterInstallEvent
Smart Meter Installation Event	This Install Event is unique to smart meters and has a lifecycle that's maintained by remote meter commands.	D1-SmartMeterInstallEvent

In Oracle Utilities Customer To Meter, when a device with a related asset is installed at a service point, the install event updates the asset's disposition, changing its **Status** and **Asset Disposition** to "Installed" and its **Location** to the service point with an effective date matching the install event's **Installation Date/Time**.

Correcting Cross Installed Meters

Cross installed meters is the term for situations in which meters are deployed at an incorrect customer service point. Consider the following example: Meter A is supposed to be installed at Service Point A and Meter B is supposed to be installed at Service Point B, but Meter B was mistakenly installed at Service Point A, and Meter A was mistakenly installed at Service Point B. This means that both customers will be billed incorrectly (customer at Service Point A will be billed for usage at Service Point B and vice versa) until the system can updated to reflect the correct meter installations.

The Cross Installed Meters process provides a tool to re-configure the meters through a correction process. When a user initiates the Cross Installed Meters process, the system will update device installation event details (like status, date and time of installation), device configurations, and measuring components with the correct information. Note that this process does NOT include changing the physical location of the meters. It only changes the relationship between the meters and service points in the application.

This process can also be used to issue financial adjustments and/or cancel/rebill any bill segments that are impacted by the change in meters, and create customer contacts related to the changes. See Adjustments and Cancel / Rebill Incorrect Bill Segments for more information about adjustments and rebilling.

Cross installed meter processing is initiated, monitored, and maintained using the **Process Flow** portal and functionality.

See Process Flows in the *Administrative User Guide* for more information regarding the process flow functionality.

See Configuring Cross Installed Meter Processing in the *Administrative User Guide* for information about configuring administrative data required by the Cross Installed Meter process.

Initiating a Cross Installed Meter Process

Use the following procedure to initiate a cross installed meter process:

1. Select **Menu**, **Device Installation**, **Cross Installed Meter Process**, **Add**. The **Cross Installed Meters** process flow screen opens on the **Swap Meters** panel.
2. Use the **Cross Installed Meter Identification** section to search for and select the first meter (Meter 1).

Copyright © 2026, Oracle and/or its affiliates.

a. Select **Meter 1**.

b. Enter the search criteria for the meter. You can search by the service point's **Address**, **City**, or **Postal** code, or the device's **Badge Number**.

c. Click **Search**.

d. Select the appropriate meter. Information about the selected meter (including its service point, device, and device configuration) will appear in the **Meter 1** box.

e. To create a field activity to investigate the status of this meter (for instance, to confirm which meter is installed at the service point), click **Investigate** under the **Meter 1** section. This will create a field activity with a Field Task Type of that defined on the Cross Installed Meter Request Type. Note that you cannot proceed with the process flow until field activities created in this way have been completed.

3. Use the **Cross Installed Meter Identification** section to search for and select the second meter (Meter 2).

a. Select **Meter 2**.

b. Enter the search criteria for the meter. You can search by the service point's **Address**, **City**, or **Postal** code, or the device's **Badge Number**.

c. Click **Search**.

d. Select the appropriate meter. Information about the selected meter (including its service point, device, and device configuration) will appear in the **Meter 2** box.

e. To create a field activity to investigate the status of this meter (for instance, to confirm which meter is installed at the service point), click **Investigate** under the **Meter 2** section. This will create a field activity with a Field Task Type of that defined on the Cross Installed Meter Request Type. Note that you cannot proceed with the process flow until field activities created in this way have been completed.

Note: After selecting the second meter, the **Insights** box will display information about the meters based on the "Cross Installed Meters - Inline" (X1-CROSSEDMETER-INLINE) Insight Type.

4. Click **Save** to save your selection before making any changes. You can return to this process at a later time using the **Process Flow Query** portal.

5. Click **Swap Meters** to initiate the process of canceling/deleting the current incorrect install events, and creating new correct install events for each meter.

Note: The **Swap Meters** button is disabled until all field activities created from the process flow are in the Completed state.

This process also retrieves any impacted Accounts, Service Agreements, and frozen Bill Segments associated to the each meter's service point and device, and creates an Off Cycle Bill Generator for each service agreement based on the **Swap Meters Service Script** defined on the Cross Installed Meter Request Type.

If the **Create Customer Contact** drop-down list on the Cross Installed Meter Request Type is "Yes", this process also creates customer contacts based on the **Contact Class** and **Contact Type** defined on the Cross Installed Meter Request Type.

When the process completes the **Insights** box will display a "Swap Meters successfully performed." message, the **Swap Meters** button will be disabled, and the **Next** button will be enabled.

If an error occurs, error information will be displayed above the **Swap Meters** button.

6. Click **Next** to move to the **Correct Bill Segments** panel. Click **Previous** to return to the **Swap Meters** panel.

Copyright © 2026, Oracle and/or its affiliates.

The **Correct Bill Segments** panel displays a summary of impacted bill segments for each Off Cycle Bill Generator that can be corrected. To the right of each summary is a **View Bill Segments** link, and below each is a **Skip** button.

7. *Optional*. Click **View Bill Segments** to view additional details about the Off Cycle Bill Generator. This displays a table that includes the calculated **Original Amount**, **Rebill Amount**, and **Net Difference** for each bill segment (and transitions the Off Cycle Bill Generator to the "In Review" state). Note: This process can take a few minutes to process, depending on the number of bill segments.

   If the **Allow Adjustment** drop-down list on the Cross Installed Meter Request Type is set to "Yes", the panel displays a **Create Adjustment** button.

   If the **Allow Rebill** drop-down list on the Cross Installed Meter Request Type is set to "Yes", the panel displays a **Rebill** button.

8. Adjust, rebill, or skip each Off Cycle Bill Generator as appropriate:

   - Click **Create Adjustment** to create an adjustment for the total Net Difference, and transition the Off Cycle Bill Generator to the "Adjustment Created" state. Adjustments created in this way are based on the Adjustment Type defined on the Cross Installed Meter Request Type.
   - Click **Rebill** to rebill the impacted bills and bill segments and transitions the Off Cycle Bill Generator to the "Rebilled" state.
   - Click **Skip** to do nothing and transition the Off Cycle Bill Generator to the "Canceled" state.

   When bill segments for both service points have been either rebilled, adjusted, or skipped (and the Off Cycle Bill Generators have been transitioned to either "Rebilled", "Adjustment Created", or "Canceled"), the **Next** button will be enabled.

9. Click **Next** to move to the **Summary** panel. Click **Previous** to return to the **Correct Bill Segments** panel

   The **Summary** panel opens, displaying **Corrected Meter Installation** , **Corrected Bill Segment Information**, and **Customer Contacts** sections.

   The **Corrected Meter Installation** section displays the updated meter and service point information for each meter and service point, including new install events.

   The **Corrected Bill Segment Information** displays lists of corrected bill segments for each meter and service point.

   The **Customer Contacts** section displays the customer contacts created for each meter and service point.

10. Click **Finish** to mark the process flow as “Closed”.

Opening a Cross Installed Meter Process

Use the following procedure to search for and open a previously-created cross installed meter process:

1. Select **Menu**, **Process Flow**. The **Process Flow Query** portal opens.
2. Select "Cross Installed Meter" from the **Process Flow Type** drop-down list.
3. To search by user, enter a user ID in the **User** field.
4. To search by date range, enter the appropriate start and end dates in the **Update Date/ Time Range** fields.
5. To include Completed process flows in your search, check the **Include Completed** checkbox.

Copyright © 2026, Oracle and/or its affiliates.

6. Click **Search**.

   The **Process Flow Query** portal displays a list of process flows that match your search criteria.

   The **Resume** column displays a **Resume** button for process flows currently in process. The **Process Flow** column displays an ID link for completed/closed process flows.

7. To open an in-process process flow, click **Resume** for the process you wish to view and/or work with. The selected process flow opens on the **Swap Meters** panel in the **Process Flow** portal.

8. To open a closed process flow, click in the ID link in the **Process Flow**. The selected process flow opens on the **Summary** panel in the **Process Flow** portal.

Initiating Cross Installed Meter Process Flows from Field Activities

There can be situations in which a field crew discovers that a meter is installed at an incorrect customer service point. For instance a crew might be taking a manual meter reading or performing some other field activity when they realize that the error. Field crews can initiate a Cross Installed Meter process flow by providing the correct meter information (the meter's badge number) as part of the completion information sent back to the field activity (via an Inbound Communication).

To create a a Cross Installed Meter process flow in this way, the inbound communication must include a "CROSSEDMETER" Field Activity Remark Type (from the Field Activity Remark Type extendable lookup) in the Completion Details/Completion Information/Remark Types section, and the Badge Number of the meter physically installed at the service point in the Response Details/Message Details/Utility Completion Information/Cross Installed Meter section. When the inbound communication is received by the field activity and the field activity is completed, it creates a completion event based on the Create Meter Process Completion Event (X1-CreateCrossedMeterProcess) business object which instantiates a Cross Installed Meter process flow and an associated To Do entry. The Process Flow type and To Do type to create are derived from the Cross Installed Meter action method (See Configuring Cross Installed Meter Processing in the *Administrative User Guide* for information about configuring the Cross Installed Meter Action Method).

Inbound communications of this type can be created using the Field Activity Inbound Communication (D1-FieldActivityIBComm) Inbound Web Service or manually using the **Communication** portal.

When the new Cross Installed Meter process flow is created, the Service Point ID and Meter ID defined in the field activity are used to populate the "Meter 1" information. Note that the Investigate option is disabled for Meter 1 for process flows created this way. The actual meter badge number in the physical world and the service point where it's installed in the application are used to populate the "Meter 2" information.

See About Service Order Field Activities for more information about field activities. See About Communication In for more information about inbound communications. See Understanding Inbound Web Services in the *Administrative User Guide* for more information about using and invoking inbound web services.

Route Management

- About Route Management
- Working with Route Management

Copyright © 2026, Oracle and/or its affiliates.

About Route Management

Route management involves managing the order of the Service Points within measurement cycles and routes.

When creating Service Points, each Service Point can reference a measurement cycle, route, and its sequence within the route. Over time, as Service Points are added and removed to and from measurement cycles and routes, it may become necessary to change the sequence of Service Points within the measurement cycles and routes used in your implementation.

- A **measurement cycle** defines WHEN the Service Point is visited
- A **route** within a cycle defines a group of Service Points in a cycle that are visited by a meter reader
- A **sequence** within a route defines the physical position of the Service Point within a route
- A **schedule** specifies the dates on which Service Points are visited.

Manually read Service Point reference a measurement cycle, route and sequence within the route. A batch process creates SP/Measurement Cycle Schedule Routes for each Service Point, which link the dates on a measurement cycle schedule to a measurement cycle route defined for the Service Point. These define the specific date on which a meter reader will visit Service Points associated with a specific measurement cycle route, and the sequence in which the Service Points on that route are visited.

Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at Service Points in that cycle. Measurement cycles can have one or more associated routes used to collect measurements.

Measurement cycles can also be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle.

Measurement Cycle Schedules

Measurement cycle schedules define the dates on which devices are scheduled to be read for a given measurement cycle and the routes used to collect measurements for the measurement cycle.

Working with Route Management

This Route Management portal is used to change the sequence of Service Points within a measurement cycle and route and to transfer Service Points from one measurement cycle and route to another.

Refer to About Route Management for additional functional information.

You can access the portal from the **Main > Device Installation > Route Management**. You are brought to a query portal with options for searching. Once you enter your search criteria, a list of related Service Points is presented.

The following actions are available once a Measurement Cycle and Route are selected:

- **Renumber Sequence**: renumbers the sequence of selected Service Points for the route. Use the Renumber Sequence page to renumber the sequence of Service Points in a measurement cycle route.

Copyright © 2026, Oracle and/or its affiliates.

- **Transfer Measurement Cycle Route**: Use this button to transfer the selected Service Points to another route. Use the Transfer Measurement Cycle Route page to transfer the selected Service Points to another measurement cycle route. You can also renumber the sequence of Service Points in their new route as part of the transfer.

Facilities

- About Facilities
- Maintaining Facilities

About Facilities

Facilities represent the network node level closest to the Service Point. In terms of electric networks, facilities typically represent transformers.

While "Facility" is the generic name for this object, many of the Business Objects created within Oracle Utilities Meter Data Management have a more specific name. One good example is the Transformer (D1-Transformer) Business Object.

Maintaining Facilities

This portal is used to display and maintain a Facility.

Refer to About Facilities for additional functional information.

You can access the portal from **Menu > Device Installation > Facility**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Facility:** defines the basic attributes of the facility
- **Network Locations Related to Facility:** displays a list of network locations associated with the current facility in reverse chronological order.
- **Service Points Related to Facility:** displays a list of Service Points related to the current facility, in reverse chronological order.

Network Locations

- About Network Locations
- Maintaining Network Locations

About Network Locations

Network locations define the location of a facility within a larger network.

Network locations hold the network nodes that provide service to a facility. In the case of electric networks, network locations define the feeder and substation associated with a transformer (facility). Because these network nodes can change over time, a facility can have many network locations (this is an effective-dated relationship). For example, for electric

Copyright © 2026, Oracle and/or its affiliates.

Service Points, whenever the substation or feeder that provides power to a given transformer is changed, a new record that references the latest substation and feeder should be created.

Maintaining Network Locations

This portal is used to display and maintain a Network Location.

Refer to About Network Locations for additional functional information.

You can access the portal from **Menu > Device Installation > Network Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Network Location**: defines the basic attributes of the network location
- **Service Points Related to Network Location**: displays a list of Service Points related to the current network location, in reverse chronological order.

Measurements

- Initial Measurement Data
- Final Measurements
- Interval Snapshot Mapping
- Reader Remarks
- Periodic Estimation
- Measurement Reprocessing
- Consumption Sync
- About Interval Prices and Price Sets
- Maintaining Interval Price Sets

Initial Measurement Data

- About Initial Measurement Data
- Legacy Measurement Processing
- Direct Measurement Processing
- Machine Learning Anomaly Scoring
- CSV Upload
- Importing Data From External Systems
- Using the IMD Control Staging Portal

About Initial Measurement Data

Initial measurement data (IMD) is the term for measurement data in its initial (or raw) form.

Copyright © 2026, Oracle and/or its affiliates.

Measurements read from a measuring component are referred to as "initial measurement data" (or initial measurements) and are used to record how much of the quantity (defined by UOM, TOU, and SQI) measured by the measuring component was consumed.

Initial measurement data for scalar measuring components contain a single "reading" or value, while initial measurement data for interval measuring components can contain multiple readings, one for each interval that falls between the start time and stop time of the measurement.

At a simple level, initial measurement data goes through the following process:

1. Initial measurements are loaded into the system.
2. Initial measurement data is validated, edited, and estimated (VEE). Please refer to About VEE for more details on the VEE process.
3. Initial measurements are converted into final measurements and stored in the Measurement (D1_MSRMT) table.
4. If using Oracle Utilities Meter Data Management, final measurements are used to calculate usage (bill determinants, etc.). Please refer to About Usage Calculation for more details on the usage calculation process.

There are two ways in which initial measurement data is processed:

- **Legacy Measurement Processing** involves creating records in the database for each initial measurement upon receipt. These records are updated through each stage of processing until the initial measurement is finalized, and are retained in the system until archived and/or deleted. Processing is defined based on the initial measurement data business object. Legacy Measurement Processing can be used with Initial Load, Manual, and Estimation initial measurement data. See Legacy Measurement Processing for more information.
- **Direct Measurement Processing** involves processing incoming measurement data in memory, based on algorithms defined on an IMD type. If using Measuring Component Ordering, this also creates staging records for each initial measurement upon receipt (staging records are deleted once the measurement has been processed). Direct Measurement Processing is only supported for Initial Load measurement data. It does not support Manual or Estimation initial measurements. See Direct Measurement Processing for more information.

Only initial measurements can be edited directly by end users, and only when using Legacy Measurement Processing..

Quantities in an initial measurement can change during processing.

- "Pre VEE" quantities are consumption values derived from the measurements recorded by the head end system or meter reader with the appropriate installation and measuring component multipliers applied.
- "Post VEE" quantities are the "final" values, after VEE processing.

Changes in quantities can be based on a number of conditions, including:

- VEE rules have changed the quantities because they are missing or obviously wrong. In this, the Pre VEE values are adjusted based on the specifics of the VEE rules applied to the initial measurement to create the Post VEE values
- Manual changes by a user.

When using Legacy Measurement Processing, initial measurement data records contain both the original (Pre VEE) and final (Post VEE) versions of the quantities recorded by the measuring component.

Copyright © 2026, Oracle and/or its affiliates.

When using Direct Measurement Processing, if any quantities are changed during VEE processing, original (Pre VEE) quantities are captured along with the final (Post VEE) quantities in the Measurement (D1_MSRMT) table.

In addition to recorded consumption values, measurements also have condition codes, used to indicate the source and quality of a measurement. For example:

- Regularly recorded measurements might have a condition code of "Regular"
- Missing measurements might have a condition code of "Missing"
- Estimated measurements might have a condition code of "External Estimated" or "System Estimated" based on where the estimation was performed.

Both Pre VEE and Post VEE values have their own condition code, which can also change during VEE processing. For example, a measurement that was missing in the Pre VEE document will likely become "System Estimated" in the Post VEE document once VEE has been executed.

Legacy Measurement Processing

Initial measurement data (IMD) received from external systems is processed through two major stages:

- **IMD Seeder**: This is the common interface point that receives all IMD information and performs critical validations and conversions. During this process the IMD Seeder identifies the appropriate specific IMD business object for further processing. The choice is made based on the category of IMD being processed (typically initial load) and the type of measuring component the IMD is for. For example, an IMD for an interval measuring component would result in the Initial Load IMD (Interval) business object being selected. If for some reason the IMD Seeder encounters an error an instance of the IMD Seeder will be created to manage the resolution of that error.
- **Specific IMD**: Each specific IMD has the appropriate calculation and validation logic for the type of data that has been received. An IMD containing scalar data will be processed differently than interval data or subtractive interval data. However, all types of data will have a similar set of steps taken:
  - Any external system specific statuses for the measurement data are interpreted into the appropriate internal statuses
  - An initial consumption calculation is performed
  - VEE rules are executed
  - Final measurements are created

IMD Seeder Processing

The IMD Seeder receives IMD data in a common structure, however, the content of the data (identifiers, date/times, etc) can vary depending on the system and the particular device that generated it. To accurately translate the external content to values understood by the system the IMD Seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into IMD Seeder processing:

- **Identifier Types:** Each IMD must identify the device and measuring component the data has been generated from. The system supports a multitude of identifiers for both the device and measuring component. The service provider configuration specifies which identifier types are used for both the device and the measuring component which enables

Copyright © 2026, Oracle and/or its affiliates.

the IMD seeder to search for the correct destination for the measurement data. For example, a service provider might indicate that devices are identified by a serial number and measuring components are identified by a channel identifier.

- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has many processing methods that are used in the evaluation of the IMD.
  - Initial Measurement Creation - Maps a given measuring component type to the initial measurement BO that should be created
  - UOM Translation - Identifies the extendable lookup that should be used to map the service provider specific UOM to the internal UOM
  - TOU Translation - Identifies the extendable lookup that should be used to map the service provider specific TOU to the internal TOU
  - SQI Translation - Identifies the extendable lookup that should be used to map the service provider specific SQI to the internal SQI
  - Time Zone Translation - Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into IMD Seeder processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming IMD date/times will be stored without an adjustment for DST.

With these inputs the IMD Seeder is able to translate the external identifiers into the appropriate identifiers to tie the IMD data to a specific measuring component in the system.

Once a measuring component is identified it is then used to select the appropriate IMD business object using the "Initial Measurement Creation" processing method of the service provider.

IMD Seeder Critical Validations

For Initial Measurement Data (IMD) to continue on to specific IMD processing it must pass several critical validations:

- The measuring component must be identified.
- The IMD must contain the appropriate date/times for the type of IMD being processed: scalar must have an end date/time and interval must have both a start and end date/time.
- The interval list must match the number of intervals expected based on the IMD start and end date/times. Note: if there are too few intervals the missing intervals can be automatically added to the IMD without generating an error. How this is done depends on whether there are date/times for each individual interval. If there are date/times for each interval the missing intervals are placed where the gaps exist. Otherwise when there are no date/times for the individual intervals the missing intervals are added to the end of the interval list.

If any of the critical validations have failed the IMD processing will be halted and an instance of the IMD Seeder will be created

Copyright © 2026, Oracle and/or its affiliates.

Note: There is special logic that exists for handling the situation where either the IMD start date/time or the IMD end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time. Typically there are two 1AMs on that day, one is considered to be during DST and the other is not. To ensure that the appropriate hour is selected the IMD Seeder uses the count of intervals and final measurements that exist prior to the IMD to identify the appropriate hour.

> **Note**
>
> Additional detail on the logic to derive the service provider, device, and measuring component identification can be found on the D1-DER-SPRMC Algorithm Type.

IMD Seeder Time Zone Translation and Conversion

To provide the most accurate tracking of measurement data all initial measurement data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the IMD Seeder contains functionality to convert incoming date/times from a source time zone into the time zone of Oracle Utilities Meter Data Management (as indicated by the base time zone)

There are three components to identifying the source time zone of the incoming IMD:

- **Service Provider IMD Import Date/Time Format:** When the IMD date/times are sent in format that identifies the time zone for each date/time the date/times will be delivered to the IMD Seeder already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
  1. The time zone as indicated by the IMD data
  2. The service point time zone
  3. The device configuration time zone
  4. The measuring component time zone
  5. The base time zone

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

> **Note**
>
> Additional detail on the logic for date/time adjustments and validation of undercount and overcount can be found on the D1-DODTTMADJ Algorithm Type.

Specific Initial Measurement Data Processing

There are several categories of initial measurement data:

Initial Load

Copyright © 2026, Oracle and/or its affiliates.

When initial measurement data is received from external systems (e.g. Head End systems) it is processed using the initial load IMD type. These IMDs are configured to execute the VEE rules that as associated to the Initial Load VEE role on the measuring component (or measuring component type in the case of a fallback). For adapter specific variations of the Initial Load IMD there is logic to map external measurement statuses to measurement conditions.

Estimation

For various reasons a meter will not report measurement data for a certain period of time. When the system detects that a measuring component is missing final measurements, it can create an initial measurement to execute estimation rules. This type of initial measurement is referred to as an estimated initial measurement.

At a high-level, the estimation process is as follows:

- Missing final measurements are detected by the Periodic Estimation process
- Estimated initial measurements are created for the "missing" time period by the Period Estimation process
- Values and consumption for the estimated initial measurements are calculated by "estimation" VEE rules.

It's important to note that the processes that detect missing measurements do NOT themselves estimate consumption. Rather, these detection processes simply create an initial measurement and let the estimation VEE rules estimate the consumption for the initial measurement.

Estimation IMDs execute the VEE rules associated to the Estimation VEE role on the measuring component (or measuring component type in the case of a fallback).

> **Note**
>
> Refer to Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.

Manual Override

Manual override IMDs are used when the creation of an IMD is user initiated as can be done from the various functions available on some of the 360 Degree zones (e.g., Final Values Overlay, Interval - Initial Measurement Overlay, etc.). Unlike other types of IMD manual override allow the user to edit the initial measurement data prior to performing VEE. These IMDs are used for more advanced tasks where a user must specify, either directly or through a measurement function, the values for the initial measurements.

These IMDs are systematically created during the reconciliation process that occurs when IMDs are finalized for measuring components with subtractive consumption calculations. They are used to recalculate the consumption on a measurement subsequent to an IMD when the start reading for that subsequent measurement has been changed.

Manual override IMDs execute the VEE rules associated to the Manual Override VEE role on the measuring component (or measuring component type in the case of a fallback).

System

A system is generated by internal system processes that need to recalculate measurement multipliers. They do not perform VEE but they can be configured to do so.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to About Measurement Reprocessing for additional functional information about how the system IMDs are created.

Data Conversion

This type of IMD has only the processing necessary to propagate converted measurement data into the system. The data provided to these IMDs is expected to be the quality of final measurements. Therefore no VEE rules will be executed.

Scalar Initial Measurement Data Processing

Initial measurement data for subtractive scalar measuring components contain start and stop readings in addition to Pre and Post VEE usage. For example, a set of initial measurements for a subtractive scalar measuring component might look like the following:

Date/Time	Start Reading	Stop Reading	Pre VEE Usage	Pre VEE Condition	Post VEE Usage	Post VEE Condition
01/01/2010 12:00AM	0	1500	1500	Regular	1500	Regular
02/02/2010 4:11PM	1500	2100	0	Missing	600	System Estimate
03/03/2010 5:22 PM	2100	2900	800	Regular	800	Regular
04/01/2010 01:00PM	2900	3500	0	Missing	600	System Estimate

Subtractive measuring components can "rollover" when the reading exceeds the maximum value based on the number of dials on the measuring component . For example, a register with a 4 dials can record values up to 9999 before rolling over to 0000. When this occurs, consumption is calculated based on the following attributes and calculated values.

- **Rollover Threshold** is the percentage of the measuring component's dial capacity at which measurements for measuring components of this type are considered to have rolled over. Dial capacity is the largest value that can be recorded for the measuring component, based on the measuring component's number of dials. For example, a measuring component with 5 dials has a dial capacity of 99999.
- **Number of Dials** is the number of dials used to record values on the register, and is equal to the **Number of Digits Left** field configured on the measuring component.
- **Max Dial Capacity** is the maximum value for the number of dials, rounded up to the next whole multiple of 10 (or 10 raised to the power of the number of dials). For example, for a register with 4 dials, the max dial value is 10000.
- **Max Acceptable Difference** is the maximum acceptable consumption that can be recorded for the register. This is equal to the Max Dial Capacity multiplied by the rollover threshold. For example, the Max Acceptable Difference for a register with 4 dials and a rollover threshold of 90% would be 9000. If the consumption is greater than this value, the initial measurement is transitioned to the Error state.
- **Difference**: The difference between the Stop Reading and Start Reading, obtained by subtracting the Start Reading from the Stop Reading. If the Difference is less than zero (<0), then add the Max Dial Capacity to calculate Rollover.

Copyright © 2026, Oracle and/or its affiliates.

- **Rollover**: The adjusted consumption for a reading on a register that has rolled over. Only applicable if the Difference (Stop Reading - Start Reading) is less than zero (<0).
- **Consumption**: The calculated consumption for the reading, equal to either the Difference or Rollover. If the Difference is greater than or equal to zero, consumption is equal to the Difference. If the Difference is less than zero (<0), and the Rollover is less than or equal to the Max Acceptable Difference, the consumption is equal to the Rollover.

Example: Consider an initial measurement with the following attributes:

- Number of Dials: 4
- Rollover Threshold: 90 (%)
- Start Reading: 8900
- Stop Reading: 0500

For this reading,

- Max Dial Capacity = 10000
- Max Acceptable Difference = 9000 (10000 * 90)
- Difference = 0500 (Stop Reading) - 8900 (Start Reading) or **-8400**
- Rollover = 10000 (Max Dial Capacity) + -8400 (Difference) or **1600**
- Consumption is equal to **1600** (Rollover).

Each scalar IMD that is processed will perform a rollover validation based on the start reading. For those scalar IMDs that have measurements subsequent to the date being processed there will be an additional rollover validation forwards to ensure that the current IMD does not invalidate a future reading.

If the rollover validation fails when validating either backwards, forwards, or both then the system will attempt to automatically correct the situation by evaluating whether the measurements prior to or subsequent to the IMD can be removed. This analysis is done based on the condition of the measurement and the condition of the IMD. For example. If there is a rollover based on the start measurement and the start measurement is a system estimate but the IMD is regular then the start reading will be bypassed and the system will search backwards until:

- It identifies a measurement with a reading that does not cause a rollover when used as the start reading for the IMD
- It encounters a measurement that causes a rollover but is also regular and therefore cannot be bypassed. At this point the IMD would be placed in an error status.

When a rollover error is identified that cannot be automatically corrected there are a few options available for fixing the situation:

- The "Override Condition" functionality can be used. This is an action button available on the IMD from within the Error status. It allows the user to provide a higher quality condition for the IMD to force the rollover process to bypass the measurement (either prior to or subsequent to the IMD) that was the source of the rollover threshold error. For example, if the start read was a regular measurement and the IMD was also a regular measurement the user could update the IMD to have a condition of "Super" which would allow the regular start reading to be bypassed.
- A Manual Override IMD could be created to adjust the measurements surrounding the IMD to ensure that the rollover error is no longer encountered.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When a measurement is bypassed by the rollover automatic correction logic it will be set to Do Not Use when the IMD is finalized.

A scalar IMD can generate a manual override IMD during the finalization process when the need for reconciliation is identified. This happens whenever an IMD is received that occurs prior to an already existing scalar measurement. Since the newly processed IMD will introduce a new start reading for the existing subsequent measurement it is necessary to use the manual override IMD to recalculate the consumption. For example, imagine a scalar measuring component had the following readings:

Date/Time	Start Reading	Stop Reading	Pre VEE Usage	Pre VEE Condition	Post VEE Usage	Post VEE Condition
01/01/2010 12:00AM	0	1500	1500	Regular	1500	Regular
02/02/2010 4:11PM	1500	2100	0	Regular	600	Regular
04/01/2010 01:00PM	2100	3500	1400	Regular	1400	Regular

When the February reading is finally received it will spawn a reconciliation Manual Override IMD to correct the consumption of the April reading given the new start reading:

Date/Time	Start Reading	Stop Reading	Pre VEE Usage	Pre VEE Condition	Post VEE Usage	Post VEE Condition
01/01/2010 12:00AM	0	1500	1500	Regular	1500	Regular
02/02/2010 4:11PM	1500	2100	600	Regular	600	Regular
03/03/2010 5:22 PM	2100	2900	800	Regular	800	Regular
04/01/2010 01:00PM	2900	3500	600	Regular	600	Regular

> **Note**
>
> These reconciliation IMDs are generated during the finalization of the initiating IMD and must finalize themselves. If for some reason the reconciliation IMD does not complete it will prevent the initiating IMD from finalizing as well. Typically the reason for this issue will be a VEE exception that is being raised against the reconciliation IMD. The reason this rule is enforced is to ensure that the quantity of a measurement always aligns to the reading of that same measurement. If this situation arises the data source of the IMD can be used as an input to the VEE rule eligibility to skip certain rules when the IMD is a reconciliation IMD since the reconciliation process add a data source of "Generated by Reconciliation" to each reconciliation IMD.

Copyright © 2026, Oracle and/or its affiliates.

Interval Initial Measurement Data Processing

Given the consumption calculation for interval channels is done using the consumptive method the Interval IMD variants do not contain much special processing outside of the general pattern described in Processing Initial Measurement Data.

Subtractive Interval Initial Measurement Data Processing

Initial measurement data for subtractive interval measuring components readings and usage for each interval in addition to Pre and Post VEE usage. Furthermore, there is a condition for both the reading and the usage which allows for more accurate recalculation of intervals. The start reading for each interval is considered to be the reading from the prior interval. For the first interval the start reading will be identified from the measurement data for the measurement immediately prior to the IMD. For example, an initial measurement for a subtractive interval measuring component might look like the following when we assume a start reading of 1490:

Date/Time	Pre VEE Reading / Condition	Pre VEE Usage / Condition	Post VEE Reading / Condition	Post VEE Usage / Condition
01/01/2010 01:00AM	1500 / Regular	10 / Regular	1500 / Regular	10 / Regular
01/01/2010 2:00AM	0 / Missing	0 / Missing	1520 / System Estimate	20 / System Estimate
01/01/2010 3:00AM	1525 / Regular	5 / System Estimate	1525 / Regular	5 / System Estimate
01/01/2010 4:00AM	1540 / Regular	15 / Regular	1540 / Regular	15 / Regular

Subtractive interval IMDs can validate for rollovers based on the configuration of the measuring component. See Scalar Initial Measurement Data Processing for more details on how rollovers are calculated.

How subtractive interval IMDs identify rollover errors is identical to scalar IMDs. However, the resolution of those errors is much different. With scalar any bypassed measurements are handled at the point of finalizing the IMD and the measurements are set to Do Not Use. With subtractive interval any bypassed measurements are added to the IMDs interval list so that they are processed along with the IMD that identified them as causing a rollover.

Subtractive interval IMDs also must initiate the reconciliation of an interval that comes directly after the IMD being processed. Unlike scalar this is not handled through a separate IMD. Rather the next measurement after the IMD being processed is added to the current IMD and the reconciliation of the usage for that measurement is computed as part of the IMD. This allows for any validation errors that are encountered in the recalculation of that measurement to be resolved along with the IMD.

> **Note**
>
> There is one scenario that does result in a reconciliation IMD being generated. When the last interval of an IMD has the reading changed during the VEE process it will require that a reconciliation IMD be generated to adjust the usage of the next measurement after the IMD (if applicable).

The subtractive interval IMDs are responsible for ensuring that the reading for each interval is in line with the usage calculated. This maintenance is performed each time the IMD is run

Copyright © 2026, Oracle and/or its affiliates.

through the VEE rules. After which the IMD will examine each interval and identify which intervals require an update to the reading or quantity.

Initial Measurement Data Valid Date Boundaries

Meter Data Management can be configured to define an acceptable date range for initial measurement processing, providing a mechanism to restrict either future measurements or very old measurements from being processed by Meter Data Management.

Defining this date range is done via the following parameters in the **Device Measurement Parameters** section of the MDM Master Configuration:

- **Prevent Future Measurement Records**: Indicates if the system should prevent processing for measurements deemed to be "future" measurements, based on the **Number of Days Before Considered Future** parameter. If a measurement is deemed to be a "future" measurement and this is set to "Yes", the initial measurement is moved to the "Error" state.
- **Number of Days Before Considered Future**: The number of days in the future (based on the IMD From Date Time) before a measurement is considered to be a "future" measurement. If the I**MD From Date Time** for a measurement is more days in the future than this setting, the measurement is considered "future".
- **Prevent Old Measurement Records**: Indicates if the system should prevent processing for measurements deemed to be "old" measurements, based on the **Number of Days Before Considered Old** parameter. If a measurement is deemed to be an "old" measurement and this is set to "Yes", the initial measurement is moved to the "Error" state.
- **Number of Days Before Considered Old**: The number of days in the past (based on the IMD From Date Time) before a measurement is considered to be an "old" measurement. If the **IMD From Date Time** for a measurement is more days in the past than this setting, the measurement is considered "old".

Deferring Initial Measurement Data Processing

Meter Data Management can be configured to defer processing for initial measurements that are in the past, but that are not yet deemed to be "old" (based on the **Number of Days Before Considered Old** parameter on the MDM Master Configuration). This ensure that a meter that "wakes" up and sends a large payload of data, including old measurements, will not adversely impact initial measurement processing during critical processing windows.

Configuring the system for deferring initial measurement processing is done via the following parameters in the **Device Measurement Parameters** section of the MDM Master Configuration:

- **Defer Old Measurement Records**: Indicates if the system should defer processing of measurements whose **IMD From Date Time** falls between the **Number of Days Before Deferring** and the **Number of Days Before Considered Old** parameters. Note that when this parameter is set to "Yes", the **Number of Days Before Deferred** must be supplied.
- **Number of Days Before Deferred**: The number of days in the past (based on the **IMD From Date Time**) before a measurement should be deferred. Initial measurements with dates older than this setting but not as old as the **Number of Days Before Considered Old** will be deferred (the measurement is moved to the Defer state). If provided, this must be a positive number and must be less than **Number of Days Before Considered Old** setting.

Deferring initial measurement data processing requires that the Compare IMD Date/Times to Master Config Boundaries algorithm (D2-CMIMDDTMC) be defined as an Enter algorithm on the VEE Ready state for all initial measurement business objects for which deferring

Copyright © 2026, Oracle and/or its affiliates.

processing is to be enabled. See the **Detailed Description** of this algorithm for more information.

Initial Measurement Data Prioritization

By default, initial measurements are created and/or received into the system in an initial state of "Pending" and are transitioned to subsequent states via the "IMD Monitor - Physical Devices" Monitor process. This Monitor process can be skipped by setting the Execution Method flag on the initial measurement to "Real Time" (D1RT), in which case an Enter algorithm transitions the initial measurement to the next state in its lifecycle immediately. This feature can be used to prioritize processing of certain types of initial measurements.

The Execution Method flag can be set in initial measurements sent from the head end system, or can be inserted during Oracle Utilities Smart Grid Gateway Adapter processing.

Oracle Utilities Smart Grid Gateway Adapters set this flag to "Real Time" on all measurements created as the result of smart meter commands (Remote Connect, Remote Disconnect, or On-Demand Read) or completion events (including Field Activity Completion Events used with Service Order Management).

- Maintaining Measuring Component Dates

Maintaining Measuring Component Dates

There are several date/times systematically maintained on measuring components that aid in the efficient processing of initial measurement data as well as informing the periodic estimation process.

During initial measurement data finalization these dates are maintained by a series of algorithms based on the type of measuring component and type of initial measurement.

Important Measuring Component Dates

Each initial measurement is evaluated to determine if the following dates should be updated on the measuring component.

Last Contiguous Read Date/Time

This represents the last point where an uninterrupted series of measurements were received for the measuring component. Periodic estimation uses this field to generate estimates for any periods of missing measurements for the measuring component.

For interval measuring components this date will be updated whenever the current value falls between the initial measurement's From and To Date/Time fields (inclusive) in one of two ways:

1. If there are no measurements subsequent to the initial measurement being processed it will be set to the initial measurement To Date/Time
2. If there are measurements subsequent to the initial measurement and the initial measurement has filled a gap of missing measurements then it will be set to the Last Contiguous Measurement Date/Time that is subsequent to the initial measurement To Date/Time

For scalar measuring components there is an added check to ensure that the initial measurement’s To Date/Time is exactly one expected measurement after the current last contiguous read date/time

Adjusted Latest Measurement Date/Time

Copyright © 2026, Oracle and/or its affiliates.

For scalar meters this date is derived by assigning the last contiguous read date/time to an expected measurement date/time based on the measuring component type configuration for Expected Hours Between Measurements, First Daily Measurement Time, Early Threshold and Late Threshold.

For example, if a measurement is expected every 6 hours and the adjusted last contiguous read date/time on the measuring component is January 1st at 12:00:00AM, then the initial measurement's To Date/Time must align to the expected measurement date/time of January 1st at 06:00:00AM so the latest measurement date/time can be updated.

The expected measurement date/times are calculated using the expected hours between measurements and the first daily measurement time elements from the measuring component type. For example, if the expected hours between measurements was 6 hours and the first daily measurement date/time was 12:00:00AM, then the expected measurement date/times for each day would be:

- 12:00:00AM
- 06:00:00AM
- 12:00:00PM
- 06:00:00PM

To account for scalar reads that are not received for an exact date/time, the Early Threshold and Late Threshold elements of the measuring component type are used to align a given measurement to one of the expected measurement date/times.

Most Recent Read Date/Time

This represents the maximum measurement date/time that exists for the measuring component. This field is leveraged throughout the system to identify when certain logic must be executed. For example, if there are no measurements in the future to what is being processed then no logic dealing with future measurements need be executed.

It will be updated to the initial measurement's To Date/Time whenever the initial measurement's To Date/Time is greater than the current most recent read date/time.

Most Recent Non Estimated Read Date/Time

Represents the latest measurement processed that was not estimated. This is leveraged by the outage storm mode module.

It will be updated to the latest non-estimated measurement date/time in the initial measurement that is more recent than the current most recent non-estimated read date/time.

Identifying Data for Historical Extracts

Another important role for each of the algorithms that maintains measuring component dates is to identify whether the data being processed needs to be re-extracted for one of the existing measurement extract methods:

1. Specialized Measurement Extract
2. Specialized IMD Header Extract
3. DataConnect
4. Configurable Consumption Extract

If the data should be re-extracted a record will be created in the general processing table to be extracted at a later point.

Copyright © 2026, Oracle and/or its affiliates.

It is important to note that by default these checks are turned off for performance reasons. If you would like to capture historical changes for any of these extracts it will need to be turned on for each individual algorithm.

Date Maintenance Algorithms

The following algorithms are used to maintain measuring component dates.

D1-UPD-DTMC - Update Latest Measurement Date/Time on MC with Consumption Sync

This algorithm applies to initial load and manual initial measurements for interval measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the first time Periodic Estimation will be executed after initial conversion the Latest Read Date/Time has not been initialized for any measuring components the "No of Hours in Past to Retrieve Last Useable Measurement" option of the Feature Configuration Type "Measurement Data Option" should be configured. This enables logic that will defer the initialization of the Last Read Date/time to the Periodic Estimation process for any measuring component where it is not currently populated but does have existing final measurements. Please refer to About Periodic Estimation for more information.

If the initial measurement's device configuration is configured for related measuring component consumption synchronization this algorithm can optionally identify whether there are any consumption synchronization eligible measurements on the related measuring component (eligible measurements are defined as those whose condition falls between the bottom and top ranges for consumption sync or the additional condition code for consumption sync). When the initiating initial measurement is not "back dated" or the synchronization period does not extend past the initial measurement's date range the appropriate interval or scalar consumption sync activity will be selected based on the measuring component type of the related measuring component. When the initiating initial measurement is "back dated" and there is a contiguous period of synchronization eligible measurements that extends past the date range of the initial measurement the Gap Consumption Sync Activity will be created.

Lastly this algorithm can be configured to maintain outage related activities for the device. Since the receipt of initial load data for a device indicates it is now powered on and communicating if there are any outstanding outage or estimation suppression activities (initiated via widespread outage) they can be ended as of the from date/time of the initial measurement.

D1-UPD-DTMCF - Update Latest Date/Time on MC from a Future Measurement

This algorithm applies to estimation initial measurements for interval measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

Copyright © 2026, Oracle and/or its affiliates.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

As this algorithm is used on estimation initial measurement data it is not common that it will update the most recent non-estimated read date/time. However, there are circumstances where high quality estimations based on a check channel might be given a measurement quality in the regular range. In our documentation we sometimes refer to these as prorated measurements.

It is also important to note that this algorithm will not initiate consumption sync processing.

D1-UDTSCMCRE - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync

This algorithm applies to initial load and manual initial measurements for scalar measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Adjusted Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the last contiguous read date/time is not currently set for the measuring component then the initial measurement is assumed to be the first initial measurement received for the measuring component the Last Contiguous Read Date/Time will be set to the initial measurement To Date/Time. Furthermore, even if the Last Contiguous Read Date/Rime does not align to one of the expected measurement date/times, the Last Contiguous Read Date/Time and Adjusted Latest Contiguous Read Date/Time will be updated.

> **Note**
>
> If Auto-Read Scalar Periodic Estimation is being implemented for Auto-Read Scalar Devices that have existing measurement history, then the "No of Hours in Past to Retrieve Last Useable Measurement" option of the "Measurement Data Option" Feature Configuration Type should be configured. Logic will not initialize the latest measurement date/time information for any measuring component that does not have any latest measurement date/time information but does have existing final measurements. In these cases, the setting of the latest measurement date/time is deferred to the Auto-Read Scalar Periodic Estimation process. Please refer to About Periodic Estimation for more information.

Copyright © 2026, Oracle and/or its affiliates.

If the initial measurement's device configuration is configured for related measuring component consumption synchronization this algorithm can optionally identify whether there are any consumption synchronization eligible measurements on the related measuring component (eligible measurements are defined as those whose condition falls between the bottom and top ranges for consumption synchronization or the additional condition code for consumption synchronization). When the initiating initial measurement is not "back dated" or the synchronization period does not extend past the initial measurement's date range the appropriate interval or scalar consumption synchronization activity will be selected based on the measuring component type of the related measuring component. When the initiating initial measurement is "back dated" and there is a contiguous period of synchronization eligible measurements that extends past the date range of the initial measurement the Gap Consumption Sync Activity will be created.

Lastly this algorithm can be configured to maintain outage related activities for the device. Since the receipt of initial load data for a device indicates it is now powered on and communicating if there are any outstanding outage or estimation suppression activities (initiated via widespread outage) they can be ended as of the from date/time of the initial measurement.

D1-UPDDTSCMC - Update Latest Date/Time on Scalar Measuring Component

This algorithm applies to estimation initial measurements for scalar measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Adjusted Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the last contiguous read date/time is not currently set for the measuring component then the initial measurement is assumed to be the first initial measurement received for the measuring component the Last Contiguous Read Date/Time will be set to the initial measurement's To Date/Time. Furthermore, even if the Last Contiguous Read Date/Time does not align to one of the expected measurement date/times, the Last Contiguous Read Date/Time and adjusted Latest Contiguous Read Date/Time will be updated.

> **Note**
>
> If Auto-Read Scalar Periodic Estimation is being implemented for Auto-Read Scalar Devices that have existing measurement history, then the "No of Hours in Past to Retrieve Last Useable Measurement" option of the "Measurement Data Option" Feature Configuration Type should be configured. Logic will not initialize the latest measurement date/time information for any measuring component that does not have any latest measurement date/time information but does have existing final measurements. In these cases, the setting of the latest measurement date/time is deferred to the Auto-Read Scalar Periodic Estimation process. Please refer to About Periodic Estimation for more information.

Copyright © 2026, Oracle and/or its affiliates.

As this algorithm is used on estimation initial measurement data it is not common that it will update the Most Recent Non Estimated Read Date/Time. However, there are circumstances where high quality estimations based on a check channel might be given a measurement quality in the regular range. We sometimes refer to these as prorated measurements.

It is also important to note that this algorithm will not initiate consumption synchronization processing.

Direct Measurement Processing

Direct Measurement Processing is a method of processing incoming initial measurement data designed to reduce the volume of initial measurement records in the database. Rather than create and update initial measurement data records, Direct Measurement Processing processes initial measurement data in memory and creates final measurements in the Measurement (D1_MSRMT) table. With Direct Measurement Processing, records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table are created only if an error is encountered during processing (including VEE processing). If using Measuring Component Ordering (see below), Direct Measurement Processing also creates temporary staging records (in the IMD Control Staging (D1_IMD_CTRL_STAGE) table).

> **Note**
>
> Direct Measurement Processing is available with Native Implementations only, and only for processing Initial Load measurements. It cannot be used for Manual or Estimation measurements.

> **Note**
>
> The **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts for measurement data processed using Direct Measurement Processing.

IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing. There are two main categories of IMD types - seeder types (based on the Seeder IMD Type (D1-IMDSeederType) business object) and specific types (based on the Specific IMD Type (D1-IMDType) business object).

- **Seeder IMD types** represents an initial measurement data (IMD) seeder, which serves as the universal input into the system for external callers or for initial measurement payload processing. The main purpose of the seeder IMD type is to determine the most appropriate specific IMD type (such as interval or scalar, or a type used with a particular Smart Grid Gateway adapter) to use for further processing and prepare the input data for that IMD type. Additionally, a seeder IMD type is used to perform critical validation (such as an interval count validation), device and channel identification, mapping of external values (such as Unit of Measure and time zone) to internal values, and time zone conversion for date/time fields.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> There can only be on seeder IMD type active at any one time.

- **Specific IMD types** define the rules for processing a specific initial measurement data (IMD) record. Different specific IMD types contain rules for processing the different types of initial measurement data that can be received. For example, one IMD type defines the rules for processing a scalar measurement from the Landys+Gyr head end system while another IMD Type instance defines the rules for processing an interval initial measurement from the Sensus head end system. There are four types of algorithms supported for specific IMD types: Pre Processing, Pre-VEE, VEE, and Post-VEE. These algorithms are responsible for validating that the measurement has been correctly mapped to the IMD Type (Pre Processing), calculating consumption (Pre-VEE), performing VEE (VEE), writing to the final measurement table (Post-VEE), among other responsibilities.

IMD Types are defined for specific measuring component types in the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Creating Head End Systems in the *Administrative User Guide* for more information about processing methods.

See Understanding IMD Types in the *Administrative User Guide* for more information.

VEE Processing and Exceptions with Direct Measurement Processing

When using Direct Measurement Processing, VEE is performed by the Direct Measurement Processing Service based on algorithms defined for the initial measurement's IMD Type.

- If any quantities are changed during VEE processing, original (Pre VEE) quantities are captured along with the final (Post VEE) quantities in the Measurement (D1_MSRMT) table.
- If an initial measurement fails a VEE rule with only exceptions with a Severity of "Information", a header record is created in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table. This header serves as a parent record for the VEE exception(s). In this case, final measurements are created as part of the process.
- If an initial measurement returns an error during Pre-VEE processing or fails a VEE rule with a Severity of "Issues" or "Terminate", a record is created in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table for the initial measurement, and further processing is done using Legacy Measurement Processing. The type of initial measurement data record created is based the configuration of the IMD type.

The table below summarizes how VEE exceptions are created and and whether final measurements are created when using Direct Measurement Processing.

Scenario	Initial Measurement Data (D1_INIT_MSRMT_DATA) Record	VEE Exceptions Created	Final Measurements Created
No exceptions	No	No	Yes
Error in Pre VEE algorithms	Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval)	No	No

Copyright © 2026, Oracle and/or its affiliates.

Scenario	Initial Measurement Data (D1_INIT_MSRMT_DATA) Record	VEE Exceptions Created	Final Measurements Created
"Information" and "Issue" or "Terminate" VEE Exceptions	Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval)	Yes	No
"Issue" or "Terminate" VEE Exceptions	Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval)	Yes	No
Only "Information" VEE exceptions	Yes, Informational Exception Header (D1-InfoExceptionHeader)	Yes	Yes

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

Measuring Component Filtering

Direct Measurement Processing supports filtering of incoming data for external measuring components. See External Measuring Components for more information about external measuring components.

Measurement data for external measuring components can be filtered (in other words, dropped) or passed through to an external system via file system or object storage.

Filtering is controlled using the **IMD Handling** flag on the external measuring component types. This flag has two values, "Filtered" and "Pass through". As data is processed, the Direct Measurement Processing Service evaluates this flag to determine how to proceed.

- When this flag is set to "Filtered", the Direct Measurement Processing Service drops the measurement records, returns the initial measurement payload to the payload processing service (marked as "Filtered") so it will be captured in payload processing statistics.
- When this flag is set to "Pass through", the Direct Measurement Processing Service returns the initial measurement payload to the payload processing service (marked as "Pass through") so it will be captured in payload processing statistics, and the Payload Processing batch process sends the payload to a file system or object storage (for use in external systems).

  In the event of an error when processing initial measurements for "pass through" external measuring components, the measurement can be captured in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table for correction (similar to what happens if an error is encountered during processing of regular measurement data). Errors for "pass through" measurements can include errors related in critical validations during IMD seeder

Copyright © 2026, Oracle and/or its affiliates.

processing, status mapping, or consumptions calculation. Processing of "pass through" measurements in error is handled via "Pass through" IMD types and measurements are captured in "Pass through" initial measurement business objects. When the error is resolved, the measurement enters the "Waiting for Extract" status, and a batch process will retrieve the payload and send it the file system or object storage.

Measuring Component Ordering

In some implementations, it's important that measurement data is processed in a specific order. For example, when using a scalar channel as a "check channel" to estimate/validate a non-primary interval channel, it's important to process the scalar channel first. Another example are implementations in which measurement data for some (usually daily) scalar measuring components is occasionally sent to the system out of order. For example, data for four days in January might arrive in the following order:

- January 5
- January 2
- January 3
- January 4

By default, Direct Measurement Processing processes initial measurements as they are received into the system, meaning that the data for the January 5 measurement would be updated three times as each of the measurements that came after it is processed.

In cases like these, measurements should be processed based on the initial measurement's Start and Stop dates. In Legacy measurement processing, situations like this are handled by the IMD Monitor batch processes (such as D1-IMD and D1-IMDV2), which processes the data based on Start and Stop dates rather than in the order in which the data was received in the application. For Direct Measurement Processing, this can be supported through Measuring Component Ordering.

Measuring Component Ordering can be used with specific measuring components or measuring component types, and is defied for measuring components or measuring component types using the **Initial Load IMD Ordering** or **Initial Load IMD Ordering (Fallback)** flag (respectively). Initial measurements marked for Measuring Component Ordering via this flag are captured in the IMD Control Staging table with a status of "Ready to Process". Lists of these initial measurements can be viewed in either the **Initial Measurement Data History** zone of the **Measuring Component** portal, or by searching by measuring component or device or IMD identifier using the **Initial Measurement Data Search** zone in the **IMD Query** portal. Individual initial measurements in the "Ready to Process" state can be viewed using the **IMD Control Staging** portal (see Using the IMD Control Staging Portal for more information about using that portal).

"Ready to Process" records can be processed in one of two ways:

- One at a time from the **IMD Control Staging** portal
- Using the Process IMD Control Stage Data (D1-IMDCS) batch process. This batch process uses the following parameters:
  - **Value to restrict to IMD Control Stage Status**: Designates the status of records to process. When processing Measuring Component Ordering initial measurements, this should be set to "D1RD" ("Ready to Process").
  - **Enable sequence handling**: Designates (true or false) if initial measurements should be retrieved and ordered for a measuring component in ascending order by date time and create date time. When processing Measuring Component Ordering initial measurements, this should be set to "true".

Copyright © 2026, Oracle and/or its affiliates.

Enabling Direct Measurement Processing

To enable Direct Measurement Processing, do the following:

1. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
2. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.

Enabling Direct Measurement Processing with Machine Learning Anomaly Scoring

To enable Direct Measurement Processing with Machine Learning Anomaly Scoring, do the following:

1. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
2. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.
3. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.
4. Configure an SGG ML Payload Processing Configuration extendable lookup value, and enable the **Use IMD Type** option. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

- Processing Initial Measurement Data Using Direct Measurement Processing

Processing Initial Measurement Data Using Direct Measurement Processing

Initial measurement data (IMD) received from external systems is processed through two major stages:

- **Seeder IMD Seeder Type**: This is the common interface point that receives all IMD information and performs critical validations and conversions. During this process the IMD Seeder identifies the appropriate specific IMD business object for further processing. The choice is made based on the category of IMD being processed (initial load) and the type of measuring component the IMD is for. For example, an IMD for an interval measuring component would result in the Initial Load IMD (Interval) business object being selected. If for some reason the IMD Seeder encounters an error an instance of the Legacy IMD Seederbusiness object associated with the IMD Type will be created to manage the resolution of that error.
- **Specific IMD Type**: Each specific IMD type has the appropriate calculation and validation logic for the type of data that has been received. An IMD containing scalar data will be processed differently than interval data. However, all types of data will have a similar set of steps taken:

Copyright © 2026, Oracle and/or its affiliates.

- Any external system specific statuses for the measurement data are interpreted into the appropriate internal statuses
- An initial consumption calculation is performed
- VEE rules are executed
- Final measurements are created

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

Seeder IMD Type Processing

The Seeder IMD type receives IMD data in a common structure, however, the content of the data (identifiers, date/times, and so on) can vary depending on the system and the particular device that generated it. To accurately translate the external content to values understood by the system the seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into seeder type processing:

- **Identifier Types:** Each IMD must identify the device and measuring component the data has been generated from. The system supports a multitude of identifiers for both the device and measuring component. The service provider configuration specifies which identifier types are used for both the device and the measuring component which enables the IMD seeder to search for the correct destination for the measurement data. For example, a service provider might indicate that devices are identified by a serial number and measuring components are identified by a channel identifier.
- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has many processing methods that are used in the evaluation of the IMD.
  - Initial Measurement Creation - Maps a given measuring component type to the initial measurement BO that should be created
  - UOM Translation - Identifies the extendable lookup that should be used to map the service provider specific UOM to the internal UOM
  - TOU Translation - Identifies the extendable lookup that should be used to map the service provider specific TOU to the internal TOU
  - SQI Translation - Identifies the extendable lookup that should be used to map the service provider specific SQI to the internal SQI
  - Time Zone Translation - Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into seeder type processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time

Copyright © 2026, Oracle and/or its affiliates.

information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming IMD date/times will be stored without an adjustment for DST.

With these inputs the seeder type is able to translate the external identifiers into the appropriate identifiers to tie the initial measurement data to a specific measuring component in the system.

Once a measuring component is identified it is then used to select the appropriate specific IMD type using the "Initial Measurement Creation" processing method of the service provider.

Seeder IMD Type Critical Validations

For initial measurement data to continue on to specific IMD type processing it must pass several critical validations:

- The measuring component must be identified.
- The initial measurement must contain the appropriate date/times for the type of data being processed: scalar must have an end date/time and interval must have both a start and end date/time.
- The interval list must match the number of intervals expected based on the start and end date/times. Note: if there are too few intervals the missing intervals can be automatically added to the initial measurement without generating an error. How this is done depends on whether there are date/times for each individual interval. If there are date/times for each interval the missing intervals are placed where the gaps exist. Otherwise when there are no date/times for the individual intervals the missing intervals are added to the end of the interval list.

If any of the critical validations have failed the IMD processing will be halted and an instance of the IMD Seeder business object will be created, same as when using legacy processing.

Note: There is special logic that exists for handling the situation where either the start date/time or the IMD end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time. Typically there are two 1AMs on that day, one is considered to be during DST and the other is not. To ensure that the appropriate hour is selected the seeder type uses the count of intervals and final measurements that exist prior to the initial measurement to identify the appropriate hour.

> **Note**
>
> Additional detail on the logic to derive the service provider, device, and measuring component identification can be found on the D1-DER-SPRMC Algorithm Type..

Seeder IMD Type Time Zone Translation and Conversion

To provide the most accurate tracking of measurement data all initial measurement data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the seeder type contains functionality to convert incoming date/times from a source time zone into the time zone of the Oracle Utilities application (Meter Data Management or Customer to Meter), (as indicated by the base time zone)

There are three components to identifying the source time zone of the incoming initial measurement:

Copyright © 2026, Oracle and/or its affiliates.

- **Service Provider IMD Import Date/Time Format:** When the initial measurement date/times are sent in format that identifies the time zone for each date/time, the date/times will be delivered to the seeder type already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
  1. The time zone as indicated by the initial measurement data
  2. The service point time zone
  3. The device configuration time zone
  4. The measuring component time zone
  5. The time zone defined on the **Main** tab on the **Installation Options - Framework** portal

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

Note: Additional detail on the logic for date/time adjustments and validation of undercount and overcount can be found on the D1-DODTTMADJ Algorithm Type.

Specific IMD Type Processing

When initial measurement data is received from external systems (head end systems) it is processed based on the the specific IMD type defined or the measuring component type on the "Initial Measurement Creation" processing method. IMD types are configured to execute the VEE rules that are associated to the Initial Load VEE role on the measuring component (or measuring component type in the case of a fallback). For adapter specific variations of the IMD type there is logic to map external measurement statuses to measurement conditions.

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

Scalar Initial Measurement Data Processing

Scalar initial measurement processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Scalar Initial Measurement Data Processing for details.

Interval Initial Measurement Data Processing

Interval initial measurement processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Interval Initial Measurement Data Processing for details.

Copyright © 2026, Oracle and/or its affiliates.

Initial Measurement Data Valid Date Boundaries

Initial measurement data valid date boundaries work in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Initial Measurement Data Valid Date Boundaries for details.

Deferring Initial Measurement Data Processing

Deferring initial measurement data processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Deferring Initial Measurement Data Processing for details.

Initial Measurement Data Prioritization

Initial measurement data prioritization works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Initial Measurement Data Prioritization for details.

Measuring Component Dates

Measuring component dates are maintained in the way in both Legacy Initial Measurement Data Processing and Direct Measurement Processing. See Maintaining Measuring Component Dates for details.

Machine Learning Anomaly Scoring

Machine Learning Anomaly Scoring is a method of processing incoming initial measurement data designed to use Artificial Intelligence and Machine Learning neural network models to recognize "normal" versus anomalous measurement values, and only perform Validation, Editing, and Estimation (VEE) processing for measurements deemed to be anomalous. Most measurement data for most customers passes VEE processing without exceptions, but that processing requires considerable computing resources and processing time. Skipping VEE processing for "normal" data greatly reduces the compute resources and processing time needed for measurement processing.

Notes:

- Machine Learning Anomaly Scoring is available with Software-as-a-Service (SaaS) cloud Implementations only. It should only be used for processing certain types of Initial Load measurements for Residential customers. It can not be used for Estimation or Manual measurements or for Commercial or Industrial customers.
- Machine Learning Anomaly Scoring is available under Limited and Controlled Availability for select Early Adopters.
- Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.

The Machine Learning Anomaly Scoring Process

Machine learning Anomaly Scoring can be used with specific measuring components or measuring component types, and is defied for measuring components or measuring component types using the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

Initial measurements for measuring components flagged for Machine Learning (via the **Enable ML Based Validation** flag) are created as usual, but are marked with an Execution Method of

Copyright © 2026, Oracle and/or its affiliates.

"Machine Learning". These measurements are transitioned to the "Waiting for ML" State, and a copy of the measurement is sent to an Object Storage location, where it is picked up by a deep learning/neural network machine learning model. The model processes the data using the incoming measurement data, sample data, and weather data to calculate estimated values for each measurement value in the payload (see **Machine Learning Models**, below for more details), and then returns the updated data to a different Object Storage location. A **Machine Learning** payload processing batch process picks up the payload and updates the initial measurement records with the machine learning measurement values as well as the **Anomaly Score Threshold** specified on either the measuring component, measuring component type, or the MDM Master Configuration. It then calculates the difference between the original Pre-VEE value and the machine learning value for each measurement value in the payload (this is known as an "Anomaly Score") and compares each to the **Anomaly Score Threshold**. If any of the Anomaly Scores for any of the measurement values exceeds the **Anomaly Score Threshold**, the initial measurement is deemed "anomalous", an exception is created, and the measurement proceeds through the normal initial measurement data lifecycle, including VEE processing and creation of final measurements. If none of the Anomaly Scores for any of the measurement values exceeds the Anomaly Score Threshold, the initial measurement moves directly to the creation of final measurements unless Post-ML VEE processing has been configured for the measuring component or measuring component type (see **Post-ML VEE Processing** below).

> **Note**
>
> **Note**: The Anomaly Score Threshold for a measuring component is selected using the following order of precedence: (1) Measuring Component threshold (if present), (2) Measuring Component Type threshold (if present), (3) MDM Master Configuration.

Anomaly Threshold Considerations

The Anomaly Threshold can be set at three levels:

- Default (via Master Configuration): This is used to set a default threshold for all measuring components that don't have a threshold set for their type or themselves.
- Measuring Component Type: This is used to set thresholds for groups of measuring components that share a type.
- Measuring Component: This is used to set the threshold for each specific measuring component. This is a good approach to set a threshold for any unique measuring component that is idiosyncratic and is either failing with too many false positives or passing with too many false negatives.

There are several consideration to take into account when setting the Anomaly Threshold at each of these levels.

- **Interval vs. Scalar**: Because of the considerable difference between interval values and scalar values, using the same Anomaly Threshold for both interval and scalar measuring components can result in many "false positive" results. Consider a threshold of 10. That would be a reasonable threshold for a single interval but not for a scalar measurement that covers the entire day. To account for this, you might set the default threshold (via the Master Configuration) as the threshold to use for all scalar measuring components, and use measuring component type or measuring component thresholds for interval measurements.
- **Interval Size**: Similar to the difference between interval and scalar values, measurement values for 15-minute intervals are generally smaller than those for 1-hour intervals. As above, using the same threshold for all interval measuring components could result in

Copyright © 2026, Oracle and/or its affiliates.

"false positives". To account for this you might define thresholds based the Interval Size defined for different measuring component types. For example, 15-minute measuring component types might have a threshold of "4", while 1-hour measuring component types might have a threshold in the 13-16 range.

- **Service Type**: Interval and scalar values are likely to differ between commodities and service types. For example, if your system records daily scalar measurements for both gas and electricity, the scalar values for those are likely to be quite different. A household that uses 29 to 30 kilowatt-hours (kWh) of electricity per day might use only 1 to 4 therms of natural gas per day. To account for this, you might use specific measuring component type or measuring component thresholds for gas measuring components and different measuring component type or measuring component thresholds for electric measuring components.

Machine Learning Models

The neural network machine learning model used with machine learning anomaly scoring is trained on millions of data points to recognize "normal" versus anomalous usage patterns. When the model runs on your customers' data, it processes the incoming measurement data along with weather data to calculate estimated values for each measurement value in the payload.

Machine Learning Anomaly Scoring includes 3 machine learning models used with specific types of data:

- Residential 1 hour interval data
- Residential 15 minute interval data
- Residential daily scalar data

Post-ML VEE Processing

If needed, you can define VEE Groups that are executed after machine learning processing to compare or adjust the initial measurement based on a related measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the **Post-ML** VEE Role on measuring component types and measuring components. Note that Post-ML VEE Groups can only be used when machine learning is enabled for the measuring component or measuring component type via the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

Post-ML VEE Groups can include only the following VEE Rules:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement

See Configure Post-ML VEE Processing in Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

Enabling Machine Learning Anomaly Scoring

Enabling Machine Learning Anomaly Scoring involves the following:

1. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.

Copyright © 2026, Oracle and/or its affiliates.

2. Set the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** on specific measuring components or measuring component types (respectively) if needed.
3. Configure an SGG ML Payload Processing Configuration extendable lookup value. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

> **Note**
>
> Machine Learning Anomaly Scoring is available under Limited and Controlled Availability for select Early Adopters.
> Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.

Enabling Machine Learning Anomaly Scoring with Direct Measurement Processing

To enable Machine Learning Anomaly Scoring with Direct Measurement Processing, do the following:

1. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.
2. Set the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** on specific measuring components or measuring component types (respectively) if needed.
3. Configure an SGG ML Payload Processing Configuration extendable lookup value, and enable the **Use IMD Type** option. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.
4. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
5. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.

CSV Upload

Measurement data for a measuring component can be uploaded via comma separated values (CSV) files. See Uploading Measurement Data for more information.

Importing Data From External Systems

In addition to importing data from head end systems for meters and other devices, the system can also import data from other types of external systems such as weather systems or from market participants such as an independent system operator (ISO) as part of a settlement market. Data received from these types of systems is stored in standalone measuring components (see About Measuring Components for more information).

Copyright © 2026, Oracle and/or its affiliates.

When importing data for standalone measuring components, initial measurement processing uses the **Standalone Identifier** (or **Standalone ID**) to identify the measuring component for which the measurement data belongs.

Each external system from which data is imported should be defined as an external application. If the data being imported uses units of measure or time of use periods that are specific to the external system, translation methods should be defined to map the specific data to standard codes.

Using the IMD Control Staging Portal

You use the **IMD Control Staging** portal to view and maintain records in the IMD Control Staging table. You would typically access records using this portal in the following situations:

- With Direct Measurement Processing when a measuring component has been configured for Measuring Component Ordering and an initial measurement is in the "Ready to Process" state (see Direct Measurement Processing and Measuring Component Ordering for more information)

You can access the portal from either the **Initial Measurement Data History** zone of the **Measuring Component** portal, or the **Initial Measurement Data Search** zone in the **IMD Query** portal.

When you select a record in one of those zones, you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **IMD Control Staging Details**: displays basic attributes of the initial measurement.
  - To process an initial measurement from this portal, click **Process** in the **Record Actions** section.
    - Initial measurements in the the "Ready to Process" **Status** are processed using Direct Measurement Processing.
- **IMD Stage Payload**: displays the initial measurement's payload

Final Measurements

- About Final Measurements

About Final Measurements

Final measurements are measurements that have been validated, and if necessary, edited & estimated, and is ready for use in downstream processing such as bill determinants calculations.

When an initial measurement is considered "final," that is, it has passed all VEE processing and no additional modifications or changes need to be made, it is transformed into a Final Measurement, or simply a Measurement (the terms measurement, final measurement, and final consumption all reference this same "final" measurement data). Only one final measurement can exist for a given date/time for a given measuring component; one final measurement exists per interval, and likewise one final measurement exists for each scalar reading. In both cases, the final measurement value stored represents the amount consumed between its date/time and the prior final measurement's date/time.

Copyright © 2026, Oracle and/or its affiliates.

When creating final measurements from initial measurement data:

- Final measurements are created using Post VEE quantities
- Each final measurement's condition is copied from the Post VEE condition
- Initial measurements are normalized into final measurements where each final measurement is for a specific date and time.
- Because a single initial measurement may contain many "readings," a separate final measurement is created for each interval in the initial measurement. For example, if an initial measurement contains 24 hours of 15 minute readings, 96 measurements will be created, each with a specific date and time.
- A Combined Multiplier (created from multipliers defined for the measurement's measuring component type and install event) is stored for each measurement. In addition, other values are stored for measurements of specific types. The Reading Value (which often differs from the measurement value) is stored for scalar and subtractive interval measurements, and the Reading Condition is stored for subtractive interval measurements. These values are used with measurement reprocessing and other recalculation processes in the event that the original initial measurements have been archived or deleted.

The measurement date/time of a measurement represents the end of the period the measurement value covers. How the start of the period is defined varies slightly based on the type of measuring component associated to the measurement:

- **Scalar Measuring Components**: the measurement date/time represents the point at which the reading was taken. The period the measurement covers is defined as the previous measurements date/time through the measurement date/time.
- **Interval Measuring Components**: the measurement date/time represents the end of the interval period. No date/time is stored to represent the beginning of the interval period, rather it is implied by the interval size of the measuring component. For example an interval measuring component with an hourly interval size and a measurement date/time of 01/01/2010 12:00AM represents the measured quantity for the period of 12/31/2009 11:00PM through 01/01/2010 12:00AM.
- For scalar measuring components there exist special circumstances where a final measurement is not eligible for use in processing (e.g. consumption calculation, usage calculation, etc.). To prevent other processes from using these measurements they are marked with a usage flag of "Do Not Use." This is typically the result of the automatic correction of a rollover threshold error when lower quality measurements inaccurate measurements are removed to enable successful calculation of subtractive consumption.

Conditions

Conditions are used to describe the quality of a final measurement and is an important input into many of our processes. During initial measurement data processing these conditions help VEE rules identify the initial measurements that require more advanced validation or estimation. During usage calculations measurement conditions help to inform the billing system whether the billing determinants produced are based on actual data read from a meter or if it has been estimated (or in many cases a combination of the two).

Conditions follow a numeric sequencing that begins at 000000 and goes up to 999999. As the condition code rises numerically so does its perceived quality. Many areas the system take this into account:

- VEE rules use conditions to identify which initial measurement values need to be estimated or adjusted

Copyright © 2026, Oracle and/or its affiliates.

- Rollover error automatic corrections allow higher quality data to bypass lower quality data in order to achieve a consumption calculation that does not produce an error.
- Usage calculation rules can provide information about the quality of billing determinants based on the underlying measurement data that feeds into those service quantities.

Throughout the system functionality uses condition code ranges to make decisions. In general the ranges are considered to be:

Condition Range Start	Condition Range End	Description of Range
000000	199999	No measurement
200000	299999	Missing Measurement
300000	499999	Estimated measurement
500000	999999	Regular measurement

The ranges of "No Measurement" and "Missing Measurement" at first glance sound synonymous but there is an important distinction. With conditions that fall within the "No Measurement" range the expectation is that no measurement should exist. Whereas, the "Missing Measurement" range implies that a measurement was expected but never received.

Note: in some circumstances, like rollover threshold errors, the regular range is further divided to include a "Superior" range of 9000000 - 999999 which can be used to select the appropriate reading to be used to resolve the error.

During initial measurement processing the Final Measurement Replacement VEE rule can be used to increase the perceived quality of a condition code. For example, the "No Read - Outage" condition which falls in the "No Measurement" range can be considered higher quality than "Estimated Measurements" for the purposes of measurement replacement.

Refer to About IMD Validations for more information on the Final Measurement Replacement VEE rule.

Derived Values

Final measurements can record up to 10 derived values in addition to the "as measured" value. The derivation formula for each value on a final measurement is controlled by the measuring component type.

Derived values are not reliant on consumption values, but can also come from factors, historical data, or another source. For example, a set of measurements can be adjusted or converted into other units of measure. In the below example the first derived value is the result of the application of a loss adjustment factor and the second derived value is the result of applying a pre set formula against the loss adjusted value to compute BTU.

Date/Time	As Measured UOM: CCF SQI: N/A	Loss Adjusted UOM: CCF SQI: N/A	Thermal Units UOM: BTU SQI: N/A
01/01/2010 3:00PM	10	10.1	10.11
01/01/2010 4:00PM	15	15.15	15.165
01/01/2010 5:00PM	10	10.1	10.11

Refer to Understanding Measuring Component Types in the Administration Guide for additional information about configuring derivation algorithms.

Copyright © 2026, Oracle and/or its affiliates.

Final Measurements and Direct Measurement Processing

When using Direct Measurement Processing, final measurement may display additional information if any quantities were changed during VEE processing. In this case, original (Pre VEE) quantities and data are captured along with the final (Post VEE) quantities in the following values:

- Pre-VEE Measurement: The Pre-VEE value for the measurement
- Pre-VEE Measurement Condition: The Pre-VEE condition code or the measurement
- Pre-VEE Reading: The Pre-VEE value for reading (used with scalar measurements)
- Pre-VEE Reading Condition: The Pre-VEE condition code for reading (used with scalar measurements)

Interval Snapshot Mapping

- About Interval Snapshot Mapping
- Maintaining Interval Snapshot Mapping

About Interval Snapshot Mapping

Interval snapshot mappings are used to map intervals to a particular point in the day, and are used in the snapshot creation process to align a particular interval with the column it belongs to.

The Interval Snapshot Mapping (D1-IntervalSnapMapping) business service is used to create Interval Snapshot Mapping entries based on the installation time zone. This service takes Seconds Per Interval, Start Date/Time, and End Date/Time as inputs.

Maintaining Interval Snapshot Mapping

You use the **Interval Snapshot Mapping** portal to view and maintain Interval Snapshot Mapping entries.

Refer to About Interval Snapshot Mapping for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Interval Snapshot Mapping**. You are brought to a query portal with options for searching. You can search by:

- Seconds Per Interval
- Date/Time From
- Date/Time To

The search results include all of the mapping entries that match the provided Seconds Per Interval (if provided) and that fall within the specified date range (if provided).

Creating Interval Snapshot Mapping Entries

The Interval Snapshot Mapping (D1-IntervalSnapMapping) business service is used to create Interval Snapshot Mapping entries based on the installation time zone. This service can be invoked from the **Interval Snapshot Mapping** portal.

Copyright © 2026, Oracle and/or its affiliates.

To create new Interval Snapshot Mapping entries, either select **Main**, then **Settlement**, then **Interval Snapshot Mapping**, and then Add, or click **Add** in the **Interval Snapshot Mapping** portal. The **Create Interval Snapshot Mapping Entries** page opens.

Enter parameters to specify the mapping entries to create, including:

- Seconds Per Interval
- Date/Time From
- Date/Time To

Click **OK**.

Enter appropriate parameters in the search portal to view the new entries.

Reader Remarks

Initial measurements for manually-read scalar meters can create reader remarks. Reader remarks are a specific type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters.

Reader remark types are submitted with scalar initial measurements when received from a head end system or meter read collection system. Reader remarks are NOT uploaded along with other device events. Reader remarks are always associated with a scalar initial measurement.

Reader remarks can be passed onto to other subscribing systems such as a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.

Like other device events, reader remarks can create service issue monitors.

> **Note**
>
> Refer to About Service Issue Monitors for more information on Service Issue Monitors.

Creating Reader Remarks Manually

Use this procedure to create a new reader remark for an existing initial measurement.

Reader remarks must always be associated with an initial measurement.

Reader remarks are most often imported into the application with an initial measurement. However, users can also create reader remarks using the following procedure.

1. Search for and select the initial measurement that the new reader remark is to be associated with.
2. Click the **Add** link on the **Reader Remarks of Initial Measurement** zone on the **Initial Measurement** portal.
3. Select the reader remark type for the reader remark from the **Reader Remark Type** drop-down list.
4. *Optional*. If you need to create more than one reader remark for the initial measurement, click the plus sign, and select the reader remark type for the reader remark from the Reader Remark Type drop-down list. Repeat as needed.

Copyright © 2026, Oracle and/or its affiliates.

Note that only a single reader remark of any given type can be associated with a single initial measurement.

5. Click **OK**. The new reader remark is displayed in the **Reader Remarks of Initial Measurement** zone on the **Initial Measurement** portal.

Periodic Estimation

- About Periodic Estimation

About Periodic Estimation

Periodic estimation is a process that scans eligible devices and their measuring components for missing measurements that have not been received within a required amount of time. Once missing measurements are identified it then generates estimation initial measurements to fill in those missing measurements. It is important to note that the periodic estimation itself does not actually perform any estimations. It relies on the estimation initial measurement to generate the estimate through the estimation VEE rules that are executed.

The periodic estimation process is as follows:

1. Determine if measuring component is eligible for estimation.
2. Scan the eligible period for missing measurements.
3. Confirm missing measurements do not fall within a period where the device was not installed or was disconnected.
    - For scalar measuring components, initial measurements are NOT created for disconnected periods.
    - For interval measuring components, initial measurements will be created for disconnected periods, but the intervals will be marked as 'No Read - Disconnected' (Measurement Condition code 102000) and will be ignored for estimation by VEE processing.
4. Confirm there are no in progress initial measurements that cover the period of the missing measurements.
5. Either generate an initial measurement or a To Do for each measurement gap that is found.

This process relies on an important date that is maintained on each measuring component:

- **Latest Contiguous Measurement Date/Time**: this represents the last time the measuring component had a complete history of measurements without any gaps. By nature it implies that the is a gap immediately after it.

This date allows the periodic estimation process to efficiently identify those measuring components that are missing measurements and need to be estimated.

> **Note**
>
> It is possible to disable periodic estimation for a given device by updating the periodic estimation eligibility on the service point it is associated to. This is useful if there are one-off service points that should either never be eligible for estimation or should temporarily have estimation turned off.

Copyright © 2026, Oracle and/or its affiliates.

Periodic Estimation Feature Configuration

The estimation begin date for interval and scalar periodic estimation can be influenced by a feature configuration option under the feature type Measurement Data Options (D1MD).

When a **No of Hours in Past to Retrieve Last Useable Measurement** option is defined it will have two impacts on periodic estimation, both of which are only applicable when the measuring component's type does not have a value configured for the *Maximum Days to Estimate*.

1. When periodic estimation is run for a measuring component that does not yet have a **Last Contiguous Measurement Date Time** then this option will be used to compute the beginning of the search range to find a useable measurement. That date is then used to begin estimation from.
2. When running for a measuring component where the **Last Contiguous Measurement Date Time** is already set then this option will be used to compute the oldest date that estimation can begin from.

For both scenarios the begin date time calculation will be performed by subtracting the value configured from the current Process Date Time. For example, if this option were set to 744 hours (31 days) and the process date time were 02/01 12:00:00AM then the begin date time would be calculated as 01/01 12:00:00AM.

When implemented this option will also instruct the Update Latest Date/Time on Scalar Measuring Component algorithm type (D1-UPDDTSCMC), which is responsible for maintaining the latest measurement date time information for scalar measuring components, to skip initializing the latest measurement date/time information when an initial measurement is received for a measuring component where there is existing measurement data history and this feature configuration option is configured. For those scenarios the initialization of the latest measurement date/time information should be deferred to the Auto-Read Scalar Period Estimation process. Note: this only applies for situations when the latest measurement date/time information has not been set previously and the initial measurement is not the first measurement received for the measuring component.

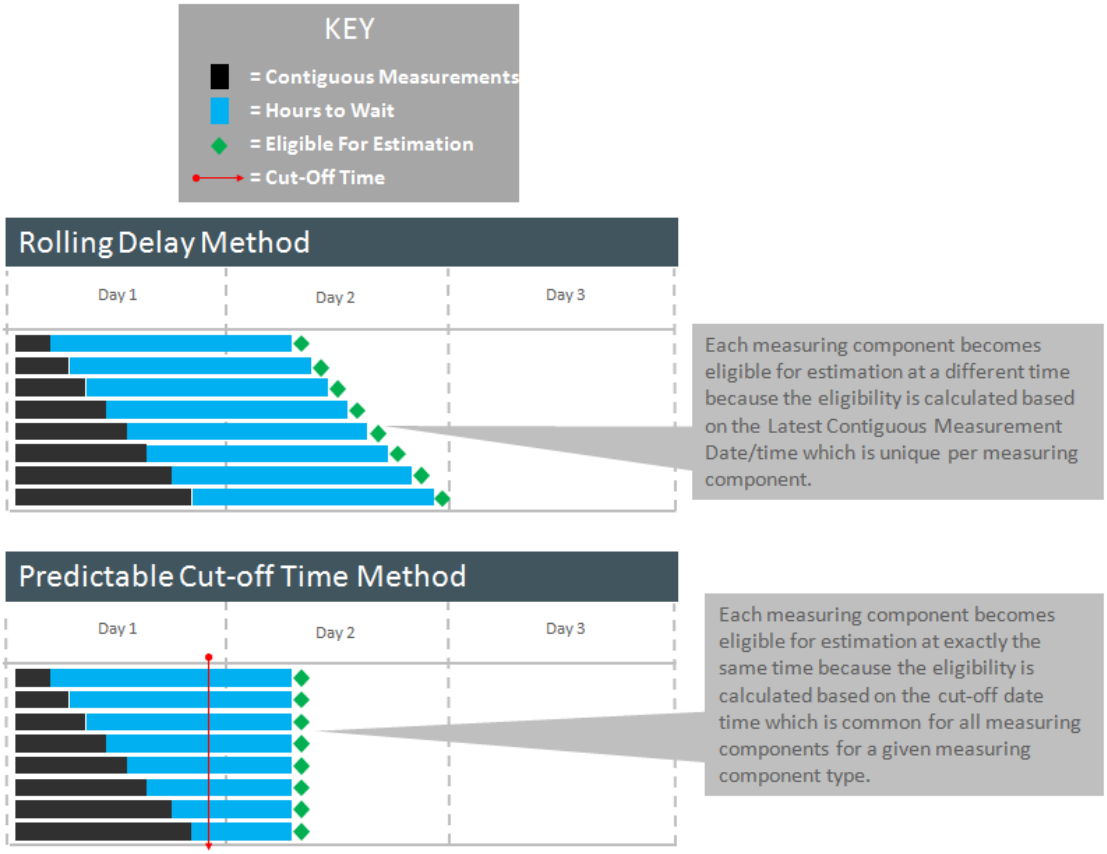
> **Note**
>
> This does not apply to the analogous logic for an interval initial measurement.

Interval Periodic Estimation

When missing measurements have been identified by the interval periodic estimation functionality an estimation initial measurement will be created for each contiguous set of missing measurements in the estimation period.

There are two methodologies available for estimating interval measuring components:

Copyright © 2026, Oracle and/or its affiliates.



- **Rolling Delay**: In this method the execution time of the periodic estimation process is an important factor in which measuring components will estimate how far each measuring component estimate. This method supports a minimum number of hours that will be estimated (as defined by the "Hours to Estimate" on the measuring component type).
- **Predictable Cut-off Time:** In this method all measuring components for a measuring component type will estimate until a specified point in time (as defined by the "Cut-off Time" on the measuring component type). With this method the cut-off time determines how far estimations are made rather than the execution time of the periodic estimation process. There is no minimum estimation size for this method it will always estimate up until the cut-off date/time even if that is only a single interval.

The cut-off time takes into consideration daylight savings time (DST) when the device is configured as sending data in legal time (meaning the Incoming Data Shift has been set to Always in Local). In these circumstances the cut-off time will be considered to be adjusted for DST. This is important because measurement data is stored in standard. For example:

Cut-off Time: 12:00AM	Measurement Time in Standard
01/15/2010 12:00AM (DST not in effect)	01/15/2010 12:00AM
04/15/2010 12:00AM (DST in effect)	04/14/2010 11:00PM

Furthermore, the cut-off time defined on the measuring component type does not have a time zone associated to it and as such it is considered to be defined in the time zone of the particular measuring component being processed. For example, a cut-off time of 12:00AM would be considered to be 12:00AM UTC -08:00 and 12:00AM UTC -05:00 based on the time zones of the respective measuring components. This allows a single measuring component to apply to many time zones and behave uniformly.

Copyright © 2026, Oracle and/or its affiliates.

Algorithm Type – Create Interval IMD and To Do Based Upon Install History (D1-CIITBIH)

Algorithms based on this algorithm type determine if a measuring component is missing final measurements during the estimation horizon for any period where the associated device configuration was installed. If final measurements are missing it will either create an estimate initial measurement, create a To Do Entry, or both.

There are a few scenarios that will result in this algorithm being used:

1. If the measuring component's service point has an **Estimation Eligibility** of Not Eligible
2. If the measuring component's service point has open Estimation Suppression activities that overlap the System Date/Time and the MDM Master Configuration field **Fill Missing Data with Zero** is not set to "Yes"
3. If the measuring component's **Latest Contiguous Measurement Date Time** is greater than the process date time less the measuring component type's **Number of Hours to Wait**

The estimation horizon begin date is determined by first determining the initial horizon start date time:

- If the Measuring Component Type's **Maximum Days to Estimate** is configured it will be the more recent of the following two dates:
  - The process date time less the **Maximum Days to Estimate**
  - The Measuring Component's **Latest Contiguous Measurement Date Time**
- Otherwise, if the Measurement Data Option feature configuration type is configured with a value set for the **No of Hours in Past to Retrieve Last Useable Measurement** option then the start will be calculated as follows:
  - If the **Latest Contiguous Measurement Date Time** is populated then take the more recent of:
    - The process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
    - The Measuring Component's **Latest Contiguous Measurement Date Time**
  - Otherwise, if the **Latest Contiguous Measurement Date Time** is not populated then:
    - Search for a measurement between the following two dates:
      - Start: the process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
      - End: the process date time
    - If a measurement is found that date is used. Otherwise the process date time less the **No of Hours in Past to Retrieve Last useable Measurement** is used.
- Otherwise, the Measuring Component's **Latest Contiguous Measurement Date Time** is used.
- Otherwise, if the device is installed it will be set to the **Installation Date Time** of the Installation Event (IE).

The estimation horizon end date will be calculated based on the measuring component type's **Periodic Estimation Execution Method**.

Rolling Delay:

The more recent of the following dates will be used:

Copyright © 2026, Oracle and/or its affiliates.

- The **Latest Measurement Date Time** plus the Measuring Component Type (MC Type) - **Number of Hours to Estimate** defined.
- The **Process Date Time** less the MC Type - **Number of Hours to Wait**
- Estimations are limited by either a device configuration change or removal. Therefore, the date determined above will be limited to the earlier of the device configuration **Effective End Date Time** or the **Removal Date Time** on Install Event.

Predictable Cut-off:

In predictable cut-off the intent is to only estimate up to the last cut-off time that has passed based on the **Number of Hours to Wait** from the measuring component type and no further.

First the process date time will be adjusted based on the **Number of Hours to Wait** configured on the measuring component type. For example, if the process date/time is 01/15/2010 06:00PM and the **Number of Hours to Wait** is 48 then the adjusted process date/time is: 01/13/2010 06:00PM.

Next, the last cut-off time to have passed will be identified and that will become the estimation end date time. In our above example this would be 01/13/2010 12:00AM.

Once the estimation horizon begin and end date times have been identified they may be further split into smaller periods based upon the Measuring Component's device configuration installation history.

For Example:

- Estimation horizon is April 10th at 12AM through April 20th at 12AM
- The Measuring Component's device configuration was installed at one service point on April 1st at 12AM and removed on April 15th at 12AM
- The Measuring Component's device configuration was then installed at a new service point on April 18th at 12AM
- Given the above information the estimation horizon would be broken up into the following two periods:
  - April 10th 12AM through April 15th 12AM (this is for the first installation)
  - April 18th 12AM through April 20th 12AM (this is for the second installation)

Each of the estimation horizon periods determined based upon the installation history will be further examined to determine contiguous periods where final measurements are missing.

For Example:

- Processing April 10th through April 15th
- One final measurement exists for April 12th for 8AM through 9AM
- Yields the following sub-periods:
  - April 10th 12AM through April 12th 8AM
  - April 12th 9AM through April 15th 12AM
- -For every missing period found,
  - Find Initial Measurements that are in a non Final state and the initial measurement's period has an overlap with the missing period.
    - For each initial measurement found,

Copyright © 2026, Oracle and/or its affiliates.

* If the initial measurement's period falls exactly within the boundaries of the missing period, skip the creation of an estimate initial measurement for the missing period.
* If the initial measurement's end date/time is before the start of the missing period, the initial measurement's end date/time is now the start of the missing period. Proceed to create an estimate initial measurement for the adjusted missing period.
* If the initial measurement's start date/time is after the start of the missing period, the initial measurement's start date/time is now the end of the missing period. Proceed to create an estimate initial measurement for the adjusted missing period.

Each time Periodic Estimation is executed the **Latest Contiguous Measurement** will be validated and potentially updated. For example, if it is before the install event's **Installation Date Time**, it will be updated to be the install event's **Installation Date Time**.

> **Note**
>
> If the **Latest Contiguous Measurement Date Time** falls within the installation window of a prior installation it will not be adjusted to the current install event's **Installation Date Time**.

Scalar Periodic Estimation

Scalar periodic estimation is only available for automatically read scalar measuring components since it requires that scalar measuring components be read on at least a daily basis to be eligible.

The scalar estimation process relies heaving on details from the measuring component type to identify when a measurement is missing. This is because scalar measurements are not required to fall on specific boundaries like interval data. For example, a scalar measuring component read on a daily basis around midnight could have measurements at 12:00AM, 11:55PM, 12:01PM or even at various other points during the day if a field reading or on demand read is performed.

Auto-read scalar measuring component types are configured to provide information that will:

- Identify at what point or points during the day the measuring component will be read. This is a combination of the Hours Between Measurements and the First Daily Measurement Time on the measuring component type.
- Identify the allowable tolerance for measurements that do not fall on the specific expected measurement date/times (e.g. 11:55PM and 12:05AM are OK but 9AM is not). This is defined by the Early and Late Measurement Threshold on the measuring component type.

This allows the scalar periodic estimation process to identify the points during the day that measurements are expected and if no measurement are found it will generate an estimation initial measurement.

Unlike interval periodic estimation each missing measurement will be created as a separate estimation initial measurement.

The Early Measurement Threshold and Late Measurement Threshold fields support situations where a head end system is not be able to report their readings at a consistent time each day. For example a head end system might send reads as follows:

- 3/1 11:55PM

Copyright © 2026, Oracle and/or its affiliates.

- 3/3 12:01AM
- 3/4 12:00AM
- 3/5 12:02AM
- 3/6 11:59 PM

These fields create a window prior to and after an expected reading where anything received within that window would satisfy the expected reading. For example:

- For a measuring component with an Early Measurement Threshold of 00:10:00 (10 minutes), a reading for 3/1 11:55PM would satisfy the early threshold for the 3/2 12:00AM expected reading.
- For a measuring component with an Late Measurement Threshold of 00:10:00 (10 minutes), a reading for 3/3 12:01AM would satisfy the late threshold for the 3/3 12:00AM expected reading.

To determine if a measuring component is eligible for estimation the process will evaluate if the Last Contiguous Measurement Date/Time far enough in the past. Specifically the number of hours elapsed since the Last Contiguous Measurement Date/Time must be greater than or equal to the Hours Before Estimation from the measuring component type.

> **Note**
>
> The first daily measurement time takes into consideration daylight savings time (DST) when the device is configured as sending data in legal time (meaning the Incoming Data Shift has been set to Always in Local). In these circumstances the first daily measurement time will be considered to be adjusted for DST. This is important because measurement data is stored in standard. For example:

First Daily Measurement Time: 12:00AM	Measurement Time in Standard
01/15/2010 12:00AM (DST not in effect)	01/15/2010 12:00AM
04/15/2010 12:00AM (DST in effect)	04/14/2010 11:00PM

Furthermore, the first daily measurement time defined on the measuring component type does not have a time zone associated to it and as such it is considered to be defined in the time zone of the particular measuring component being processed. For example, a first daily measurement time of 12:00AM would be considered to be 12:00AM UTC -08:00 and 12:00AM UTC -05:00 based on the time zones of the respective measuring components. This allows a single measuring component to apply to many time zones and behave uniformly.

Algorithm Type – Auto-Read Scalar Periodic Estimation (D1-ARSPE)

Algorithm based on this algorithm type determine if a measuring component is missing final measurements during the estimation horizon for any period where the associated device configuration was installed. If final measurements are missing it will either create an estimate initial measurement, create a To Do Entry, or both.

There are a few scenarios that will result in this algorithm being used:

1. If the measuring component's service point has an **Estimation Eligibility of Not Eligible**
2. If the measuring component's service point has open Estimation Suppression activities that overlap the System Date/Time and the MDM Master Configuration field **Fill Missing Data with Zero** is not set to "Yes"

Copyright © 2026, Oracle and/or its affiliates.

3. If the measuring component's **Latest Contiguous Measurement Date Time** is greater than the process date time less the measuring component type's **Number of Hours to Wait**

The estimation horizon begin date is determined by first determining the initial horizon start date time:

- If the Measuring Component Type's **Maximum Days to Estimate** is configured it will be the more recent of the following two dates:
  - The process date time less the **Maximum Days to Estimate**
  - The Measuring Component's **Latest Contiguous Measurement Date Time**
- Otherwise, if the Measurement Data Option feature configuration type is configured with a value set for the **No of Hours in Past to Retrieve Last Useable Measurement** option then the start will be calculated as follows:
  - If the **Latest Contiguous Measurement Date Time** is populated then take the more recent of:
    - The process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
    - The Measuring Component's **Latest Contiguous Measurement Date Time**
  - Otherwise, if the **Latest Contiguous Measurement Date Time** is not populated then:
    - Search for a measurement between the following two dates:
      - Start: the process date time less the **No of Hours in Past to Retrieve Last useable Measurement**
      - End: the process date time
    - If a measurement is found that date is used. Otherwise the process date time less the **No of Hours in Past to Retrieve Last Useable Measurement** is used.
- Otherwise, the Measuring Component's **Latest Contiguous Measurement Date Time** is used.
- Otherwise, if the device is installed it will be set to the **Installation Date Time** of the Installation Event (IE).

The estimation horizon end date will be whichever of the following dates that is furthest in the past:

- Process Date/Time - **Hours Before Estimation** + **Number of Hours to Estimate**
- The device configuration's effective end date/time
- The Off date/time of the device configuration's installation event.

Second, the algorithm identifies any expected measurement date/time without a final measurement or an in progress initial measurement. For each missing measurement an Estimate initial measurement will be created. Estimations will only be made for periods where the device is installed at a service point. If the **Evaluate Installation Event Off Period** parameter has been set to 'N' or left blank then the device must also be turned on.

To determine if there are any missing final measurements during the estimation horizon the algorithm uses the concept of expected measurement date/times as explained above. If there are any expected measurements not covered by a final measurements as dictated by the early and late thresholds then an initial measurement will be generated.

Copyright © 2026, Oracle and/or its affiliates.

Measurement Reprocessing

- About Measurement Reprocessing
- Reviewing Measurements for Reprocessing

About Measurement Reprocessing

One major problem utilities face today is when devices are installed with an incorrect consumption multiplier. When this occurs, the billing consumption and the bills are incorrect retroactively.

The Measurement Reprocessing module automatically corrects historical consumption when a change is detected for either a Measuring Component multiplier (the **Register Multiplier** for register measuring components or **Channel Multiplier** for interval and auto-read register measuring components) or an Install Event **Installation Constant** or **Installation Date/Time**. The system monitors for these changes and automatically reprocesses historical measurements if one of the values changes. When a Measuring Component has been triggered for Measurement Reprocessing, a new message will be recorded on the **Log** tab for the Measuring Component. The message will look similar to this example: "Measuring Component Multiplier updated from 1.000000 to 2.000000. Measurements will be examined for reprocessing". Also, a link to the activity created to perform the reprocessing will be shown in the Related Object column.

Two different Activities are leveraged to perform this reprocessing:

- **Measurement Reprocess Activity - Interval**: this activity reprocesses interval data by dividing existing measurements by the Combined Multiplier stored on each measurement record and creates new System IMDs.
- **Measurement Reprocess Activity - Scalar**: this activity reprocesses scalar data by creating new System IMDs from the Reading value stored on each measurement.

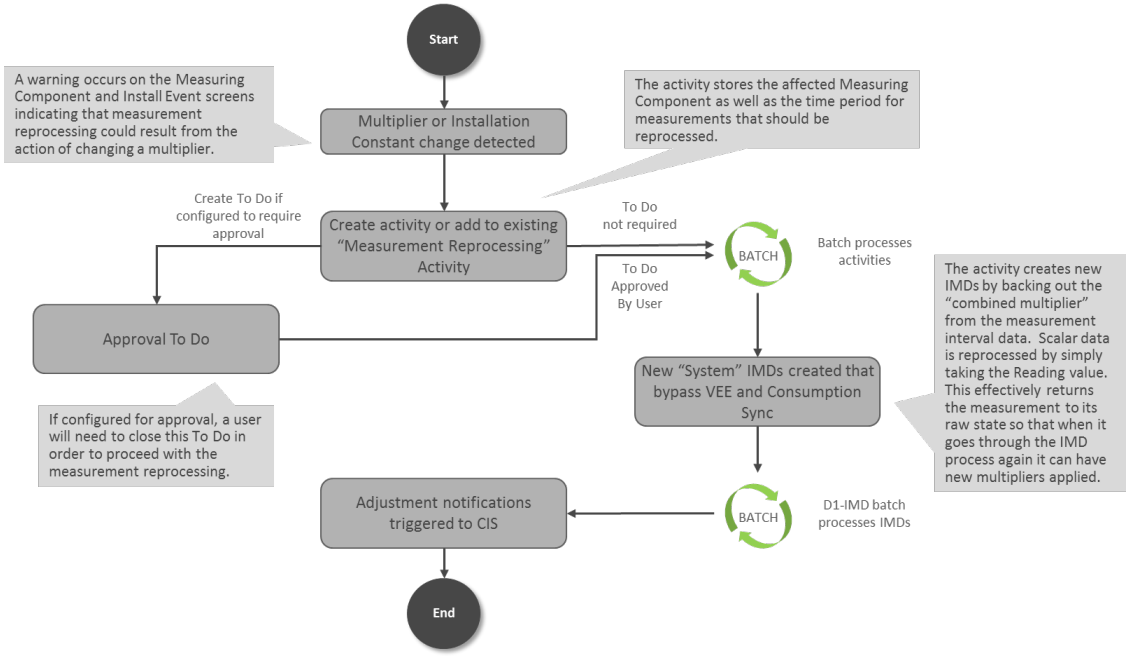
The new System IMDs that are generated will skip both VEE and Consumption Sync but will apply the new multiplier and installation constant. Once the activity is finished, a list of the spawned IMDs will be displayed on the Activity screen.

This process works in tandem with the Adjustment Notification process to trigger billing adjustment notifications to the CIS. If usage transactions have already been fulfilled once measurements are reprocessed, then each affected period will be sent to CIS to indicate that an adjustment is required.

If desired, an approval step can be set as required on the related Activity Type. If set, a To Do will be generated for review. The review steps for this To Do are covered in the Reviewing Measurements for Reprocessing section.

The diagram below helps to provide an overview of Measurement Reprocessing:

Copyright © 2026, Oracle and/or its affiliates.



For information on configuration, reference the related section for Configuring Measurement Reprocessing in the *Administrative User Guide*.

Reviewing Measurements for Reprocessing

When the Measurement Reprocessing Activity Types are configured as "Yes" in the "Approval Required?" field, then a To Do will be generated for user review. The To Do Type for this is set on the Activity Type as well (the configuration for this is covered in detail in the section for Configuring Measurement Reprocessing in the *Administrative User Guide*).

This To Do provides an approval step for a user to examine the Measurement Reprocessing Activity and perform the following:

1. Determine if the multiplier or date change appears legitimate and accurate. Information on the type of change that occurred can be found on the **Log** tab for the Activity. If the change should be rejected, this can be performed using the **Discard** button.
2. Ensure that the period being examined for reprocessing (shown as the Activity Start Date and End Date) covers the period expected. If not accurate, the dates can be changed using the **Edit** button.

If all information on the Activity is correct, then press the **Approve** button to allow reprocessing to continue. Once the activity is finished, a list of the spawned IMDs will be displayed on the Activity screen.

Refer to About Measurement Reprocessing for additional functional information.

Consumption Sync

- Introduction to Consumption Sync

Copyright © 2026, Oracle and/or its affiliates.

Introduction to Consumption Sync

Traditional interval and smart meters are often configured with two channels of consumption data, one interval and the other scalar. Utilities configure their meters this way for several reasons, including:

- Two channels provide for a redundant reading in case of a failure.
- Two channels provide for an independent check reading to ensure the meter is working properly.
- Customers often need both types of data for their business processes.

When capturing and storing usage for smart meters with related measuring components, one potential challenge is keeping measurements from related scalar and interval channels synchronized. Measurements for interval and scalar channels for the same device may fall out of alignment for many different reasons, such as:

1. Some meters use different technologies for recording interval and register readings, so the physical readings can be different
2. Readings can vary because of different measurement times
3. Estimation processes can vary by channel and produce different results
4. There can be a mix of actual and estimated measurements

Having two sources of consumption data can cause many problems for utilities. Unfortunately, utilities often need both the scalar and interval data to satisfy their business requirements. For example, scalar register readings may be used for billing, while the interval data is used on a customer self-service portal.

Keeping measurement data aligned between two related measuring components is achieved in several ways:

- During the initial load of initial measurement data VEE rules estimate for missing measurements using measurement data from related measuring components when available. Please refer to About IMD Estimations for more information.
- When measurement data is missing for one or both of the related measuring components the periodic estimation process will generate estimation initial measurement data that will estimate one measuring component then estimate the second measuring component such that the second measuring component's data can be synchronized with the first. Please refer to About Periodic Estimation for more information.
- When new, more accurate, measurement data is received for only one of the measuring components the consumption synchronization process will be initiated to align the previously estimated data with the newly received data. Note: this process is what this section will be concentrating on.

Consumption synchronization is a process by which measurement data for related measuring component is kept synchronized by re-estimating existing measurements when measurement data for related measuring components is received.

For example, consider the following scenario:

- A meter has both scalar and interval channels, and the scalar register channel is considered the "primary" or more reliable channel. The total consumption for each channel should match the consumption for the other channel.

Copyright © 2026, Oracle and/or its affiliates.

- For a particular day, the daily reading for the "secondary" (interval) channel is received before reads from the scalar register have been received. This reading contains two missing intervals, which are estimated during VEE.
- After the interval reading has been finalized, a scalar register reading for the same day is received. The total of this reading is slightly different than the total of the interval reading due to the estimation of the missing intervals.
- The consumption synchronization process can trigger a re-estimation of the two estimated intervals to adjust the total of the interval reading to match that of the scalar register reading.

Other examples of situations that can be resolved using the consumption synchronization process include the following:

- Register Readings whose start/stop times do not align with interval boundaries
- A broken meter for which estimated readings are created for several days
- Requirements that readings from both channels match exactly even if neither contains estimated measurements

Primary Versus Secondary Measuring Components

When measuring components are related, one is considered the "primary" measuring component, and the other is considered the "secondary" measuring component. How the consumption synchronization process works for a given meter is based largely on which measuring component is considered primary and which is considered secondary.

The following are important principles regarding the primary/secondary measuring components:

1. The primary measuring component has the more reliable data.
2. The secondary measuring component should validate that its consumption aligns with the primary measuring component.
3. If the secondary measuring component needs to be estimated, it should estimate using the available data from the primary measuring component.
4. When possible, initial measurements for the primary measuring component should be processed prior to those for the secondary measuring component (Note: the D1-IMD batch process will process initial measurements in this order). In some scenarios when the secondary measuring component is read more frequently than the primary measuring component, it can be advantageous to violate this principle in favor of being able to execute smaller more frequent loads of measurement data.
5. Periodic estimation should be performed on the primary measuring component first, and then the secondary measuring component.

Understanding Consumption Synchronization Processing

This section provides an overview of the process that supports keeping consumption for related measuring components synchronized.

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

1. An initial measurement is received (via initial load or manual) for a measuring component whose device configuration is consumption synchronization enabled.

Copyright © 2026, Oracle and/or its affiliates.

2. The consumption synchronization process evaluates the related "consumption reference measuring component" to identify any measurements for the time period of the initial measurement are consumption sync eligible (typically estimated).

   If eligible measurements are found a consumption synchronization activity is created. Note: the activity will not immediately process it will remain in an initial status until a background process is executed.

3. When the consumption sync background process is executed the activity initiates the re-estimation process by creating estimation initial measurements for each measurement that must be synchronized:

   - For scalar measuring components, this initial measurement contains no measurement value.
   - For interval measuring components, this initial measurement can contain a combination of "Regular" measurements (based on existing final measurements) and measurements that require estimation (based on and corresponding to the measurements with condition codes that fell between the range defined for "consumption synchronization" in step 2).

4. Each estimation initial measurement that was created will execute the estimation VEE group rules to recalculate the synchronization eligible measurements.

About Interval Prices and Price Sets

Interval price sets represent set of interval pricing values used in various types of calculations, including program settlement transactions used with Customer Program Management, settlement calculations used with Market Settlements Management, or pricing calculations used in billing or rating. Interval price values can include market prices or real-time prices. When using multiple sources of interval prices, each source might be represented by a different interval price set.

Interval prices are individual prices based on a defined interval size (for example, hourly prices).

Interval price are defined by the following data:

- **Interval Price Set Type**: The interval price set type on which the interval price set is based. The interval price set type defines a number of attributes of the price set, including the interval size for the price intervals (such as 1 hour, 15 minutes, and so on), and the unit of measure, time of use period, and service quantity identifier used to define the price values.
- **External Identifier**: An identifier for the interval price set used in external systems
- **Time Zone**: The time zone associated with the interval price set
- **Interval Prices**: A set of interval price values belonging to the interval price set, including the following:
  - **Price Date/Time**: The date and time of the interval price
  - **Price**: The price value (based on the unit of measure defined for the interval price set's type)

System Data

Interval Price Sets are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Measuring Component (D1-MEASRCOMP)
Business Object(s)	Interval Price Set (D1-IntervalPriceSet)

Interval Prices are based on the following system data:

Data Type	Description
Maintenance Object	Measurement (D1-MSRMT)
Business Object(s)	Interval Prices (D1-IntervalPrices)

Maintaining Interval Price Sets

You use the **Interval Price Set** portal to display and maintain interval price sets and interval prices

Refer to About Interval Price Sets for more information.

You can access the portal from **Menu**, select **Interval Price**, then select **Interval Price Set**. You are brought to a query portal with options for searching for a specific record. Once an interval price set has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set**: This zone displays the selected interval price set.
- **Interval Prices**: This zone displays the individual interval prices for the interval price set.

Maintaining Interval Prices

You use the **Interval Prices** zone to maintain interval prices for an interval price set. You can either manually add or upload prices.

To manually add prices, click **Add** in the upper right corner of the zone. The **Interval Prices Add** dialog opens. The **Interval Price Set ID** is prepopulated based on the interval price set.

- Enter the **Price Date/Time** and **Price** values for each interval price in the **Interval Prices** section.
- Click **Save** to save the interval prices.

To upload prices, click **Upload** in the upper right corner of the zone. The **Interval Prices Upload** dialog opens. The **Interval Price Set ID** and **Target Time Zone** are prepopulated based on the interval price set.

- Select the **Source Time Format** (Local or Standard) for the price data.
  - **Local** : Date/time values adjusted to "Standard" time (reflecting DST if applicable) and both are captured on the Measurement table
  - **Standard**: Date/time values are captured in both Standard and Local time on the Measurement table
- Specify the **Source Time Zone** for the price data. This is the time zone from which the price data originated. This allows for conversion to the time zone defined for the interval price set.

Copyright © 2026, Oracle and/or its affiliates.

- To upload prices, click **Choose File** to search for and select a comma separated values (CSV) file containing price data and click **Submit**. The **Interval Prices Upload** dialog will display the price data contained in the file. Note that you can edit the price data before saving it.
- Click **Save**to save the interval prices.

> **Note**
>
> You can add prices using both manual entry and upload, but uploading data will overwrite existing prices for the same date/time.

To update one or more prices, select the price(s) to update and click **Edit**. The **Price** dialog opens displaying the selected prices. Edit the prices as needed and click **OK**. Click **Cancel** to close the dialog without making any changes.

To delete one or more prices, select the price(s) to delete and click **Delete**. The **Price** dialog opens asking you to confirm the deletion. Click **OK** to delete the selected prices. Click **Cancel** to close the dialog without deleting the prices.

Interval Prices File Format

The file format used to upload interval prices are comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column must included in the first row of the input file.

Field	Data Type	Sample Content	Description
dt	DATE	2023-02-11-12.00.00	Date and time for the interval price
value	NUMBER	0.35	Interval price amount

Sample File

The following is a sample file for interval prices spanning a six hour event.

```
dt,value 2023-02-11-12.00.00,0.35 2023-02-11-13.00.00,0.35
2023-02-11-14.00.00,0.35 2023-02-11-15.00.00,0.35 2023-02-11-16.00.00,0.35
2023-02-11-17.00.00,0.35
```

VEE

- Introduction to VEE
- Validations
- Estimations
- Editing

Introduction to VEE

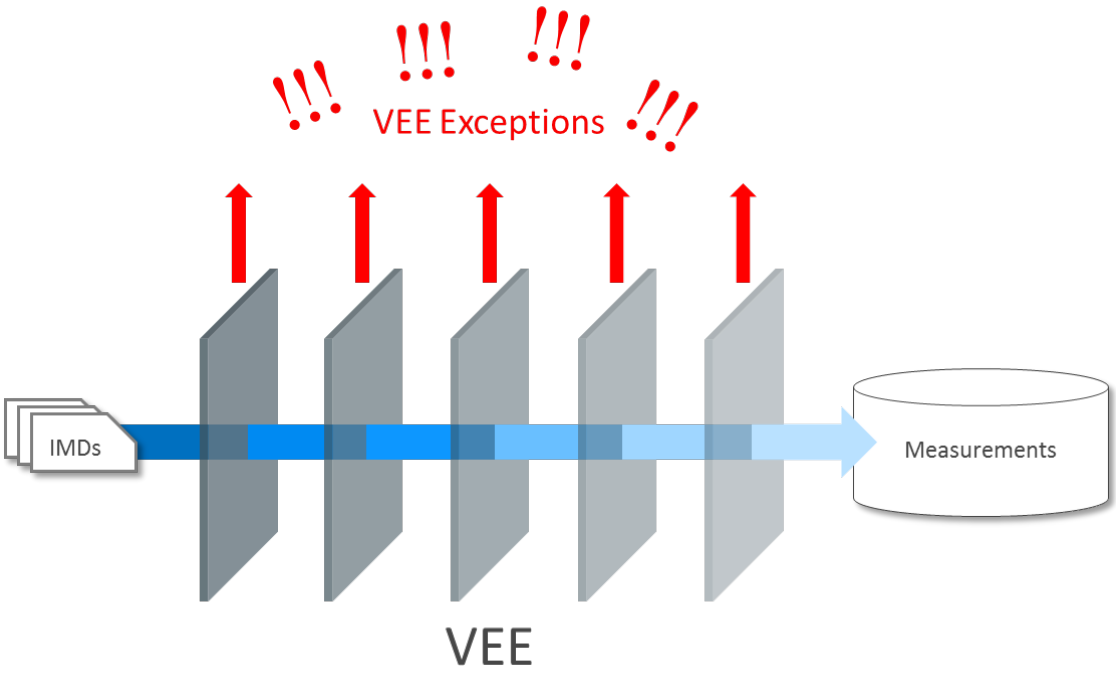
Copyright © 2026, Oracle and/or its affiliates.

- About VEE
- Decision-Making VEE Rules

About VEE

As measurements are received into Oracle Utilities Meter Data Management there is a need to ensure the data is accurate, fill in any missing data with estimations, and flag any failures for review by users. The VEE (Validation, Estimation, and Editing) process is the primary method for ensuring that these objectives are met.

VEE acts as a multi-layered filtration system for IMDs. Raw data is received from the Head End as Initial Measurement Data and only clean, validated data is received on the other end in the Measurement table. The IMD has both a Pre-VEE and Post-VEE section that details any data changes made as part of VEE.



Initial measurement data can come from different sources, such as a head-end system or estimation processes, or it can be manually created by a user (to override or estimate consumption). Measurement data from these different sources might use different VEE Rules. For example:

- Initial measurements sent a head-end system might use strict VEE Rules
- Initial measurements created by a user (to override or estimate consumption) may use less strict rules
- Initial measurements created by the system to estimate consumption have very few (if any) VEE Rules

You can define different VEE Groups for different roles and sources of incoming measurements. The base package roles are:

Copyright © 2026, Oracle and/or its affiliates.

- **Initial Load**: Used for initial measurement data received from a head-end system or import process
- **Initial Load - Streaming**: Used for initial measurement data received from a streaming import process
- **Manual Override**: Used for initial measurement data manually created by a user or created by the reconciliation process for scalar or subtractive interval measuring component types
- **Estimation**: Used for initial measurement data estimated by the system
- **Projection**: Used for "what if" services that are projecting future energy use for a measuring component
- **Post-ML**: Used for processing rules after machine learning processing, when using Machine Learning Anomaly Scoring.

Refer to About Initial Measurement Data for additional functional information about the reconciliation process.

A measuring component's Measuring Component Type can define "fallback" VEE groups for each of these roles. In addition, an individual measuring component can specific a VEE group for each role. If the measuring component doesn't have a VEE group specified for a role, the "fallback" VEE group defined for the measuring component type is used.

The VEE process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.

VEE Groups

VEE groups are collections of VEE Rules that are applied to initial measurement data. During the VEE process, the system executes the VEE Rules defined in each VEE group. The rules within a VEE group are defined in a specific sequence, allowing control over the order in which the rules are executed.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component. VEE groups associated to a measuring component override those assigned to a measuring component type.

When using Machine Learning Anomaly Scoring (LINK), you can define a VEE Group that is executed after the machine learning processing ..

For information on configuration, reference the related section for Configuring VEE Groups in the Administration Guide.

VEE Rules

The specific validation and estimation processing performed on initial measurement data is defined in individual VEE Rules, each performing a specific set of targeted logic. The base product contains many VEE Rules you can use in your implementation, but you can also create your own custom VEE Rules.

Some VEE Rules generate VEE Exceptions if the initial measurement data fails the conditions specified for the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE Rules change the Post-VEE quantities of initial measurement data, but VEE Rules can change anything on an initial measurement.

Copyright © 2026, Oracle and/or its affiliates.

For a list of validation VEE Rules, reference the About IMD Validations section. For a list of estimation VEE Rules, reference the About IMD Estimations section. For a list of general VEE Rules that handle decision-making within the VEE process, reference the Decision-making VEE Rules section.

For information on configuration, reference the related section for Configuring VEE Rules in the Administration Guide.

VEE Exceptions

Each VEE Rule defines an exception type and severity that specify how exceptions are tracked by the system. When an initial measurement fails a validation, an exception of the type specified for the failed VEE Rule is created. A single initial measurement can have multiple exceptions, one (or more) for each rule the measurement fails. This allows users to see all of the problems detected during the VEE process.

There are three levels of severity for VEE Exceptions:

- **Info**: Used to highlight something interesting, but not sufficient to cause the initial measurement to be put into the Exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.
- **Issue**: Used to report a problem that will prevent the initial measurement from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one issue exists after all rules have been applied, the initial measurement is transitioned to the Exception state.
- **Terminate**: Used to report a severe issue that will cause the VEE process to stop and the initial measurement to be transitioned immediately to the Exception state. Only one terminate exception can be issued (as the first one causes VEE to stop on an initial measurement).

Note that exceptions are not deleted when an initial measurement is adjusted or corrected. After any issues are corrected or the initial measurement is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

For information on configuration, reference the related section for Configuring Exception Types in the Administration Guide.

VEE Rule Eligibility Criteria

Each VEE Rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of VEE groups you need to create, because it allows a single VEE group to have conditional VEE Rules based on eligibility criteria (rather than requiring a distinct VEE group for every combination of VEE Rules). Below are a few examples of ways you can use eligibility criteria:

- You might create a rule that compares interval consumption against related scalar consumption (such as might be the case with a device with both interval and scalar measuring components). This rule might use eligibility criteria that specifies that the rule is only applied if the initial measurement's measuring component has a corresponding scalar register measuring component.
- You might want to have a common VEE group but only have certain rules execute to execute based on Unit of Measure. An eligibility criteria can be setup so that a VEE Rule only executed for kWh whereas is skipped for kVARh.
- Another example might be a rule that compares the current consumption against standard consumption for measuring components of a certain type for the first six months after installation. You might create eligibility criteria that specifies that the rule is only applied if

Copyright © 2026, Oracle and/or its affiliates.

the measuring component's Device Configuration has been installed at a Service Point for less than six months.

For information on configuration, reference the related section for Configuring VEE Rules in the Administration Guide.

Decision-Making VEE Rules

In addition to the VEE Rule Eligibility Criteria, there are VEE Rules delivered as part of the base product that help with decision-making when executing VEE (listed below). For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Purpose
Exception Handler	This rule allows for termination of the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions.
Execute VEE Group	This rule performs a call to execute a separate VEE Group which includes execution of all VEE Rules within that group.
Successful Termination	This rule allows VEE to be successfully terminated based on a list of exceptions.
VEE Group Matrix (Factor)	This rule provides a way to choose different instances of a VEE Rule using a Factor. This factor leverages characteristics that are defined on Service Point, Device, or Measuring Component.

Validations

- About IMD Validations
- Reviewing and Resolving VEE Exceptions
- Usage Transaction Validations

About IMD Validations

Validation of the Initial Measurement Data on the way in to Oracle Utilities Meter Data Management occurs by leveraging the diverse library of VEE Rules. Below is a list of the validation VEE Rules that are provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Applicable Data Type(s)	Purpose
Consecutive Interval Check	Interval	This validation rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data.

Copyright © 2026, Oracle and/or its affiliates.

Duplicate IMD Check	Interval or Scalar	This rule allows for a duplicate IMD to be flagged.
Dynamic Comparison Validation	Interval or Scalar	This powerful, flexible validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Then you define formulas (no programming required) for the comparison of current measurements to these statistics.
Ensure IMD Exists for Sibling MCs	Interval or Scalar	This rule validates that an IMD exists for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time.
Final Measurement Replacement	Interval or Scalar	This validation rule allows you to define a variety of configuration options to decide if new data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking.
High/Low Check	Interval or Scalar	This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown.
Inactive Measurement Check	Interval or Scalar	This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a Usage Subscription.
Interval Size Validation	Interval	This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type.

Copyright © 2026, Oracle and/or its affiliates.

Interval Spike Check	Interval	This rule checks for spikes within an IMD and generates an exception if one is found.
Multiplier Check	Interval or Scalar	This rule validates that the register multiplier supplied with the IMD is equal to the multiplier stored on the measuring component. If not, an exception is created using the register multiplier exception type and severity configured on the rule.
Negative Consumption Check	Interval or Scalar	This rule flags any IMD where the total consumption is negative.
Prolonged Estimation Check	Interval or Scalar	This validation rule creates an alert when a device has been estimated for an extended period of time.
Raise Missing Quantity Exception	Interval	This rule flags any missing interval data.
Sum Check	Interval or Scalar	This rule is used to compare the difference between interval data to scalar data for a period of time, or between a set of TOU scalar reads to a "Sum" scalar reading.
Unit of Measure Check	Interval or Scalar	This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type.
Zero Consumption Check	Interval or Scalar	This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid exceptions in that case.

Refer to About VEE for additional functional information.

Reviewing and Resolving VEE Exceptions

All To Dos generated by a VEE Exception will link back to the Initial Measurement Data that originally failed to create the VEE Exception.

Users have a number of options for dealing with initial measurements with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can reinitiate the VEE process for the IMD using the **Perform VEE** button.
- A user can **Discard** the initial measurement.

Copyright © 2026, Oracle and/or its affiliates.

- A user can use the **Force Complete** button on the IMD in the scenario that they want to approve the exception and the measurement information in the IMD. *Note:* No VEE processing is performed on initial measurement data that is force completed. Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the initial measurement entered the Exception state.

Refer to About VEE for additional functional information.

Usage Transaction Validations

There are also validations available during the execution of usage calculation. For more information on these reference the Reviewing and Resolving Usage Transactions section.

Estimations

- About IMD Estimations
- Usage Transaction Estimations

About IMD Estimations

Below is a list of the estimation VEE Rules provided as part of base product that are responsible for filling the missing data within an IMD. For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Applicable Data Type(s)	Purpose
Interval Adjustment From Scalar	Interval	This rule performs adjustments to interval values based on the values from the associated scalar data on the same device.
Interval Averaging Estimation	Interval	This rule finds historical interval data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD.
Interval Create Estimation IMD for Gap	Interval	This estimation rule creates a new Estimation IMD to fill gaps between the interval data received from the Head End. ***Note:*** *this estimation rule acts differently from the other rules as it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.*
Interval Interpolation Estimation	Interval	This rule attempts to interpolate gaps within an IMD using prior and subsequent intervals as starting points for linear interpolation.

Copyright © 2026, Oracle and/or its affiliates.

Interval Profile Estimation	Interval	This rule estimates any missing interval data for the IMD based on a referenced Profile Factor.
Interval Proxy Day Estimation	Interval	This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement.
Scalar Calculation From Interval	Scalar	This rule performs adjustments to scalar values based on the values from the associated interval data on the same device.
Scalar Estimation	Scalar	This rule finds historical scalar data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD.
Scalar Profile Estimation	Scalar	This rule will estimate any missing scalar data for the IMD based on a referenced Profile Factor.
Scalar Proration	Scalar	This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries. It will also take into account any related interval data within the same period to exclude from the calculation.
Subtractive Interval Adjustment Rule	Subtractive Interval	This rule performs adjustments to qualifying interval consumption values for subtractive interval measuring components based on an adjustment target calculated using a start and stop reading for the period that encapsulates the intervals to adjust.

Refer to About VEE for additional functional information.

Usage Transaction Estimations

There are also estimations available at the time of usage calculation. For more information on these reference the Reviewing and Resolving Usage Transactions section.

Editing

- About IMD Editing

Copyright © 2026, Oracle and/or its affiliates.

- Editing Initial Measurements
- Generating New Initial Measurements

About IMD Editing

At times, the values received in as Initial Measurement Data need to be edited by a user. This is covered in the Editing Initial Measurements section.

If measurement data is found to be incorrect, a new IMD must be generated to correct the data. This is covered in the Generating new Initial Measurements section.

Refer to About VEE for additional functional information.

Editing Initial Measurements

Initial measurements can be manually edited via functions available in the Interval Initial Measurement Lens or Scalar Initial Measurement zones in the Initial Measurement portal. Manual edits can include changes to the measurement values or condition codes.

By default, manual edits made by users of these zones are not logged on the Log tab.

Refer to Configuring Measurement Logging for additional information on how initial measurement changes can be logged.

Generating New Initial Measurements

The main location for viewing measurements and generating new IMDs for any discrepancies is through the 360 Degree View Measuring Component portal. On the portal many of the zones provide ways to apply measurement functions that will generate a new IMD.

Usage

- Introduction to Usage Calculation
- Usage Subscriptions
- Usage Subscription Market Participants
- Usage Transactions
- Usage Requests
- Usage Calculation Requests
- Usage Subscription Quantities
- Time of Use Maps
- Dynamic Options
- Consumption Extract Requests
- Contacts

Copyright © 2026, Oracle and/or its affiliates.

Introduction to Usage Calculation

- About Usage Calculation

About Usage Calculation

Oracle Utilities Meter Data Management can calculate and publish usage calculated from measurement data to service providers on an ongoing basis. In addition, external systems can request usage whenever needed. Usage calculations derive a usage transaction's usage quantities using the measurements linked to a usage subscription's service points.

Before bill determinants can be calculated, you must first create a Usage Subscription. The usage subscription defines the external system to which the data should be sent as well as the usage calculation group that will be used for calculation. The usage calculation rules defined for the usage calculation group are responsible for performing the specific calculations, validations, or estimations for bill determinants.

Usage Service Quantities (bill determinants) are derived from the Measurements of the measuring components installed at the Usage Subscription's Service Points during the calculation period. A Service Point is linked to measuring components through an Install Event linked to the measuring components' Device Configuration.

The results of all calculations performed by Usage Subscriptions are stored as a Usage Transaction.

At any instance in time:

- A Usage Subscriptions may be linked to multiple Service Points
- A Service Point may be linked to a single Device Configuration
- A Device Configuration may have multiple measuring components

The calculation period for bill determinant calculations can span many days and over this period:

- The Service Points linked to the Usage Subscription can change (Service Points can be added and removed)
- The Device Configurations installed at the Service Point can change (due to device reconfigurations and meter exchanges)

This means that values for each Usage Subscription can be calculated using multiple Service Points and measuring components.

The Usage Calculation process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.

Usage Calculation Groups

Usage Calculation Groups are collections of usage calculation rules that are used to either calculate bill determinants or validate bill determinants. During the Usage Transaction process, the system executes the usage calculation rules defined in the usage calculation group referenced on the usage subscription. The rules within a usage calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

For information on configuration, reference the related section for Configuring Usage Calculation Groups in the Administrattive User Guide.

Copyright © 2026, Oracle and/or its affiliates.

Usage Calculation Rules

The specific calculation, validation, or estimation processing performed on a Usage Transaction is defined in individual usage calculation rules, each performing a specific set of targeted logic. The base product contains many usage calculation rules you can use in your implementation, but you can also create your own custom usage calculation rules.

For a list of validation usage calculation rules, reference the About Usage Validations section. For a list of estimation usage calculation rules, reference the About Usage Estimations section. For a list of all other usage calculation rules that handle calculation of bill determinants, reference the Inventory of Other Usage Calculation Rules section.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administration Guide.

Usage Transaction Exceptions

Each usage calculation rule defines an exception type and severity that specify how exceptions are tracked by the system. When a Usage Transaction fails some part of a usage calculation rule, an exception of the type specified for the failed usage calculation rule is created. A single Usage Transaction can have multiple exceptions, one (or more) for each rule that failed. This allows users to see all of the problems detected during the Usage Calculation process.

There are three levels of severity for Usage Transaction Exceptions:

- **Information**: Used to highlight minor issues, but not sufficient to cause the Usage Transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the usage transaction from being sent. Multiple "issue exceptions" can be created during usage transaction processing. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the Usage Calculation process to stop and the Usage Transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a Usage Transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full usage period should always be set to Terminate as an Exception Severity.

Note that exceptions are not deleted when a Usage Transaction is adjusted or corrected. After any issues are corrected or the Usage Transaction is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

In addition to exceptions, usage processing can also trigger the creation of To Do Entries related to failed validations. If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the usage transaction is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the usage transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

For information on configuration, reference the related section for Configuring Usage Transaction Exception Types in the Administration Guide.

Copyright © 2026, Oracle and/or its affiliates.

Usage Calculation Rule Eligibility Criteria

Usage calculation rule eligibility criteria are user-definable conditions that could cause a given usage calculation rule to be applied or skipped. This can involve the evaluation of some attribute of the usage subscription or service point, or something else entirely. A usage calculation rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administration Guide.

Usage Subscriptions

- About Usage Subscriptions
- Maintaining Usage Subscriptions

About Usage Subscriptions

A usage subscription is an ongoing request to send one or more service points' usage to one or more external systems. The usage subscription defines which usage calculation group should be used to calculate service quantities (often referred to as bill determinants).

Usage subscriptions must reference a:

- Service Provider. The service provider is used as the identity of the subscribing system. In other words, you must set up a service provider for any system that subscribes to bill determinants.
- Contact: Contacts are individuals or business entities with which a company has contact. A contact exists for every individual or business related to a usage subscription. A single usage subscription can have many contacts, and a single contact may be referenced on many different usage subscriptions. Contacts have a 1-to-1 correlation with a "person" in a customer information system (CIS) and the CIS is considered the system of record for contact information.

> **Note**
>
> Usage Subscriptions can be created from Customer Care and Billing Service Agreements via Customer To Meter Master Data Synchronization.

Maintaining Usage Subscriptions

This portal is used to display and maintain a usage subscription.

Refer to About Usage Subscriptions or About Usage Calculation for additional functional information.

You can access the portal from **Menu > Usage > Usage Subscription**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Usage Subscription**: Defines basic information about the usage subscription, including factor overrides, usage groups, and service points.
- **Usage Subscription Market Participants**: Lists market participants associated with the usage subscription (via Usage Subscription Market Participants)
- **Sub Usage Subscriptions**: Lists the first fifty "sub" usage subscriptions for the master usage subscription displayed in the Usage Subscription zone. This zone is only displayed for "master" usage subscriptions.

> **Note**
>
> Usage Subscriptions can be created from Customer Care and Billing Service Agreements via Customer To Meter Master Data Synchronization. Usage subscriptions created from service agreements should not be modified using this portal.

Usage Subscription Market Participants

- About Usage Subscription Market Participants
- Maintaining Usage Subscription Market Participants

About Usage Subscription Market Participants

Usage subscription market participants connects a Market, Market Participant and Usage Subscription together over a specified date range. Usage subscription market participants can be created either via the **Usage Subscription Market Participant** portal, or by adding a market participant to a usage subscription via the **Usage Subscription** portal.

Usage subscription market participants are defined by the following:

- **Usage Subscription**: The usage subscription for which this usage subscription market participant was created.
- **Market Relationship Type**: The market relationship (Distributor, Supplier, etc.) of the market participant and its role within a market.
- **Start Date/Time**: The start date/time for when the record is applicable.
- **End Date/Time**: The end date/time for when the record is applicable.
- **Market Participant**: The market participant associated with the **Usage Subscription**, including an optional ID used by the market participant to uniquely identify the account (usage subscription).

Maintaining Usage Subscription Market Participants

You use the **Usage Subscription Market Participant** portal is used to display and maintain usage subscription market participants.

Refer to About Usage Subscription Market Participants for additional functional information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal from the **Main > Usage > Usage Subscription Market Participant**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Market Participant**: This zone displays the basic attributes of the usage subscription market participant
- **Sub Usage Subscriptions**: lists sub usage subscriptions for the usage subscription market participant's usage subscription

Usage Transactions

- About Usage Transactions
- About Usage Validations
- Reviewing and Resolving Usage Transactions
- Triggering Adjustment Notifications
- Inventory of Other Usage Calculation Rules

About Usage Transactions

Usage Transactions are records of bill determinant calculations for a Usage Subscription. All of the Service Quantities calculated are stored as part of the Usage Transaction.

Most requests for usage result in the creation of a usage transaction, but it is possible for an external application to invoke the usage calculation engine real-time. In other words, usage can be retrieved for a usage subscription real-time without creating a usage transaction. This technique is only recommended for online requests, not as part of batch processes.

Usage Calculation Group Determination

The first step in calculating the usage is to determine the usage calculation group holding the rules set used in calculating or validating the bill determinants. There are two ways of configuring the usage calculation group to use in the processing. It may be specified on the usage subscription during its creation with an effective and expiration date. If there are no usage calculation groups in effect for the usage transaction, the fallback usage calculation group specified in the usage subscription type will be used. The other way is to specify an algorithm – Determine Usage Calculation Group Override, on the usage subscription type. This option provides flexibility in cases where special or custom logic will be required to identify the usage calculation group. Base product delivered an algorithm that identifies the usage calculation group based on the rate and installed device. Additional details can be found in the Detailed Description of the D2-DRVUSGGRP Algorithm Type.

Calculation Period

Usage transaction requests must specify the date range for the usage transaction. This date range is referred to as the calculation period for the usage transaction. The dates that define the calculation period are specified by a subscribing system when it requests a usage transaction.

Some usage calculation rules have been enhanced to support usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement

Copyright © 2026, Oracle and/or its affiliates.

data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

Multiple Device Configurations and the Calculation Period

In situations where a meter exchange has taken place at a service point, more than one device configuration can be in effect during the usage transactions cutoff period (the time period between the **End Date/Time From** and **End Date/Time To** values).

To account for this, the **Meter Exchange Option** on the Usage Subscription Type can be used to specify how the usage calculation should handle multiple device configurations in effect during the cutoff period. If the flag is set to 'Defer Calculation', new device configurations are excluded in the current usage transaction. If the flag is blank or set to 'Calculate Usage', new device configurations are included in the current usage transaction.

Date Breaks

As noted above, date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point. For example, suppose a subscribing system requests usage for the month of January. The customer for this request has an interval meter, and the customer's usage is calculated by applying a TOU map to their interval consumption. The subscribing system detects that the customer's rate changed in the middle of January (on January 16) and wants the TOU consumption calculated in two "chunks" (before and after the rate change). Because the customer has an interval meter, the exact consumption amounts before and after the rate change can be precisely calculated (as opposed to calculating each period's amount by dividing the total usage by the number of days in each period).

One approach to this situation would be for the subscribing system to request two usage transactions (where each has the desired date range). Another approach is for the subscribing system to request a single usage transaction with date breaks that define the date ranges before and after the rate change. To continue the above example, a usage transaction could be created with a date break on January 16.

If a usage transaction has date break(s), the usage calculation engine segregates the usage into multiple usage periods based on the date breaks. If there are no date breaks in the usage transaction, a single usage period is created for the entire calculation period.

As of the v2.0.0 release, the only way a usage transaction can have date breaks is if the subscribing system supplies these when it requests usage (only the subscribing system knows if and when its prices and pricing rules change during the billing period). If an implementation has additional criteria that causes date breaks, these criteria can be easily added to the usage transaction's business rules.

Service Quantities

Every usage period created for a usage transaction contains one or more service quantities. Service quantities are calculated by the usage calculation group's rules specified for the usage subscription from which the usage transaction is created. For example, a usage transaction might have service quantities calculated by a single rule that applies a TOU map to the kWh channel on the device configuration installed at the service point.

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

Copyright © 2026, Oracle and/or its affiliates.

Service Quantity Rounding

The manner in which service quantities are rounded can be defined using the **SQ Rounding Details** parameters. These parameters allow configuration of the method by which quanitities are rounded (Up, Down, or Nearest) and the number of decimal positions retained after rounding.

Quality Assessment for Bill Print Options

Usage transaction service quantities can also contain information about the quality of the measurements used in their calculations and other information that can be sent to an external customer information or bill print system. This information is included in both the usage transaction and its related outbound message.

Quality Assessment

The Daily Scalar Usage Calculation Rule, Get Interval Data, Get Scalar Details, and Get Time of Use Mapped Usage usage calculation rules include a "Perform Measurement Quality Assessment" option that indicates (yes or no) if the measurement conditions of the measurements used when calculating usage should be evaluated and included in the resulting usage transaction and outbound message. When enabled (set to "Yes"), the usage calculation rule algorithm evaluates the condition codes of the measurements used in the calculation to determine if they fall within a "Regular" range of conditions (this range is defined by the "Bottom Regular Condition Range" and "Top Regular Condition Range" algorithm parameters). The result of this assessment is displayed in the "Data Quality Assessment" column in the Usage Periods section which indicates if the service quantity is based on "regular" or "non-regular" data. (If the "Perform Measurement Quality Assessment" option is disabled, the "Data Quality Assessment" column displays "No Assessment Done). In addition, the "Source Measurement Quality List" column contains XML that contains a list of measurements aggregated by condition/quality.

Maximum/Minimum Dates and Times

The Get Interval Data, Get Time of Use Mapped Usage, and Vector and Service Quantity Math usage calculation rules can be configured to identify the date/time of the maximum or minimum measurement value used when calculating the service quantity. For these functions, if "Max" or "Min" is selected as the "Calculation Function" (or the "Comax" and "Comin" functions for the "Vector and Service Quantity Math" rule), the date/time of the maximum/minimum value will be included in the resulting usage transaction and outbound message. The "SQ Highlight Date/Time List" column contains XML that contains a list of maximum/minimum date/times, based on condition/quality.

Usage Transaction Export Configuration

Usage transactions can also include additional information about its related meters and service points, as well as an indication if the service quantity is based on estimated usage. The "Usage Transaction Export Configuration" master configuration is used to control the export of this data. It holds the device and service point business objects used to supply meter and service point information included in the usage transaction and outbound message.

The "Usage Transaction Export Configuration" master configuration also defines the threshold percentage above which a service quantity being exported via the usage transaction outbound message is marked as an estimate. For example, of the "Estimation Threshold Percentage" is set to 35%, any service quantity being exported that was calculated from measurements containing more than 35% estimated usage is marked "estimated".

Copyright © 2026, Oracle and/or its affiliates.

Service Quantities for Scalar Usage

Usage transactions and service quantities for scalar usage differ in some ways from those created from interval usage. For example, suppose a subscribing system requests usage for the month of January for a customer with a scalar meter that has been exchanged mid-month. In this case, the customer's usage is calculated by finding the scalar readings in the requested period (including all meter exchanges), and the subscribing system requires a record of all scalar readings AND a total of their consumption

Scalar rules cannot use date breaks because the system doesn't store interval values so it cannot accurately compute the amount in each period. In this case, scalar usage calculation rules retrieve scalar readings for the service points linked to the usage subscription, and then creates a single usage period for the usage transaction's entire calculation period. The individual scalar readings are captured as "scalar details" in the usage transaction.

To continue the example above, there would be 2 entries in the Scalar Details, one for each meter, and a single usage period that contains the total consumption for the entire calculation period. The tables below illustrate what this might look like.

Usage Estimation

When calculating usage for monthly read scalar meter, the end read may not be available by the end of the bill cycle window. If the request indicated that estimated measurement is allowed, the Get Scalar Details usage calculation rule creates an estimated end read that will be used to calculate the scalar usage.

Calculating Usage for Items

Usage transactions can also calculate usage for items installed at service points. Usage for items is calculated through use of Get Item Counts and Consumption usage calculation rule. This rules calculate usage based on the number of items installed at each service point and the average daily service quantity defined for each item's type. The specifics of how usage is calculated for items is based on the service point category (defined on the service point type) at which the items are installed (item or multi-item).

Hypothetical Calculations - Rate Comparisons

When Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Self Service, customers can perform rate comparison calculations to see the impact to their usage and billing charges based on a set of user-defined usage adjustments. Rate comparison usage transactions are flagged "Hypothetical Calculation" in the **Calculation Mode** field.

Profile Accumulation Usage Calculation Rules and Usage Adjustment Profile Factors

Rate comparison usage transactions can be created through the use of Profile Accumulation usage calculation rule. This rule combine a service point's measurements with "usage adjustment" measurements derived from measurement data for profile measuring components defined as characteristic values for one or more profile factors.

Usage adjustment profile factors are associated to usage adjustment types defined in the Usage Adjustment Type extendable lookup and rates defined in the CC&B Rate Schedule extendable lookup (this association is defined in the Self-Service Master Configuration). When the usage transaction is created, the rate determines the usage calculation group to execute and usage adjustment profile factors to include in the calculation based on the CC&B Rate Schedule extendable lookup.

Copyright © 2026, Oracle and/or its affiliates.

Note: The Profile Accumulation usage calculation rule should be configured with eligibility criteria such that they are only executed when the Calculation Mode for the usage transaction is set to "Hypothetical Calculation" (D2HC).

Including Summary Service Quantities and Interval Data Snapshots with Usage Transactions

Usage transactions can be configured to include summary service quantities and an interval data "snapshot" along with service quantities and other information sent to external systems. This used when sending usage to a billing system that performs interval-based pricing calculations.

Only usage transactions created (in whole or in part) by either the Get Interval Data or Vector and Service Quantity Math usage calculation rules can include interval data snapshot information. These rules use the **Extract Interval Data** flag to indicate ("Yes" or "No") if a "snapshot" of the interval data resulting from the usage calculation rule should be sent to external systems when usage from the usage transaction is sent to external systems.

In the base package, the Usage Transaction Outbound Message (D2-UsageTranOutboundMesg) business object is used to create outbound messages containing usage transactions. This business object includes the "Build Summary SQs" (D2-BLDSUMSQ) Pre-Processing algorithm which calculates and creates the interval data snapshot and other summary service quantities.

How Summary Service Quantities and the Interval Data Snapshot is Calculated

As noted above, the "Build Summary SQs" (D2-BLDSUMSQ) algorithm builds the Summary Usage Periods List based on the usage transaction's details. For every usage period within the usage transaction:

An entry is inserted into the summary service quantity list for every distinct UOM/TOU/ SQI combination in the usage period service quantity list.

An entry is inserted into the service point service quantity list for every distinct service point ID/UOM/TOU/SQI combination in the usage period service quantity list.

Interval data for multiple service quantities are combined into a single interval data curve per UOM/TOU/SQI combination. Intervals are summarized and scaled based on a common SPI

The "Build Summary SQs" (D2-BLDSUMSQ) algorithm uses the "Target Interval Size" parameter to define the interval size of the combined curve when one or more curves are combined with different interval sizes. For example, when combining interval data with 15 minutes intervals (SPI 900) and data with 60 minute intervals (SPI 3600), a Target Interval Size of 60 minutes (SPI of 3600) would result in combined data with an interval size of 3600 (the 15 minute intervals would be scaled up to 60 minute intervals before being combined with the 60 minute interval data).

Refer to About Usage Calculation for a broader functional view of the overall process. For information on how to configure the request and response process for Usage Transactions, reference the Integrating Usage Transactions section of the Admin Guide.

About Usage Validations

For the sake of keeping the volume of To Dos to a manageable level it may make sense to shift certain validations to the usage transaction process rather than the IMD load process. The validation should be done on a usage transaction when the result is only crucial for the systems receiving usage transactions rather than more real time needs such as online presentation to a customer. Below is a list of the usage calculation rules used for validating

Copyright © 2026, Oracle and/or its affiliates.

usage transactions that are provided as part of base product. To get more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose
Usage High/Low Rule	Interval or Scalar	This validation rule compares the Service Quantities of the Usage Transaction to historical values. If the current value is too high or too low compared to historical data then an exception is thrown.
Validate Against Tolerance	Interval or Scalar	This usage calculation rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule.
Business Flag Hold	Interval or Scalar	This usage calculation rule can stop a usage transaction from proceeding when there have been business flags for the applicable service points. The hold can either be indefinite or set to expire a configurable amount of time prior to the calculation window ending.

Refer to About Usage Transactions or About Usage Calculation for additional functional information.

Reviewing and Resolving Usage Transactions

This portal is used to review and/or resolve a Usage Transaction.

Refer to About Usage Transactions or About Usage Calculation for additional functional information.

You can access the portal from the **Main > Usage > Usage Transaction**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Transaction**: Defines basic information about the usage transaction, including usage period date ranges, date breaks within the usage period, scalar usage details, and usage quantity details.
- **Service Quantity Overlay**: Displays interval data for each service quantity for the usage transaction's usage period
- **Time of Use By Day**: Displays TOU-mapped interval data by day for the usage transaction's usage period

A number of important actions are available for a Usage Transaction depending on its status:

Action	Allowed Statuses	Details

Copyright © 2026, Oracle and/or its affiliates.

Approve	Approval In Progress Issue Detected (only when the exceptions are marked to require approval)	Allows a user to bypass any exception or failure message for the Usage Transaction and attempt to send it to the requesting system.
Re-Calculate	Approval In Progress Issue Detected	Reruns the entire process of calculation along with any validations (all usage calculation rules) for the Usage Transaction.
Subsequent Correction	Sent	Allows a user to indicate that the usage transaction was corrected after being sent. Sets the status of the usage transaction to Subsequent Correction.
Resend	Send Error	Allows a user to re-send a usage transaction (via its related outbound message) to an external system. Used when an error occurs when sending usage transactions.

Triggering Adjustment Notifications

If any new Measurements are received from a device after that period has been calculated in a Usage Transaction then a discrepancy exists between Oracle Utilities Meter Data Management and the external system (often a customer information system or CIS). As a result, Oracle Utilities Meter Data Management provides a method to notify external systems of these adjustments to Usage Transactions. A **Usage Transaction Correction Processor** activity is created for each usage transaction that may have been impacted by an initial measurement. The **Usage Transaction Correction Monitor** batch (D1-UTCRN) is used to process these activities.

On the External Application for CIS, a processing method of "Usage Trans Subsequent Correction Notification" must be configured to specify the message that should be sent to notify CIS when an adjustment occurs for a Usage Transaction in Oracle Utilities Meter Data Management.

The IMD process handles the logic to look for existing Usage Transactions to determine if an adjustment will be required. The Usage Transaction will be moved to a "Subsequent Correction" status once this adjustment occurs.

Inventory of Other Usage Calculation Rules

Beyond validations and estimations usage calculation rules, the Usage Transaction process involves many other important rules that prepare and calculate Service Quantities. A list of these rules is provided below. For additional detail, follow the provided link that will take you to the Configuration Guide.

Copyright © 2026, Oracle and/or its affiliates.

Pre-Calculation Usage Calculation Rules

Below is a list of the pre-calculation usage calculation rules provides as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose
Alignment and Delay Calcluation Rule	Interval or Scalar (Frequently Read)	This rule can be used to handle two main needs: 1) aligns all measuring components for a Usage Subscription to the same date (whether on the same device or separate devices) 2) delays the usage transaction until the end of the retry window based on the quality of available data and other configured parameters
Check Existence of Installed Device	Interval or Scalar	This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period.

Calculation Usage Calculation Rules

The following is a list of the calculation usage calculation rules provided as part of the base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose
Apply Math (Interval Data)	Interval	This rule is used to perform calculations on interval data and stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions). This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a usage subscription. This rule can also multiply total usage by a factor using a custom formula.

Copyright © 2026, Oracle and/or its affiliates.

Daily Scalar Usage Calculation Rul	Scalar	This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified usage period. It creates bill determinants by taking the difference between the beginning and ending reading for the bill period. This rule can also be used to provide register readings by measuring component. For consumption values, only the beginning and ending readings are exported This rule supports date breaks (the normal Get Scalar Data rule does not).
Get Interval Data	Interval	This rule is used to get interval quantities from interval measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. This rule retrieves the interval data for measuring components associated to a usage subscription by TOU, SQI and UOM. This rule converts the interval data to another interval length or unit of measure.
Get Item Counts and Consumption	Scalar	This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and Service Point, and calculates item-based consumption.

Copyright © 2026, Oracle and/or its affiliates.

Get Scalar Details	Scalar	This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period. This rule creates bill determinants by summing all scalar readings for the bill period. This rule can also be used to provide register readings by measuring component. All readings are exported by this rule. Note: This rule is used for traditional monthly read meters.
Get Subtractive Interval Details	Subtractive Interval	This rule is used to get interval quantities from subtractive interval measuring components installed on the service points linked to the usage subscription for the specified 'interval' usage period. It also identifies the start and stop readings for each usage period using subtractive interval readings.
Get TOU Mapped Usage	Interval	This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period.
Interval Tier Calculation	Interval	This rule calculates the difference between a source and reference vector. This rule loops through each tier that is configured and calculates the imbalance amount associated to that tier level. This rule breaks down that difference into one to many positive or negative tiers, and create a service quantity for each tier calculated.
Profile Accumulation	Interval or Scalar	This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction.

Copyright © 2026, Oracle and/or its affiliates.

Round and Adjust Usage	Interval or Scalar	This rule copies identified source and target Service Quantities and inserts these as Service Quantities that are rounded and adjusted.
Vector and Service Quantity Math	Interval	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations. Typical uses: Perform math using interval data, e.g., take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc. Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values). Support math functions: sin, cos, square root, etc. Store derived curves in memory that can be used in subsequent calculations Please note, this rule is not as efficient as other rules.

Decision-Making Usage Calculation Rules

There are usage calculation rules delivered as part of the base product that help with decision-making when executing running the usage calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Purpose
Execute Usage Calculation Group	This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group.
Exception Handler	This rule evaluates the exception list that was accumulated during the execution of the calculation rules against exception criteria configured on this rule.

Usage Requests

Usage request records are created when bill determinants are requested during billing. The consumption information is then used to create a bill segment for the associated service agreement.

Copyright © 2026, Oracle and/or its affiliates.

Refer to The Big Picture of Usage Requests for more details

Use **Menu**, and then **Usage**, and then **Usage Request** to open this page. When you select a usage request record, the maintenance portal opens and displays the associated information for the usage request record. You can also navigate to the maintenance portal from the Bill Segment - Usage page.

Refer the embedded help on this portal's zones for more information.

Usage Calculation Requests

- About Usage Calculation Requests
- Maintaining Usage Calculation Requests

About Usage Calculation Requests

Usage calculation request are calculations executed at a customer level where there may be more than one calculation per usage subscription. For example, a customer may have a usage subscription for producing billing determinants but there may be additional calculations that will be executed on a much less frequent basis (annually, quarterly, etc). More specifically, usage calculation requests are used to calculate service point quantities (see About Service Point Quantities for more information).

Usage calculation requests are defined by the following:

- **Usage Calculation Type**: The usage calculation type for the request. Usage calculation types define the service point quantity type and UOM/TOU/SQI for quantities calculated for requests of that type.
- **Start Date/Time**: The start of the date range for the request.
- **End Date/Time**: The end of the date range for the request.

Maintaining Usage Calculation Requests

You use the **Request** portal to view and maintain usage calculation requests.

Refer to About Usage Calculation Requests for additional functional information.

You can access the portal from the **Main > Usage > Usage Calculation Request**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Request**: defines the basic attributes of the usage calculation request
- **Request Log**: lists log activities for the usage calculation request

Usage Subscription Quantities

- About Usage Subscription Quantities
- Maintaining Usage Subscription Quantities

Copyright © 2026, Oracle and/or its affiliates.

About Usage Subscription Quantities

Usage subscription quantities are quantities that are stored at the usage subscription level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given usage subscription to the market wide coincidence peaks.

Usage subscription quantities are defined by the following:

- **Usage Subscription**: The usage subscription for which the quantities were calculated
- **Usage Subscription Quantity Type**: The Usage Subscription Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Usage Subscription Quantities**: Calculated quantities based on **Quantity Identifiers** configured on the Usage Subscription Quantity Type (see Understanding Usage Subscription Quantity Types for more information)

> **Note**
>
> There cannot be more than one Usage Subscription Quantity for the same Usage Subscription and Usage Subscription Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

Maintaining Usage Subscription Quantities

You use the **Usage Subscription Quantity** portal to view and maintain usage subscription quantities.

Refer to About Usage Subscription Quantities for additional functional information.

You can access the portal by selecting **Main**, then **Usage**, then **Usage Subscription Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Quantity**: This zone displays the basic attributes of the usage subscription quantity

Time of Use Maps

- About Time-of-use Processing
- About TOU Maps
- Maintaining TOU Maps

Copyright © 2026, Oracle and/or its affiliates.

About Time-of-use Processing

A type of rate used at many utilities is known as a "time of use" rate. For these rates, the price is based on when the usage occurs. For example, prices are typically higher during peak consumption periods (commonly referred to as "On Peak" periods) to encourage lower use during these times. The time range when a given price is applicable is known as a time-of-use (TOU) period.

Time of use periods can (and often do) change during the year. For example, in the summer, TOU periods might be defined as follows:

Time Of Use Period	Time Periods
On Peak	8:00 AM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday
	9:00 PM - 12:00 AM Monday - Friday
	12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

While winter TOU periods might be defined as follows:

Time Of Use Period	Time Periods
On Peak	11:00 AM - 5:00 PM Monday - Friday
Off Peak	12:00 AM - 11:00 AM Monday - Friday
	5:00 PM - 12:00 AM Monday - Friday
	12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

This is an example of a "simple" mass-market time-of-use schedule. Commercial and industrial schedules can be much more complex.

Generally speaking, there are two different types of meters that can be installed at service points that participate in TOU rates:

Scalar meters with dedicated registers for each TOU period. For this type of meter, the scalar measurements are the TOU consumption and no special derivation is necessary. The assumption is that the TOU schedule is programmed into the device and no TOU Map is required within Oracle Utilities Meter Data Management.

Interval meters that measure how much is consumed each interval. For this type of meter, the interval measurements are used to derive the consumption in each TOU period based on the TOU Map defined in Oracle Utilities Meter Data Management.

For information on configuration, reference the related section for TOU Mapping in the Administration Guide.

About TOU Maps

Interval consumption is mapped into time-of-use periods using a TOU map. TOU maps define the TOU periods for each interval. TOU map data is structured in a format very similar to final measurement data, with specified intervals of time each designated to a TOU period (instead of a measurement value). For example, the table below lists a set of TOU map data and corresponding final measurements for a period of 4 hours on January 1, 2010.

Copyright © 2026, Oracle and/or its affiliates.

Table Title

TOU Map	Measuring Component		
Date/Time	**TOU Code**	**Date/Time**	**Consumption**
1-Jan-10 3:00pm	ONPEAK	1-Jan-10 3:00pm	14.678
1-Jan-10 4:00pm	ONPEAK	1-Jan-10 4:00pm	29.12
1-Jan-10 5:00pm	ONPEAK	1-Jan-10 5:00pm	12.12
1-Jan-10 6:00pm	OFFPEAK	1-Jan-10 6:00pm	17.16

Any type of final measurement can be mapped to a TOU period, including derived quantities. While the principal requirement for TOU Maps is to aggregate interval consumption into TOU periods for a Usage Transaction, the system has many zones in the user interface that display interval consumption using TOU periods.

Maintaining TOU Maps

This portal is used to display and maintain a TOU Map.

Refer to About TOU Maps for additional functional information.

You can access the portal from **Menu > Usage > TOU Map**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map**: Defines basic information about the TOU map, including dynamic options
- **TOU Map Data List**: Lists TOU map data created from the TOU map

The **Time Of Use Map Data Generation Monitor** batch (D1-TOUTR) is used to generate TOU Map data automatically based on the TOU Configuration. TOU Map data can also be generated manually through the TOU Map Portal in the UI.

Dynamic Options

- About Dynamic Options
- About Dynamic Option Events
- Maintaining Dynamic Options
- Maintaining Dynamic Option Events

About Dynamic Options

Dynamic Options specify terms that override how usage is normally calculated, such as a critical peak period that affects the TOU mapping of interval consumption.

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests or other demand response

Copyright © 2026, Oracle and/or its affiliates.

events. During these types of events, the TOU rules defined for a TOU map may be overridden. This is done through the use of dynamic options and dynamic option events.

Dynamic options allow flexibility for ad-hoc and special events that might impact how usage calculations are performed. For example, if a utility requests that its customers curtail their usage for a specific time period, a dynamic option and corresponding dynamic option event could be used to represent the curtailment request. During the time period of the curtailment request, usage might be calculated differently (with a different set of TOU periods, for example), which in turn might impact how the customer's bill is calculated.

> **Note**
>
> Dynamic Options can be created from Customer Care and Billing Contract Options via Customer To Meter Master Data Synchronization.

Dynamic Options and TOU Maps

To apply a dynamic option (and one or more of its related dynamic option events), you reference the dynamic option on a TOU map, along with a corresponding "dynamic" TOU map to be used during the dynamic option event.

For example, with Critical Peak Periods you might add a new TOU period (called "Critical Peak"), and create a new TOU map that is the same as your standard TOU map, but that also includes your Critical Peak TOU period. For example, in the summer your new set of TOU periods might be defined as follows:

Time Of Use Period	Time Periods
Critical Peak	10:00 AM - 2:00 PM Monday - Friday
On Peak	8:00 AM - 10:00 AM Monday - Friday 2:00 PM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday 9:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

If a dynamic option event occurs during a usage transaction period, the standard TOU map is overridden with the "dynamic" TOU map. In our example, a usage transaction period that includes August would use the dynamic TOU map when generating TOU map data for the time between 10:00 AM and 2:00 PM on August 2, but would use its standard TOU map for the rest of the month.

About Dynamic Option Events

Dynamic Option Events are specific periods of time during which a dynamic option is applicable.

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests or other demand response events. During these types of events, the TOU rules defined for a TOU map may be overridden. This is done through the use of dynamic options and dynamic option events.

A dynamic option may have many dynamic option events over time.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Dynamic Option Events can be created from Customer Care and Billing Contract Option Events via Customer To Meter Master Data Synchronization.

Maintaining Dynamic Options

This portal is used to display and maintain a Dynamic Option.

Refer to About Dynamic Options for additional functional information.

You can access the portal from **Menu > Usage > Dynamic Option**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Dynamic Option**: defines the basic information about a Dynamic Option.
- **Dynamic Option Event List**: provides a list of the Dynamic Option Events that have been created for this Dynamic Option.

> **Note**
>
> Dynamic Options can be created from Customer Care and Billing Contract Options via Customer To Meter Master Data Synchronization. Dynamic Options created from Contract Options should not be modified using this portal.

Maintaining Dynamic Option Events

This portal is used to display and maintain a Dynamic Option Event.

Refer to About Dynamic Option Events for additional functional information.

You can access the portal from **Menu > Usage > Dynamic Option Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zone may appear as part of the portal's **Main** tab page:

- **Dynamic Option Event**: defines the basic information about a Dynamic Option Event.

> **Note**
>
> Dynamic Option Events can be created from Customer Care and Billing Contract Option Events via Customer To Meter Master Data Synchronization. Dynamic Option Events created from Contract Option Events should not be modified using this portal.

Consumption Extract Requests

- About Consumption Extract Requests

Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Consumption Extract Requests

About Consumption Extract Requests

Consumption Extract Requests are created every time measurement or interval data is extracted for Oracle Utilities DataConnect or via Configurable Consumption Extracts. The consumption extract request can be created manually, typically to initiate an initial load type of extract, or created automatically, typically for an incremental type of extract, based on a configuration on the consumption extract type.

The Consumption Extract Request is responsible for submitting and monitoring the actual batch jobs that perform the data extract. In addition, it can be used to store any additional filters to further trim down data that will be extracted.

Refer to the Defining DataConnect Options section in the *Administrative User Guide* for more details on how consumption extract requests are used with DataConnect.

Refer to Configurable Consumption Extracts in the *Administrative User Guide* for more information about how they are used with configurable consumption extracts.

Maintaining Consumption Extract Requests

This portal is used to display and maintain a Consumption Extract Requests.

Refer to About Consumption Extract Requests for additional functional information.

You can access the portal from **Menu > Usage > Consumption Extract Requests**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Request**: displays the main attributes of a Consumption Extract Request.
- **Request Log**: displays all log information related to the Consumption Extract Request.

Contacts

- About Contacts
- Maintaining Contacts

About Contacts

Contacts are individuals or business entities with which a company has contact. Service Points can have associated contacts which define the individual or business entity that uses the service (electricity, gas, water, etc.) delivered at the Service Point. A contact is optional for Service Points, but Usage Subscriptions must reference contacts.

Note: The base-package name search on the 360 Degree Search looks for Usage Subscription-related contacts. Use the Service Point Query portal to find a Service Point using a Service Point-related contact.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In Oracle Utilities Customer To Meter, contacts are created from Customer Care and Billing Persons via Customer To Meter Master Data Synchronization.

Maintaining Contacts

This portal is used to display and maintain a Contact.

Refer to About Contacts for additional functional information.

You can access the portal from **Menu > Usage > Contact**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact**: defines the basic attributes of the contact
- **Service Points**: This zone lists service points associated with the contact.

360 Degree and Unified Search

- Using 360 Degree Search
- Using Unified Search

Using 360 Degree Search

The 360 Degree Search feature allows users to search for devices, measuring components, service points, and/or usage subscriptions using a variety of search criteria.

From the search results list, you can navigate to the 360 Degree View portal to view data on one of the following tabs:

- Measuring Component
- Device
- Service point
- Usage Subscription
- Contact

When accessed, the 360 Degree View opens on the tab corresponding to the type of object (device, measuring component, etc.) you select. For example, if you click the Measuring Component description, the Measuring Component tab opens.

Refer to 360 Degree Views for more details about 360 Degree View.

The following options are available for 360 Degree Search:

- **Search by Name:** Allows search by name/address OR contact identifier OR usage subscription identifier. To search by address, click Go To Search by Address. To search by device, click Device.

Copyright © 2026, Oracle and/or its affiliates.

- **Search by Address:** Allows search by address/name OR service point identifier. To search by name, click Go To Search by Name. To search by device, click Device.

  > **Note**
  >
  > "Search by Address" search results contain service points with one or more active usage subscriptions only.

- **Search by Device:** Allows search by device identifier OR measuring component identifier. To search by name, click Go To Search by Name. To search by address, click Address.

Using Unified Search

The Unified Search feature allows you use the toolbar search field to search for business entities as well as search for menu items. See Unified Search in theUser Interface Standard Features section for more information about the general use of the unified search field.

In Oracle Utilities Customer To Meter and Oracle Utilities Customer Cloud Service, the Unified Search field allows searching by a service point's address (Address), a contact (Name), an account ID (Account ID), a customer identifier (ID Number), contact information such as a phone number or email address (Contact Info) or a meter’s badge number (Badge Number) or serial number (Serial Number).

You can use the following filters to indicate your specific search criteria:

Filter	Filter Rules
Address	One or more numeric characters followed by space and an alphanumeric character
Name	Minimum of 5 characters excluding whitespaces
Account ID	10 numeric characters
ID Number	Social Security Number: 3 numeric characters followed by a dash, 2 numeric characters, a dash and 4 numeric characters
Contact Info	Phone Number: Plus character and 1-3 numeric characters (country code, optional) followed by a space, period, dash or open parentheses, 3 numeric characters (area code), followed by a space, period, dash or close parentheses, followed by 3 numeric characters (exchange code), followed by a space, period or dash, followed by 4 numeric characters (subscriber number). Email Address: Non-whitespace characters followed by an @ sign, non-whitespace characters, a period, non-whitespace characters
Badge Number	Minimum of 5 characters excluding whitespaces
Serial Number	Minimum of 5 characters excluding whitespaces

For example, to search for a service point located at 1401 Flower Street, you could enter “1401 Flower”.

To search explicitly using a specific filter, precede the filter value with one of the following predefined hints:

Copyright © 2026, Oracle and/or its affiliates.

Filter	Hint	Examples
Address	ad:	ad:1401 Flower
Name	n:	n:Smith, J
Account ID	ac:	ac:1234567890
ID Number	id:	id:123-45-6789
Contact Info	c:	c:555-2287 c:mbrazil@oracle.com
Badge Number	bn:	bn:12345
Serial Number	sn:	sn:6789

Examples:

- To search for a service point located at 1401 Flower Street, you could enter "ad:1401 Flower" (hint) or “1401 Flower” (no hint)
- To search for a person named John Smith, you could enter "n:Smith, J" (hint) or “Smith,J” (no hint)
- To search for a person whose phone number is 555-2287, you could enter "c:555-2287" (hint) or “555–2287” (no hint)
- To search for a meter whose badge number is 12345, you could enter "bn:12345" (hint) or “12345” (no hint)
- To search for a meter whose serial number is 6789, you could enter "sn:6789" (hint) or “6789” (no hint)

Clicking the **Advanced Search** link opens the **Customer 360 Search** portal and the **Customer 360 Search** zone.

Fore information about configuring the Unified Search for your implementation, refer to Understanding Unified Search in the *Administrative User Guide*.

Search Results

Each result in the search result list may include any of the following:

- Name (highlighted when using the Name filter) or “No Person Information”
- Address (highlighted when using the Address filter) or “No Premise Information”
- Account ID (highlighted when using the Account ID filter) or “No Account Information”
- ID number (highlighted when using the ID Number filter) or “No Contact Information”
- Contact Info (highlighted when using the Contact Info filter) or “No Contact Information”
- Measuring Component Info (highlighted when using the Badge Number or Serial Number filter)
- Filter Name and Search Criteria (always displayed)

To select an item from the search results, click the result.

360 Degree Views

- 360 View Measuring Component

Copyright © 2026, Oracle and/or its affiliates.

- 360 Degree View Device
- 360 Degree View Service Point
- 360 Degree View Usage Subscription
- 360 Degree View Contact

360 View Measuring Component

- Using 360 Degree View Measuring Component
- Applying Measurement Functions

Using 360 Degree View Measuring Component

This portal is used to view a broad set of information related to a Measuring Component.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Measuring Component has been selected you are brought to the 360 Degree View Measuring Component portal.

The following zones may appear as part of the portal's **Measuring Component** tab page:

- **Measuring Component Overview**: Displays the Service Point, device, Device Configuration, Install Event, and Usage Subscription for the selected device.
- **Device Configuration Overview**: Displays an overview of the measuring component's parent device's configurations over time, including the measuring components associated with each configuration, and Service Points (along with the relevant Install Events) at which the device's configurations have been installed over time, if any exist.
  - To create a new reading for a scalar measuring component associated to a Device Configuration, click the **New Reading** link in the zone title bar.
- **Interval - Initial Measurement Overlay**: Overlays initial interval measurements (either pre-VEE or post-VEE) against the final measurements for a given period in graphic and tabular formats.
- **Final Values Overlay**: Graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components that measuring different quantities, such as temperature.
- **Interval - Audit View**: Allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.
- **Time of Use By Day**: Displays daily TOU-mapped usage data for an interval measuring component based on a user- defined time period and TOU map.
- **Time of Use Overlay**: Displays an overlay of the TOU periods on a final interval measurement along with totalized TOU consumption based on a user-defined time period and TOU map.
- **Scalar - Initial Measurement Overlay**: Overlays initial scalar measurements (either pre-VEE or post-VEE) against the final measurements for a given period in graphic and tabular formats.

Copyright © 2026, Oracle and/or its affiliates.

- **Scalar - Audit View**: Allows users to view a scalar measurement for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the measurement and see how the measurements looked at different points in time.
- **Measurements**: Displays measurement details for the current measuring component in a tabular format. (split into two zones for v2.2)
- **Usage Subscriptions Directly Linked to MC**: Displays Usage Subscriptions that are directly linked to current measuring component.

Functions are available in many of these zones that allow you to interact with the measurement data and/or create new data. For more information of these functions refer to the Applying Measurement Functions section.

Applying Measurement Functions

Many of the Interval measurement zones on the Measuring Component tab of the 360 Degree View provide access to the following functions:

- **Create Scratchpad**: Creates a scratchpad measuring component along with new initial measurement data by copying all or some of the final measurements from the measuring component displayed in the zone.
- **Create / Override**: Creates new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements or create new measurement data for the initial measurement it creates.
- **Export to Excel**: Exports measurement data to a comma-separated values spreadsheet.
- **New Reading**: Creates a new interval initial measurement reading for the measuring component.
- **Rederive Values**: Rederives the measurement's values displayed in the zone.
- **Save As**: Creates new initial measurement data for a measuring component other than that being displayed by copying all or some of the final measurements shown in the zone.

Many of the Scalar measurement zones on the Measuring Component tab of the 360 Degree View provide access to the following functions:

- **Convert to Interval**: Converts scalar consumption into interval consumption. The converted consumption is held in a new scratchpad measuring component.
- **Create / Override**: Creates new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements or create new measurement data into the initial measurement it creates
- **Export to Excel**: Exports measurement data to a comma-separated values spreadsheet.
- **New Reading**: Creates a new scalar initial measurement reading for the measuring component.
- **Rederive Scalar Values**: Rederives the measurement's values displayed in the zone.

360 Degree View Device

- Using 360 Degree View Device

Copyright © 2026, Oracle and/or its affiliates.

Using 360 Degree View Device

This portal is used to view a broad set of information related to a Device.

You can access the portal by selecting **Main**, then **360 Search**, then [options]. You are provided with the option of searching by Name, Address, or Device. Once a Device has been selected you are brought to the 360 Degree View Device portal.

The following zones may appear as part of the portal's **Device** tab page:

- **Device Overview**: Displays the device, Service Point, Install Event, and Usage Subscription for the selected device.
- **Device Configuration Overview**: Displays an overview of the device's configurations over time, including the measuring components associated with each configuration, and Service Points (along with the relevant Install Events) at which the device's configurations have been installed.
- **Scalar Readings**: Displays a list of scalar readings for scalar measuring components associated with the current device and Device Configuration.
  - To create a new reading for scalar measuring components associated with the Device Configuration, click **New Reading**.
  - To edit an existing reading, click the edit icon for the reading you wish to edit. Editing a reading creates new initial measurements that are validated. If these edits do not pass validations, including VEE Rules, the changes are not saved.

360 Degree View Service Point

- Using 360 Degree View Service Point

Using 360 Degree View Service Point

This portal is used to view a broad set of information related to a Service Point.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Service Point has been selected you are brought to the 360 Degree View Service Point portal.

The following zones may appear as part of the portal's **Service Point** tab page:

- **Service Point Overview**: Displays the Service Point, device, Device Configuration, Install Event, and Usage Subscription for the selected Service Point.
- **Measuring Component Types**: Lists the measuring component types configured for the Service Point. To display consumption history for a measuring component, click the **Broadcast** icon next to the Measuring Component Type. Doing so will display either the Interval Consumption History or the Scalar Consumption History.
- **Interval Consumption History**: Displays a graph of interval consumption for a broadcast interval measuring component type.
- **Scalar Consumption History**: Displays a graph of scalar consumption for a broadcast scalar measuring component type.

Copyright © 2026, Oracle and/or its affiliates.

- **Timeline**: Displays a timeline of events for the Service Point, including usage transactions, activities, and VEE exceptions. Each event is depicted as a button containing the date on which the event occurred. To search for a specific month and year in the Timeline zone, select the month and year from the drop-down lists and click the **Search** icon. To navigate forward and back along the timeline, use the left and right arrows.
- **Service Issue Monitors Related to SP**: displays a list of service issue monitors and related service investigative orders related to the current Service Point, in reverse chronological order.

360 Degree View Usage Subscription

- Using 360 Degree View Usage Subscription

Using 360 Degree View Usage Subscription

This portal is used to view a broad set of information related to a Usage Subscription.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Usage Subscription has been selected you are brought to the 360 Degree View Usage Subscription portal.

The following zones may appear as part of the portal's **Usage Subscription** tab page:

- **Usage Subscription Overview**: Displays the Usage Subscription, main customer, Service Point, and direct channels for the selected Usage Subscription.
- **Usage Transaction History**: Lists usage transaction records for the Usage Subscription, including the start date/time, end date/time, description, and duration (in days) for each.
- **Timeline**: Displays a timeline of events for the Usage Subscription, including usage transactions, activities, and VEE exceptions. Each event is depicted as a button containing the date on which the event occurred. To search for a specific month and year in the Timeline zone, select the month and year from the drop-down lists and click the **Search** icon. To navigate forward and back along the timeline, use the left and right arrows.

360 Degree View Contact

- Using 360 Degree View Contact

Using 360 Degree View Contact

This portal is used to view a broad set of information related to a Contact.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Contact has been selected you are brought to the 360 Degree View Contact portal.

The following zones may appear as part of the portal's **Contact** tab page:

- **Contact Overview**: Displays contact information (name, phone number, email address, etc.) for the selected contact.
- **Contact's Service Points**: Lists the Service Points related to the contact, including the relationship type (main contact, main customer, etc.) for each.

Copyright © 2026, Oracle and/or its affiliates.

- **Contact's Usage Subscriptions**: Lists the Usage Subscriptions related to the contact, including the relationship type (main contact, main customer, etc.) for each.

Service Orders

In this section, we describe how to manage the fieldwork that takes place at your premise's service points.

- About Service Order Activities
- How Are Service Order Activities Created?
- About Service Order Field Activities
- What Happens After A Field Activity Is Completed?
- About Appointments
- Field Activity Maintenance
- Outage Calls
- Outage Event
- Customer Outage Information
- Outage Information
- Service Calls

About Service Order Activities

Service order activities represent specific requests to perform work at service points

These activities orchestrate the steps necessary to fulfill the request, based on the current state of the service point related to the request. For example, the system creates an "enable service" activity when a request is made to enable service at a particular service point. This activity examines the current state of the service point, and creates one or more additional activities if appropriate. For example:

- If there is no meter currently installed at the service point, the service order activity would create an "install meter" service order field activity and send it to a field work system (such as Oracle Utilities Mobile Workforce Management).
- If there is a meter installed, but the meter is not currently registered with the head end system, the service order activity might create a "commission meter" smart meter command.
- If the meter is active and commissioned, but not currently connected, the service order activity might create a "remote connect" smart meter command.

The base package provides the following types of service order orchestration activities:

- **Enable Service:** Used to enable service at a service point.
- **Disable Service:** Used to disable service at a service point.
- **Cut for Non-Payment:** Used to cut off service at a service point due to non-payment of past due amounts.

Copyright © 2026, Oracle and/or its affiliates.

- **Reconnect for Payment:** Used to restore service at a service point after receipt of past due payment.
- **Exchange Meter:** Used to orchestrate the exchange of meters at service point, such as in the event that a customer upgrades their meter.
- **Back-to-Back:** Used to orchestrate a change of service when the customer at a service point changes; such as, when owners/tenants change.

Service order orchestration activities are defined by the following:

- **Activity Type**: The service order's activity type.
- **Service Date/Time**: The date and time the service order request was received.
- **Service Point**: The service point associated with the service order request.
- **Service Agreements**: A list of one or more service agreements related to the service order request (not used with all service order orchestration activities).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Collection Data Details**: Details regarding past due balances (used with Cut for Non-Payment activities only).
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.

> **Note**
>
> Some service order activities use additional data.

In the event that an active service order activity needs to be updated and/or cancelled, the system also supports the following maintenance service order activities:

- Cancel Orchestration: Used to cancel a currently active service order activity.
- Update Orchestration: Used to update a currently active service order activity.

Service order maintenance activities are defined by the following:

- **Activity Type:** The service order's activity type.
- **Orchestration Activity:** The original orchestration activity related to the maintenance activity.
- **Related Activity:** A specific activity related to the maintenance activity; such as, a service order field activity or smart meter command.
- **Cancel Reason:** The reason for cancelling the original activity (for cancel activities only).
- **Comments/Instructions:** Specific comments and/or instructions applicable to the maintenance activity (for update activities only).
- **Start Date/Time:** The date and time the maintenance activity was initiated (for update activities only).
- **Request Information:** Details of the service order request, including requester and external system information.
- **Activity Hierarchy:** A tree that displays the activity and related activities (including service order field activities and smart meter commands), inbound and outbound communications, and completion events.

Copyright © 2026, Oracle and/or its affiliates.

How Are Service Order Activities Created?

Service order activities are typically created due to a field activity request as follows:

- Starting or stopping service indirectly causes field activity requests to be created. These field activity requests initiate the service orders to orchestrate the actions necessary to connect or disconnect service. Refer to Starting Service and Field Activities for more information about how field activity requests are created to start and stop service.

  > **Note**
  >
  > If you use the Order transaction to start service, field activity requests will also be created in the same manner described under the Starting Service and Field Activity Requests discussion

- The activation of a severance event with fieldwork causes field activity requests to be generated to disconnect or reconnect service. For more information about severance events and field activity requests, refer to The Big Picture Of Severance Events).

Service order activities can also be created due to a request received from an external system or created directly within Oracle Utilities Smart Grid Gateway.

About Service Order Field Activities

Service order field activities are specific types of tasks performed by field personnel. Examples include installing meters, repairing/replacing meters, etc.

Service order field activities are created when completing specific types of requests for service requires field work. Once created, service order field activities are sent to a field work system, where they are scheduled and assigned to a field work crew. When the task is completed by the crew, the field work system sends an response to the system and the service order field activity is updated as appropriate.

When communicating with field work systems, service order field activities create outbound communications to send the required information to the field work system. When the field work system sends its response, the system creates an inbound communication. See Communications for more information about communications. Service order field activities that have created outbound communications can also be manually completed or canceled if necessary.

Service order field activities are created when a service order orchestration activity examines the current state of the service point, and determines that field work is required to complete the service request. For example, if an "enable service" activity determines that there is no meter currently installed at the service point, the service order activity would create an "install meter" service order field activity and send it to a field work system.

Service order field activities are defined by the following:

- **Activity Type**: The service order field activity type.
- **Status**: The current status of the service order field activity.
- **Service Date/Time**: The date and time the service order field activity was created.

Copyright © 2026, Oracle and/or its affiliates.

- **Service Point**: The service point associated with the service order field activity.
- **Field Task Type**: The field task type for the service order field activity. The list of available field task types can be filtered based on the **Valid Field Task Types** defined for the service point's service point type via the **Restrict Field Task Type by Service Point Type** option on the Service Order Management Master Configuration.
- **Recipient**: The service provider to which the service order field activity is sent for scheduling and assignment.
- **Device**: The device related to the service order field activity (if applicable).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.
- **Address Information**: The address of the service point associated with the service order field activity.

What Happens After A Field Activity Is Completed?

When a service order activity completes, if there is a related field activity request this will also be completed. Completing a field activity request that is linked to a pending start service agreement, will also automatically activate the service agreement. Similar, completing a field activity request linked to pending stop service agreements will stop the associated service agreement.

After a field activity that is linked to a severance event is completed, the severance event becomes complete. This, in turn, triggers the activation of the next event.

NOT_SUPPORTED

For more information about field activities linked to pending starts, refer to Starting Service and Field Activities.

For more information about field activities linked to pending stops, refer to Stopping Service and Field Activities.

For more information about severance events and field activities, refer to The Big Picture of Severance Events.

About Appointments

The Appointment Booking portal is used to request and track appointments related to service order field activities that have been scheduled in an external field work scheduling system.

When service order field activities send outbound communications to the field work system, appointments are created in field work system so that the work can be assigned to a crew. Requests for appointments are most often requested automatically when the service order field activity is created, but if needed, users can request appointments manually.

Users work with the Appointment Query portal to search for activities and related appointments.

Copyright © 2026, Oracle and/or its affiliates.

Users work with the Appointment Booking to request and track appointments for service order field activities. This portal includes the following zones:

- **Service Point Activities**: Displays a list of activities and related appointments for the current service point in date descending order. This zone also includes a filter area which allows the user to define the inclusion or exclusion of activities according to certain criteria
- **Activity Summary**: Displays details of a selected activity in the Service Point Activities zone (indicated by a star).
- **Appointment Booking Results**: Displays a list of available appointment slots as returned by the scheduler. This zone allows the user to book an appointment by selecting a slot and subsequently updating the orchestration or field activity.

Field Activity Maintenance

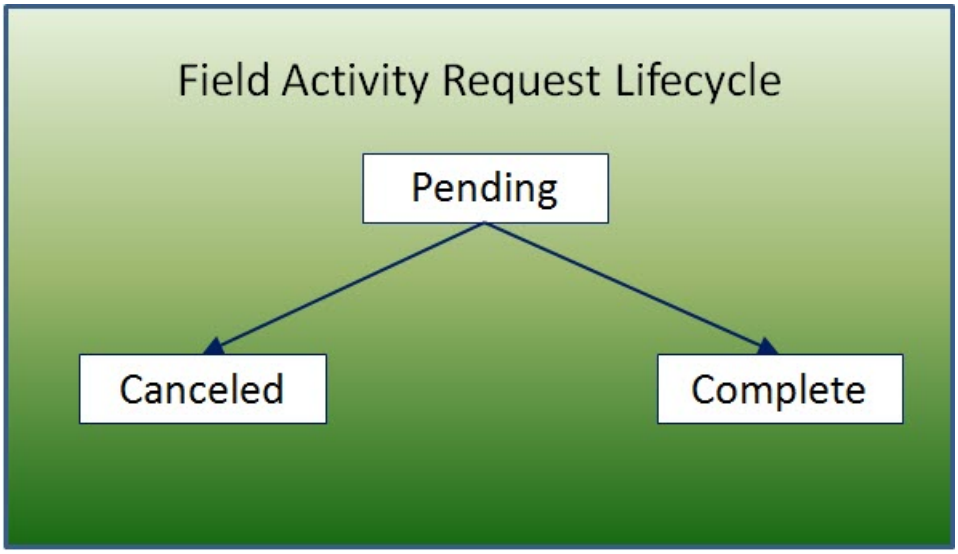
A field activity request is a request to perform actions at a service point. Examples of field activity requests include start service, stop service, and start/stop.

The topics in this section describe the pages on which field activity request-related information is maintained.

- Lifecycle of a Field Activity
- Field Activity - Main Information
- Field Activity - Steps
- Field Activity - Characteristics/Remarks
- Field Activity - Field Activity Portal
- Field Activity - Log

Lifecycle of a Field Activity

The following diagram illustrates the lifecycle of a field activity.



Pending: A field activity starts its life in the Pending state.

Copyright © 2026, Oracle and/or its affiliates.

Complete: After the related actions have been performed, the system is updated to reflect what happened and the field activity request becomes complete. No changes may be made to a Complete field activity request.

Canceled: A Pending field activity request may be Canceled.

Field Activity - Main Information

The Main page contains core field activity information.

Description of Page

The following fields are displayed on every page.

Field Activity is a concatenation of summary information regarding this field activity request.

Field Activity ID is the system-generated unique identifier for the field activity request.

The remaining fields are specific to the Main page.

Indicate the **Service Point ID** where the activity takes place. Information about the service point is displayed adjacent.

Use **Activity Type** to define the type of field activity request. Only those activity types defined as valid for the service point's SP type's field activity profile may be selected. The activity type's description is displayed adjacent.

Displayed just below the **Activity Type** is a system-generated message regarding how the field activity request was created. This message includes the date and time the field activity request was created and the process that created it. If a user manually creates a field activity request the user's ID is concatenated into the message.

Use **Schedule Date / Time** to define when the field activity is scheduled to be done.

Enter **Reschedule Reason** when changing the field activity's schedule date/time.

> **Note**
>
> **Reschedule Reason** only appears if the **Fieldwork Options** feature configuration exists with the **Reschedule Reason Required** option set to Y.

Field Activity Status defines the state of the field activity. The following values may exist:

- **Pending** A field activity is originally created in the Pending state.
- **Completed** When an activity's fieldwork is done the system is updated with the results and the activity becomes Completed.
- **Canceled** If an activity cannot be done, it should be Canceled.

To change the status of the field activity request, use the **FA Actions**, described below.

If you cancel a field activity, you must supply a **Cancel Reason Code**.

Field Activity Priority this field is not used when using Service Order Management.

Use **Comments** to describe anything unusual about the activity.

Copyright © 2026, Oracle and/or its affiliates.

If you have plugged in an FA Additional Information algorithm on the Installation record, **Additional Info** displays the results of the algorithm, if any.

The **FA Actions** buttons allow you to change the status of the field activity request.

- Click **Complete** to change the status of the field activity to complete. This button is only enabled if the status is pending.
- Click **Cancel** to change the status of the field activity to canceled. This button is only enabled if the status is pending.

> **Note**
>
> **Pend** and **Held** are not applicable for service order related field activity requests.

At the bottom of this page is a tree that shows the account(s) and service agreement(s) that are currently linked to the field activity request's service point. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained. Note that only field activities created by start/stop or severance create the link between the service point and service agreement.

Field Activity - Steps

The Steps page is not applicable for service order related field activity requests.

Field Activity - Characteristics/Remarks

Use this page to link additional information to the field activity requests.

Description of Page

Summary information about the **Service Point** where the activity will occur is displayed, as well as the service point's system-generated **Service Point ID**.

The **Characteristics** collection contains information that describes miscellaneous information about the field activity request.

To modify a characteristic, simply move to a field and change its value. To add a new characteristic, click the plus button to insert a row, then fill in the information for each field. The following fields display:

- **Characteristic Type** Indicates the type of characteristic.
- **Characteristic Value** Indicates the value of the characteristic.

> **Note**
>
> **Default Note.** A field activity's characteristics default from the field activity type.

Typically fieldwork related remarks are processed by service order field activities and if applicable a service task. In addition remarks can be added to the **Remarks** grid.

Copyright © 2026, Oracle and/or its affiliates.

The **Field Activity Remark Status** is only relevant if a **Field Activity Remark** code has algorithms associated with it. The execution of the algorithm is performed by the field activity remark activation a background process (FACT).

- When you add a remark that has an algorithm, the status is Pending . It remains in this state until the FACT process next runs.
- When the FACT process next runs, if the system is successful in executing the algorithm, the status of the field activity remark may become Complete. If the system is not successful in executing the algorithm, the status becomes Error.

Field Activity - Field Activity Portal

The Field Activity Portal page displays additional information associated with a field activity request.

Depending on your implementations requirements, you may define additional zones to appear on this Portal.

The contents of this section describe the zones that are available on this portal page.

General Information

Field Activity ID, **Field Activity Info** are displayed.

Service Order Management Information

This zone displays the related service order activities in Service Order Management.

- Clicking a service order activity's information navigates to the Service Order Management Activity Maintenance Portal.
- If the field activity requires appointment, formatted information, indicating that appointment must be booked, appears on top of the field activity tree. Clicking "Book Appointment Now", navigates to Service Order Management application's Appointment Booking Portal.

Field Activity Attachments

This zone provides the ability to add, edit, delete, upload, and view attachments related to field activity request.

- Clicking attachment information navigates to Attachment Maintenance Portal.

Field Activity - Log

Use this page to view and add log records for a field activity request.

Description of Page

The **Log** grid displays log entries that audit the progress of a field activity. You cannot modify or delete log entries that have been created by the system.

You can manually add a log entry of type User.

The Log grid displays the following information:

- **Date/Time** Displays the date and time when the log entry was created.
- **Log Type** Indicates the entry event for which the log entry was created. The log type for log entries created online is User.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The values for the log type are customizable using the Lookup table. This field name is FA_LOG_TYP_FLG.

Details Contains user-specified or system-generated information about the log entry.

User Displays the user who caused the log entry to be created.

Sequence Displays the sequence that makes up part of the unique identifier of this log record (together with the Field Activity ID).

Outage Calls

> **Note**
>
> The information in this section applies to the integration with the Oracle Utilities Network Management System (NMS).

You can use the Outage Call page to view trouble calls and outage information, and to send outage call reports to NMS for a specific service point. You can also use this page to schedule job site appointments. Open this page using **Menu**, and then **Communication**, and then **Outage Call**, and then **Search.**

Description of Page

The Outage Call Search zone is a standard multi-query portal used to search for and list existing Outage Call records. In the search results list, click the Outage Call link to view and maintain the information about the outage call.

- Outage Call Zone

Outage Call Zone

The outage call zone contains display-only information about the record.

Outage Event

This page displays information about an outage event reported from an external outage management system. Outage event notifications may be interfaced from any outage management system using the **Outage Events Notifications** web service.

You can access the portal from the **Main Menu**, and then **Field Work**, and then **Outage Event**. You are brought to a query portal with options for searching for a specific outage event. Once an outage event has been selected you are brought to the maintenance portal to view the selected record.

The following zones may appear as part of the portal's **Main** tab page.

Copyright © 2026, Oracle and/or its affiliates.

- **Outage Event.** This zone provides information about the selected outage event.
- **Outage Event Characteristics.** This zone lists additional details about the outage event in the form of characteristics . The zone is visible if such details exist.
- **Impacted Customers.** This zone lists pairs of account and service points impacted by the outage event.

Customer Outage Information

This page displays outage information for an account as reported from an external outage management system. Outage event notifications may be interfaced from any outage management system using the **Outage Events Notifications** web service.

The portal is used to provide additional information for outage insights when presented as part of the Customer 360 dashboard. You can also access the portal from the Account context menu .

The following zones may appear as part of the portal's **Main** tab page.

- **Outages.** This zone lists active and planned outage events impacting the account.
- **Recently Restored.** This zone lists recently restored outage events related to the account.
- **Outage Details.** This zone displays additional information about an outage event selected in the open or recently restored outage events.

Outage Information

> **Note**
>
> The information in this section applies to the integration with Oracle Utilities Network Management System (NMS).

The Outage Information portals provide a real-time interface with NMS to search for the following network management items:

- Job History
- Call History
- Planned Outages.

Open this page using **Menu**, and then **Communication**, and then **Outage Information.**

- Job History
- Call History
- Planned Outages

Job History

Copyright © 2026, Oracle and/or its affiliates.

You can use the Job History portal to retrieve current and historical outage and non-outage jobs from NMS by a customer service point or account ID, or proximity to a street intersection or street segment, or other identifiers such as a 911 incident identifier, caller name, or call phone number.

Description of Page

The Job History zone can be used to search for an existing Job History record. If the search results list contains records for a single service point, the service information for that service point and the job history results will be displayed. If the list contains multiple service point's records, a list of service points are displayed and the service information and job history results will be displayed for one of these service points.

Please see the zone's help text for information about this zone's fields.

Call History

You can use the Call History portal to retrieve trouble call history from NMS by a customer service point, or account id, or proximity to a street intersection or street segment, or other identifiers such as a 911 incident identifier, caller name or call phone number.

Description of Page

The Call History zone can be used to search for an existing Call History record. If the search results list contains records for a single service point, the service information for that service point and the call history results will be displayed. If the list contains multiple service point's records, a list of service points are displayed and the service information and call history results will be displayed for one of these service points.

Please see the zone's help text for information about this zone's fields.

Planned Outages

You can use the Planned Outages portal to retrieve planned outage information from NMS.

Description of Page

The Planned Outages zone can be used to search for current, future and past planned outage status and details from NMS for a particular service point. Planned outage information will only be available once a planned outage has been scheduled and the list of impacted customers for that planned outage has been generated.

Please see the zone's help text for information about this zone's fields.

Service Calls

> **Note**
>
> The information in this section applies to the integration with an asset management system, such as Oracle Utilities Work and Asset Management (WAM).

Copyright © 2026, Oracle and/or its affiliates.

The following sections describe the Service Call Process and Service Call Communication.

- The Service Call Process and Service Call Communication
- Maintaining Service Call Communications

The Service Call Process and Service Call Communication

Service Call Process

The service call process contains five steps or panels:

Service Call Search. This panel is used input the contact details of the service call.

At least one filter should be populated before search can be performed.

The result is a combination of the following:

- Service calls in the asset system retrieved via a real-time call.
- Local non-final service calls that were not sent to the asset system yet.

By default, the maximum number of search results is ***20***. This can be overridden by setting the override search limit in the service call communication type.

The service call information includes the service call number, issue description, required by date and call back indicator. Service call number is the service call's reference number in the asset management system.

> **Note**
>
> The details of a service call can be viewed using the **View Service Call Details** icon. This pops up a window with the details of the service call. Details of service calls in the asset management system are retrieved via real-time call and includes communication logs and related documents. For authorized users, hyperlink to the service call in the asset management system is available. Refer to Setting Up the System To Enable Asset Management System Integration for more details.

When a service call is selected, either **Add Comment** or **Cancel** action should be selected to continue.

Continuing without selecting a service call means a new service call.

Location. This panel is used to input or display the location of a new or existing service call, respectively. For an existing service call, the location details are displayed for information purposes.

When the process is initiated with a premise in context, the premise is displayed and can be used to populate the location elements and limit the service points that can be selected. If the current premise is used, the location elements are populated and disabled. The geographic type defined in the service call communication type is used to populate the latitude and longitude elements based on the premise's geographic data.

When service point is set, the location elements are populated with the corresponding elements of the service point's premise and are disabled. Clearing the service point will re-enable the location detail elements with their values retained. The geographic type defined in the service call communication type is used to populate the latitude and longitude elements based on the premise's geographic data.

Copyright © 2026, Oracle and/or its affiliates.

Country and address should be populated to continue.

Contact. This panel is used input the contact details of the service call.

Communication type is defaulted to the default communication type defined in the service call communication type.

When the process is initiated with a person in context, the person is displayed and can be used to populate the name and phone numbers. If the current person is used, the name and phone number elements are populated and disabled:

- First and last names are derived from the person's primary name. Note that installation option's name separator configuration applies.
- Primary phone is set to the person's primary phone contact
- Mobile and home phones are set to the person's phone contact where the contact type is the mobile and home phone contact types defined in the service call communication type, respectively.

Contact details are not required to continue. **For Add Comment** or **Cancel** actions, either First Name or Last Name or one of the phone numbers must be provided.

Details. This panel is used to capture the issue details of the service call.

For a new service call, enter the details of the issue. Questions linked to the selected service code are presented. Set the resolved switch if based on the answers to the questions, the issue is resolved. Doing so will prevent a message from being sent to the asset system.

Description is required to continue. Service Code may also be required if configured as such in the service call communication type.

For **Add Comment** or **Cancel** action, details of the original service call are displayed for informational purposes.

Review and Submit. This panel provides an opportunity to review the information before submission.

When the service call process is finished, the service call information is copied to a new service call communication record and the service call communication ID is displayed on the Review and Submit panel. If there are questions linked to the service code, the questions and answers are copied to a new response record. The response is linked to the service call communication. The service call process is completed and service call communication takes over.

Service Call Communication

Service call communications are used to and process service call information gathered from interactions with the customer, the public or other external entity. These interactions can be reporting a new issue/request, adding comments to an existing issue/request or cancelation (issue is resolved or request no longer needed). The following service call information can be gathered:

- **Issue/Request Details**. This includes description, service category, service code, call back requirement, require by date, etc.
- **Location**. This is the location of the service call. This can be either a known service point or a freeform address

Copyright © 2026, Oracle and/or its affiliates.

- **Contact**. This is the contact information the interaction is with. This can be a known customer or freeform contact information. This is information is optional to allow anonymous interactions.

A service call communication record is created for each interaction.

- Determine if a service call information needs to be sent to the asset management system. If the service call process is canceled or if the issue was resolved during communication, the asset management system is not notified.
- Ensure that service call information is sent out to the asset management system in the correct sequence. New service call communication is sent out first, followed by additional comments and finally, cancelation.
- Send a service call communication outbound message to the asset management system. This is an asynchronous interface between Oracle Utilities Customer Care and Billing and the asset management system. Because this asynchronous, there should be no processing after the message is sent out in the same transaction. This is to avoid the case where the transaction is rolled back due to some error, but the message is already sent to the external system. Refer to Setting Up the System To Enable Asset Management System Integration for more details.
- Once the outbound message is sent, the service call communication will transition and stay in the Sent state until an acknowledgement is received from the asset management system. If a positive acknowledgment is received, the service call communication is completed. If a negative acknowledgment is received, the service call communication is transitioned to the Error state. A service call communication in Error state can be manually canceled or re-sent.

Service Call Communication Acknowledgement

The inbound web service ***C1-UpdateServiceCallCommunication*** is used to process a service call communication acknowledgement message. The message contains communication ID, service call ID, service call number and error information. This service stores this information into the service call communication. If the message indicates an error, the service call communication is transitioned to the Error state. Otherwise, the service call communication is completed.

Maintaining Service Call Communications

A service call communications is used to process service call information gathered from interactions with the customer, the public or other external entity. These interactions can be reporting a new issue/request, adding comments to an existing issue/request or cancelation (issue is resolved or request no longer needed). Refer to Service Calls for background information on service calls.

To view and maintain service call communications, open Menu > Field Order > Service Call Communication > Search . You are brought to a query portal with options for searching for service call communications.

Once a service call communication record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are displayed:

- **Service Call Communication** zone shows the details of the service call .
- **Questions** zone shows the related questions and answers to the service call communication. This information is stored in the ***Response*** object.

Refer to the zones' embedded help for more information.

Copyright © 2026, Oracle and/or its affiliates.

Billing

In this section, we describe how to manage your customer's bills.

- The Big Picture of Billing
- Maintaining Bills
- Maintaining Bill Segments
- Off Cycle Bill Generator
- Multi Cancel/Rebill
- SA Billing History
- Bill Exception
- Bill Segment Exception
- Maintaining Billable Charges
- Uploading Billable Charges
- Budget Review

The Big Picture of Billing

A bill is used to communicate changes in an account's financial obligations to the customer. Over time, a customer receives many bills. The topics in this section provide background information about a variety of billing topics.

> **Warning**
>
> Some organizations are responsible for presenting 3 rd party charges on their bills. Others are responsible for sending their charges to a 3 rd party for presentation on their bills. If this describes your organization, refer The Big Picture of SA Relationships for a description of how this functionality is implemented.

- An Illustration Of A Simple Bill
- An Overview of the Bill Creation Process
- Bill Errors
- Cancel / Rebill Incorrect Bill Segments
- Credit Notes - Presenting Cancellation Details on a Separate Bill
- Correction Notes - Presenting Correction Details on a Separate Bill
- How the System Requests How Much Was Consumed
- How To Override The Commodity Amount In A Bill Segment
- How Rates Affect the Information on Bill Segments

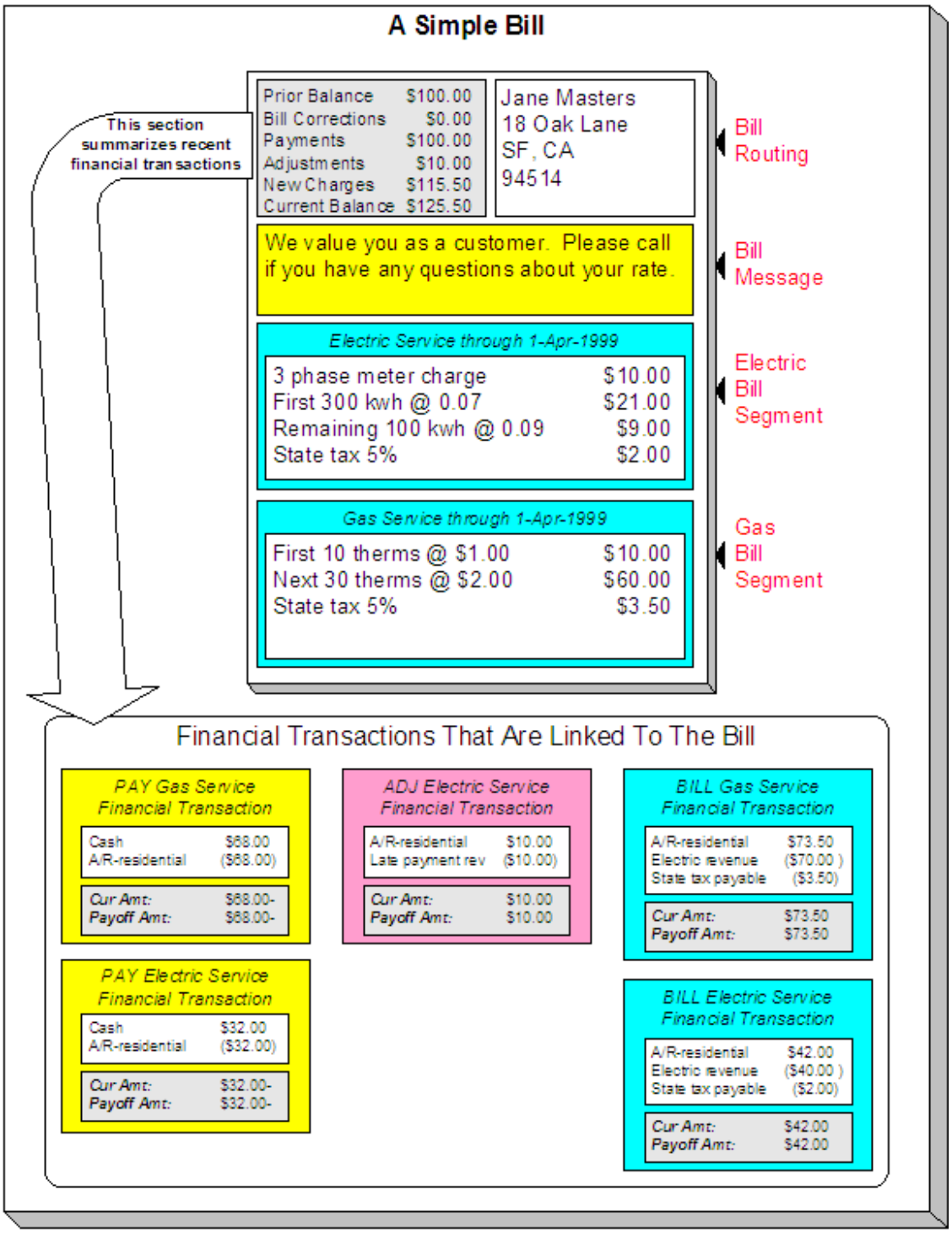
Copyright © 2026, Oracle and/or its affiliates.

- Bill Frequency - Bill Cycle vs Bill Segment Duration
- Prorating Charges When a Rate is Applied
- Batch Billing
- Billing Financial Transaction Considerations
- The Source Of Bill Routing Information
- Bill Messages
- A Bill May Affect More Than Just Customer Balances
- Budget Billing
- Net Energy Metering
- Using Billable Charges For Pass Through / Convergent Billing
- Printing Bills
- Sequential Bill Numbers
- Document Numbers
- Digital Signatures
- Writing Off Bills
- Idiosyncratic Manual Bill Cancellation

An Illustration Of A Simple Bill

The following diagram illustrates a simple bill:

Copyright © 2026, Oracle and/or its affiliates.



The following concepts are illustrated above:

A bill is produced for an account A bill is produced for an account. Over time, an account receives many bills.

A bill summarizes financial transactions A bill contains information about the various financial transactions that have taken place since the last bill was produced (i.e., payments, adjustments, and bill corrections). The above illustration shows a bill with financial transactions for new charges, a payment, and an adjustment.

Copyright © 2026, Oracle and/or its affiliates.

A bill is routed to persons A copy of the bill is sent to every person linked to the account who requires a copy of the bill.

A bill contains messages A bill may contain messages.

A bill typically contains bill segments A bill typically contains one bill segment for every active service agreement linked to its account.

A bill segment contains calculation details A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

An Overview of the Bill Creation Process

When the system is asked to produce a bill for an account, it attempts to create one or more bill segments for every non-cancelled / non-closed service agreement linked to the account. Whether or not a service agreement contributes bill segment(s) to a bill is a complicated subject as the system supports a wide variety of bill segment creation methods. For example, a service agreement that charges for metered service is produced as follows:

- The system first requests for the amount or usage consumed (or billing determinants) from Oracle Utilities Meter Data Management; for example, the number of lamps, the number of gallons of water, or the peak kW. For more information, refer to How the System Requests How Much Was Consumed.
- Next, the system applies the service agreement's rate to the amount consumed in order to calculate how much the customer owes. A bill segment and a financial transaction are generated to reflect the results of this calculation. The details of the calculations are captured in the bill calculation lines. For more information, refer to How Rates Affect The Information On Bill Segments for the details. Refer to Effective Dates and Price Proration for information about how the system prorates changes to rates and prices during a bill period.

> **Note**
>
> **There are many bill segment creation algorithms.** The above points explain how a bill segment is produced for a bill determinants required service agreement. It's important to be aware that the system has other bill segment creation algorithms. For example, if the service agreement charges for a deposit, the system neither amalgamates consumption nor does it apply a rate. You may be wondering how the system knows which method to use. It uses the bill segment type on the service agreement's SA type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

If errors are detected during the bill segment creation process, the bill segment is saved with its error. The system then proceeds to the account's next service agreement. This way, a user can see all problematic bill segments so they can be corrected en masse. Refer to Bill Errors for information about how to deal with bill errors.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Interdependent service agreements.** Your organization may have some service agreements whose charges must be calculated in a specific order. For example, you might have a service agreement whose bill segments are calculated using charges from a different bill segment. If this describes your organization, be aware that the system creates the bill segments in the order of the billing processing sequence on each service agreement's SA type.

When every segment on a bill is error free, the bill is ready to be completed. When a bill is completed, the system:

- Creates a bill routing for each person linked to the account who requires a copy of the bill. The routing information controls the format of the printed bill and how the bill is sent to the person. Refer to The Source Of Bill Routing Information for the details.
- Links all bill messages to the bill. A bill message can come from a variety of sources. Refer to The Source Of Bill Messages for the details.
- If the Freeze At Bill Completion installation option has been turned on:
  - All freezable bill segments will be frozen.
  - All freezable adjustments whose adjustment type indicates Freeze At Bill Completion will be frozen.
- Sweeps recent financial transactions that have been created since the last bill was completed. Refer to Financial Transactions Created Between Bills for information about how payments, adjustments and bill corrections appear on a bill.
- There are several other functions that happen at completion time. Refer to the description of the Complete button under Bill Lifecycle for the details.

And that's it; the bill can now be sent to the customer.

The remaining topics in this section provide more information about the creation and completion of bills.

> **Note**
>
> **Batch and real-time bill creation.** Anything the batch bill process does for whole sets of accounts, you can do to a specific account on-line / real time. Refer to How To Create A Bill For All Service Agreements Linked To An Account for information about how to create a bill on-line / real time. Refer to Batch Billing for more information about the batch bill creation process.

Bill Errors

The topics in this section describe errors that are detected when the system creates a bill.

- Bill Segment Errors
- Bill Completion Errors

Copyright © 2026, Oracle and/or its affiliates.

Bill Segment Errors

If you consider the huge number of variables involved in the creation of a bill segment, it probably won't surprise you to learn that occasionally, some bill segments are in error.

Errors tend to be caused by missing or inconsistent data. Some examples of classic errors:

- **Missing meter reads.** For example, if the system cannot find a relevant read (and it's not allowed to estimate) for a metered service point, it generates an error.
- **Missing master data.** For example, if a bill is supposed to be routed to the account's mailing address and the account doesn't have such an address, an error is generated.
- **Missing rate data.** For example, if a rate contains a customer specific charge and there is no value defined on the customer's service agreement, an error is generated.

The system saves bill segments that are in error just as it saves bill segments that are error-free. This is done because bill segments are nothing more than a snapshot of the data that was used to calculate the charges. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively fix the cause of the error. The standard way to fix an error is to:

- Look at the bill segment to determine the cause of the error.
- Correct the cause of the error.
- Regenerate the bill segment. Regeneration simply deletes the offending segment and recreates a new segment using the corrected information.

For every bill segment in error, a record is written to the Bill Segment Exception table.

NOT_SUPPORTED

Refer to How To Correct A Bill Segment That's In Error for instructions describing how to correct a bill segment.

It's obvious but worth stressing that a bill may contain some segments that are perfect and others that are in error. Once all segments that are in error are corrected, the bill can be completed (and sent to the customer).

If the bill segment in error was created as part of the batch bill process, the system attempts to fix the offending segment(s) each night during the account's bill window by regenerating it. Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And once a bill is error-free, it will be completed and sent to the customer.

NOT_SUPPORTED

Refer to Batch Billing for more information.

Copyright © 2026, Oracle and/or its affiliates.

Bill Completion Errors

In addition to errors on bill segments, there may also be errors detected when the system attempts to complete a bill. For example, if the system cannot find a mailing address, the bill will be in error (as opposed to one of its bill segments).

Errors tend to be caused by missing or inconsistent data. Some examples of classic errors:

- **Missing mailing address.** For example, if the system cannot find an address to which the bill can be routed, it generates an error.
- **Bill segments are in error.** If a bill contains bill segments that are in error, it cannot be completed and an error is generated.

The system saves bills that are in error just as it saves bills that are error-free. This is done because bills are nothing more than a snapshot of the data that was used to calculate the charges. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively fix the cause of the error. The standard way to fix an error is to:

- Look at the bill to determine the cause of the error.
- Correct the cause of the error.
- Recomplete the bill.

For every bill in error, a record is written to the Bill Exception table.

NOT_SUPPORTED

Refer to How To Correct A Bill That's In Error for more information.

If the bill completion error was created as part of the batch bill process, the system attempts to recomplete the bill each night during the account's bill window. Therefore, if the cause of the error is fixed during the day, the system will automatically complete the bill when batch billing next runs; you don't have to manually complete each bill.

NOT_SUPPORTED

Refer to Batch Billing for more information.

Cancel / Rebill Incorrect Bill Segments

Sometimes the error on a bill segment is not detected by the system. Such errors occur when the data used to calculate the bill is valid, but wrong. For example, the dial reading on the meter read is incorrect due to slipped fingers, or the sales tax percent was entered incorrectly on a bill factor.

There is no way the system can detect such problems and therefore the system freezes the bill segment and routes the bill to the customer. To correct such a bill segment, you must cancel

Copyright © 2026, Oracle and/or its affiliates.

the offending segment and create a new segment (after correcting the cause of the problem). We refer to this process as cancel / rebill.

NOT_SUPPORTED

Refer to How To Cancel / Rebill A Bill Segment for more information about how to cancel / rebill a frozen bill segment.

You must use the cancel / rebill process to correct problems on frozen bill segments. This is because a frozen bill segment can be thought of as having been posted to your general ledger (even if the GL interface hasn't run). And once a financial transaction is posted to the general ledger, it can only be removed via a reversal. Contrast this to bill segments that are in error - they can be deleted because they were never posted to the general ledger.

NOT_SUPPORTED

Refer to Bill Segment Lifecycle for more information about the differences between bill segments that are frozen and those that are in error.

Note

Bill segments are canceled / rebilled. It's important to stress that bill segments are canceled and rebilled (as opposed to bills). If every bill segment on a bill is incorrect, you must cancel / rebill each individual bill segment.

Using the cancel or cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, then you will need to use Credit Notes. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information. If your organization requires cancellation and rebill details to be presented to the customer separately from the new bill, then you will need to use Correction Notes. Refer to Correction Notes - Presenting Correction Details on a Separate Bill for more information.

Credit Notes - Presenting Cancellation Details on a Separate Bill

The system's Cancel / Rebill functionality will include cancellation and rebilling details on the customer's next bill. Some organizations are required to present cancellation details to the customer separately and not to commingle them with a new bill. This separate statement containing only the cancellation information is referred to as a Credit Note.

Credit Notes may be created for an individual bill segment or for all the bill segments linked to a bill. To create a credit note, a Credit Note button is available on the Bill Segment page and on the Bill page.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to How To Create A Credit Note for more information about how to create credit notes for a bill or bill segment.

The topics in this section provide describe the impact of this functionality.

- Credit Notes Are Only Possible if the Installation Option Is Turned On
- Credit / Correction Notes and Auto Pay
- Adjustments On Credit Notes
- Credit Note Routing Suppression

Credit Notes Are Only Possible if the Installation Option Is Turned On

The Credit Notes button on the bill segment and bill pages will only be available if you have selected the Credit Note Bill Correction option on the Installation record.

Credit / Correction Notes and Auto Pay

If a credit or correction note is being created for a customer who is enrolled in Auto Pay, note the following functionality:

When creating a credit or correction note for a bill (and all its bill segments):

- The system will cancel an automatic payment behind-the-scenes if a credit or correction note is issued BEFORE the automatic payment is interfaced to the financial institution. When you recomplete the bill, the system will create a new automatic payment that reflects the new amount due (and the canceled automatic payment will net out the original automatic payment). This is standard auto pay logic. Refer to How Are Automatic Payments Cancelled? for more information.
- If the customer has automatic payments that have already been interfaced to the financial institution, the creation of the credit or correction note will have no affect on any automatic payment. A warning will be issued in this case.

When creating a credit or correction note for a bill segment:

- Because automatic payments are for an entire bill, creating a credit or correction note at the bill segment level will not affect any automatic payments. A warning is issued in this case.

NOT_SUPPORTED

For more information about correction notes, see Correction Notes - Presenting Correction Details on a Separate Bill.

Copyright © 2026, Oracle and/or its affiliates.

Adjustments On Credit Notes

Above, we described how a credit note is a separate document that contains the cancellation details of the various bill segments that appeared on a given bill. Some implementations also want the adjustments that appeared on the original bill to be cancelled when a credit note is produced. Then, when the next bill is produced for the customer, they want these same adjustments reproduced. The following points describe how to implement this functionality:

- Plug in a Bill Completion algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-RADJ) will reverse a bill's adjustments when a credit note is completed.
- Plug in a Post Bill Completion algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-ADNB) will recreate the adjustments on the original bill after the credit note is completed. These adjustments will be swept onto the next bill produced for the account.

You can restrict the above functionality to specific adjustment types (for whatever reason). You do this by putting a given char type / char value on the adjustment types that should be reversed and recreated when a credit note is created. You define this char type / char value combination as parameters of the two algorithms described above.

> **Note**
>
> **When the original bill contains adjustment cancellations.** The above discussion explained how adjustments can be canceled on a credit note and then recreated on the next bill. These same algorithms also do the opposite, i.e., if the original bill contains an adjustment cancellation, the credit note will contain the recreated adjustment, and the next bill will contain its cancellation. Without this, the credit note would not be the reversal of the bill (and this is how some organizations operate).

Credit Note Routing Suppression

Some implementations do not send credit notes to their customers. To implement this functionality, plug in a Bill Completion algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-BRT) will cause the routing information to be suppressed when a credit note is completed.

Correction Notes - Presenting Correction Details on a Separate Bill

The system's Cancel / Rebill functionality will include cancellation and rebilling details on the customer's next bill. Some organizations are required to present cancellation and rebilling details to the customer separately and not to commingle them with a new bill. This separate statement containing the cancellation and rebill information is referred to as a Correction Note.

Correction notes may be created for an individual bill segment or for all the bill segments linked to a bill. To create a correction note, a **Correction Note** button is available on the Bill Segment and Bill pages.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to How To Create A Correction Note for more information about how to create correction notes for a bill or bill segment.

The topics in this section describe correction notes.

- Correction Notes Are Only Possible if the Installation Option Is Turned On
- Correction Notes Versus Credit Notes
- Correction Notes and Auto Pay

Correction Notes Are Only Possible if the Installation Option Is Turned On

The **Correction Note** button on the bill segment and bill pages will only be available if you have selected the Correction Note Bill Correction option on the Installation record.

NOT_SUPPORTED

If your implementation uses Correction Notes, it is recommended to also use the Freeze At Bill Completion installation option. For more information, see Installation Options - Billing.

Correction Notes Versus Credit Notes

Correction notes are a mixture of Credit Note and Cancel / Rebill functionality. However, unlike credit notes, correction notes are like regular bills and are subject to payment.

When you issue a correction note for a bill, the system cancels and rebills the bill segments for that bill. The new bill (the correction note) contains both the cancelled bill segments and the rebilled bill segments.

Similar to credit notes, the new bill (or correction note) is stamped with the ID of the original bill. After you review the bill segments, you can complete the correction note. During completion, the rebilled bill segments and the cancelled financial transactions (of the original bill's bill segments) are swept onto the correction note.

Correction Notes and Auto Pay

NOT_SUPPORTED

Refer to Credit / Correction Notes and Auto Pay for a description of how the system handles correction notes for customers using auto pay.

Copyright © 2026, Oracle and/or its affiliates.

How the System Requests How Much Was Consumed

There are two types of service agreements:

- Those that charge for some commodity; such as, electricity, water, garbage, and street lighting.
- Those that charge for something intangible; such as, deposit, charitable contribution, and payment arrangement.

For service agreements that charge for something intangible, there is no notion of requesting or determining how much was consumed and therefore the topics in this section are not relevant. For service agreements that charge for a commodity, the first step in the creation of a bill segment is requesting the amount or usage consumed (or bill determinants) from Oracle Utilities Meter Data Management; for example,

- If the service agreement bills for electricity, the system must know how much electricity was consumed.
- If the service agreement bills for lamp service, the system must know how many lamps the customer uses.

After the system requests and receives the amount or usage consumed (or bill determinants) during the bill period from Oracle Utilities Meter Data Management, it saves the consumption or usage details on the bill segment in the following snapshots:

- The **item snapshot** contains information about the items that existed at the service points received from Oracle Utilities Meter Data Management
- The **calculation/audit read details** contains information about the meter read details received from Oracle Utilities Meter Data Management
- The **calculation/audit usage details** contains information about the usage details received from Oracle Utilities Meter Data Management.
- The **service quantity snapshot** contains the billable service quantities received from Oracle Utilities Meter Data Management

> **Note**
>
> **This logic is in a plug-in.** The logic described below resides in algorithms that are plugged-in on the bill segment type referenced on a service agreement's SA type. If you need different logic, you should develop the appropriate algorithm and plug it in on your bill segment types.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Usage Requests for more information about how Oracle Utilities Customer Care and Billing requests and receives consumption / usage information from Oracle Utilities Meter Data Management.
>
> Refer to Usage for more information about how Oracle Utilities Meter Data Management calculates usage.

Copyright © 2026, Oracle and/or its affiliates.

- Item Snapshot
- Service Quantity Snapshot
- Calculation / Audit Read Details
- Calculation / Audit Usage Details

Item Snapshot

Items (such as, lamps, hydrants, and traffic lights) are not read so there are no meter reads to access.

- There are two sources of items:
  - Item-based service points have a single, **badged** item that may be installed at any instant in time. These types of service points are used, for example, when your organization keeps track of the installation history of specific lamps. If an item is installed at an item-based service point during the bill period, it will be included in the item snapshot.
  - All types of service points (metered, item-based, and non-badged) may contain one or more **unbadged** items. Multiple items exist at a service point, for example, when your organization chooses to record multiple lamps on a single service point rather than creating a separate service point for each lamp (this might be useful for a parking lot at a shopping mall). Each item referenced in the service point's collection of unbadged items is included in the snapshot.

NOT_SUPPORTED

Refer to Understanding Service Point Types for more information about service points with multiple items.

Service Quantity Snapshot

For service agreements where the meter reads are maintained in Oracle Utilities Meter Data Management and bill determinants are requested via a usage request, the service quantity (SQ) snapshot is the total amount consumed (or peak amount) for each unit of measure referenced on the bill segment's usage request.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Pre-Processing Calculation Rules.** If the service agreement's rate has pre-processing calculation rules, they may alter the amount of measured consumption or add additional units of measure to the SQ snapshot. Refer to All Calculation Rule-Based Rates Share a Common Structure for more information about pre-processing calculation groups.
>
> **Lamps and estimated consumption.** If any of the items linked to the service agreement's service points indicate the system should estimate consumption (e.g., lamps), additional entries will be added to the SQ snapshot for each such item. Refer to Device Types for more information.
>
> Meter read details used for interim calculations are ignored when producing the service quantity snapshot.

Calculation / Audit Read Details

We use the term "calculation / audit read" to describe a meter read that appears on a bill segment but for which no "real" meter read exists. The following points provide examples of when calculation / audit reads may be required:

- Some retail companies upload meter read details when they upload pass through charges. They do this in order to print the read details that were used to calculate the pass-through charges. It should be stressed that these reads are not used to derive billable consumption (because the charges have already been calculated). These reads are only uploaded for printing purposes and therefore "real" meter reads are not created. We categorize these reads as "calculation / audit reads". Refer to Billable Charge Upload - Read Details for more information about uploading meter reads with pass-through charges.
- You may have interval billing rates where you're required to print the intervals that contain, for example, the intervals with the three highest peak charges. To do this, you setup a "calculation only" calculation rule to find these intervals and create a "calculation / audit read" for each such interval. Refer to the RCCL-CVTOSC algorithm for an example of such an algorithm that would be plugged-in on a "calculation only" calculation rule.
- You may have interval billing rates that map interval consumption into time-of-use consumption. If you use a base-package algorithm to do this (refer to the RCTMPS and RCTMS algorithms for examples), a "calculation / audit read" is created for each resultant time-of-use quantity. These algorithms do this in case you need this information for bill print purposes or in case you have multiple bill lines that need to use the same mapped consumption.
- You may have situations where a meter has been temporarily removed from a service point but the business practice dictates that usage for the service point should be calculated during billing.
- When integrating with an MDM, meter reads are stored outside of CCB and usage calculation requests are sent to MDM during billing. Reads used in the MDM usage calculation might be returned to CCB as part of the usage response. These reads are used for printing purposes and therefore "real" meter reads are not created in CCB. Instead these reads are captured as "calculation / audit reads". Refer to Usage Requests for more information.

Copyright © 2026, Oracle and/or its affiliates.

Both real and calculation / audit reads are stored in a bill segment's read details (the name of the table is CI_BSEG_READ). The following points describe the differences between these two types of rows on this table:

- Calculation / audit reads may or may not reference a service point.
- Calculation / audit reads do not reference the prime key of a real read.
- Calculation / audit reads have a usage flag with a value of Calculation .
- Calculation / audit reads do not require a UOM code. If a UOM is not supplied, an SQI must be supplied.

If you encounter other scenarios during your implementation that necessitate calculation / audit reads, you can develop bill segment type and calculation rule algorithms to add additional calculation / audit reads to a bill segment's read details.

In order to avoid confusing end-users, we do not show calculation / audit reads on Bill Segment - Read Details (this page only shows "real" reads). Rather, calculation / audit reads are shown in a separate section on Bill Segment - SQ Details.

Calculation / Audit Usage Details

We use the term "calculation / audit usage" to describe service quantities retrieved from the bill segment's usage request. An example of when calculation / audit usage may be required:

- When usage requests are sent during billing to Oracle Utilities Meter Data Management, usage details may be returned to Oracle Utilities Customer Care and Billing as part of the usage request response. Oracle Utilities Meter Data Management might return both usage details and meter reads to print on the bill.

Usage service quantities are also stored in a bill segment's read details (the name of the table is CI_BSEG_READ). The following points describe the main features of calculation / audit usage:

- Calculation / audit usage may or may not reference a service point.
- Calculation / audit usage does not reference the prime key of a real read.
- Calculation / audit usage has a usage flag with a value of Service Quantity.
- Calculation / audit usage does not require a UOM code. If a UOM is not supplied, an SQI must be supplied.

If you encounter other scenarios during your implementation that necessitate calculation / audit usage, you can develop bill segment type and calculation rule algorithms to add additional calculation / audit usage to a bill segment's read details.

Calculation / audit usage details are shown in a separate section on Bill Segment - SQ Details.

How To Override The Commodity Amount In A Bill Segment

As described earlier, the calculation of the amount consumed is one of the first steps that takes place when a bill segment is created. 99.9% of the time, the system does a perfect job in this effort. However, there are rare circumstances when, due to missing or incorrect data, the quantity calculated by the system is not what you want to charge the customer for (consider what the consumption should be if the meter is broken?).

Copyright © 2026, Oracle and/or its affiliates.

To help you deal with these rare situations, a "back door" exists that allows a bill segment's billable quantities to be overridden. This truly is a "back door" because the proper way to fix the amount of a commodity is to correct the cause of the error and then cancel / rebill the bill segment.

How Rates Affect the Information on Bill Segments

After consumption has been compiled, the system applies the service agreement's rate to the consumption to determine how much to charge the customer. We refer to this process as the "rate application."

The rate application is a very sophisticated process and can affect every aspect of a bill segment. See Rate Schedule Execution for an overview of how the system processes a rate.

Rates not only control which charges appear on the bill, but also the verbiage. In addition, there are variables that can be used in the configuration to customize the message that the customer receives. See How to Use Description on Bill for more information

> **Note**
>
> **Not all service agreements use a rate!** There are a small number of service agreements that don't use a rate to calculate the charges that appear on their bill segments. For example, a service agreement used to levy a customer's payment arrangement amount doesn't need a rate because the amount to charge is defined on the service agreement. Whether or not a service agreement needs a rate is defined by the service agreement's SA type. Whether or not the system calls the service agreement's rate to calculate charges is controlled by the bill calculation algorithm plugged-in on the bill segment type that's referenced on the service agreement's SA type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

Proration. A rate contains a variety of effective-dated information (meaning that the prices are effective-dated, and the structure of the rate is effective-dated) If this effective-dated information changes during a bill period, the system may need to prorate the charges; for example, if the sales tax percentage changes mid-period, the system can prorate the tax change. Refer to segments. Refer to Effective Dates and Price Proration for more information.

There are many plug-in spots available on a calculation rule. These plug-ins can manipulate virtually every aspect of the bill segment. This means if you require functionality that isn't support by the base package calculation rule, you can build additional calculation rule plug-ins to do whatever you need. See Base Package Calculation Rule Descriptions for more information

> **NOT_SUPPORTED**
>
> Refer to Rates for more information about how rates are constructed.

Copyright © 2026, Oracle and/or its affiliates.

Bill Frequency - Bill Cycle vs Bill Segment Duration

An account's bill cycle defines when the system attempts to create bill segments for the account's service agreements. The word "attempt" is stressed because a service agreement may not have a bill segment on every bill created for its account.

Some examples will help make the point:

- An account may be on a monthly bill cycle (meaning we attempt to create a bill every month for the account) but contain only biannual service agreements. However, this is not sensible. Why? Because the system would attempt to create a bill every month for the account's service agreements, but only twice during the year would it succeed (because the service agreements have a biannual duration).
- An account may be on a biweekly bill cycle (meaning we attempt to create a bill every 2 weeks for the account) and contain a mixture of biweekly, monthly, and quarterly service agreements. In this scenario, every two weeks the system would create a bill that contains at least one bill segment (for the biweekly service agreement). However, 12 times a year, the bill will contain an additional bill segment for the monthly service agreement. And 4 times a year, the bill will contain one more bill segment (for a total of 3) for the quarterly service agreement.

In sum, the account's bill cycle controls when the system attempts to create a bill for the account's service agreements. Whether a service agreement contributes a bill segment to the bill is a complicated subject. The topics in this section describe how the system knows it's time to create a bill segment for a service agreement.

> **Important**
>
> An account's bill cycle should attempt to create bill segments at least as often as the shortest service agreement duration. For example, if an account has both monthly and quarterly service agreements, the account should be placed on a monthly bill cycle. Refer to How Does An Account Get Its Bill Cycle? for more information.

- Ways To Control The Start Date Of A Bill Segment
- Ways To Control The End Date Of A Bill Segment
- Preventing Short Bill Segments

Ways To Control The Start Date Of A Bill Segment

Bill segments produced for a service agreement have two time periods:

- The bill segment period. The bill segment period defines the entire period of time covered by a bill segment's charges.
- The consumption period. The consumption period defines the period of time used to calculate the number of days for daily charges.

The consumption period almost always starts one day after the bill segment period. The consumption period always ends on the bill segment's end date. For example, a bill segment period that spans 5-Jan-2002 through 6-Feb-2002 will almost always have a corresponding

Copyright © 2026, Oracle and/or its affiliates.

consumption period of 6-Jan-2002 through 6-Feb-2002. The reason that the start dates don't match is because a bill segment's start date equals the end date of the prior bill segment (i.e., the start date was already counted in the previous bill segment's consumption period and we don't want to count it twice).

The only time when the previous paragraph isn't true is the first bill segment that's produced for a new service agreement. This is because different companies count the first day of a new service agreement differently than others. Refer to Initial Consumption Period Considerations for examples and an explanation of how you can control the start date of the consumption period.

Ways To Control The End Date Of A Bill Segment

The following points describe the different methods that can be used to define the end date on a service agreement's bill segments:

- **Meter Read Date.** For service agreements with metered consumption (excluding real time meters), the end date is the date of the meter read used on the last day of the bill segment.
- **Bill Period Schedule.** Non-metered service agreements may have the end date of their bill segments defined on the user-maintained bill period calendar. This option is used when bill segments must fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter).
- **Anniversary**. In addition to the Bill Period Calendar method, non-metered service agreements may have their bill segment end date based on the first day of service. For example, if service started on the 16 th of some month, the ongoing bill segments will start on roughly the 16 th of each month. This option is frequently used to bill for garbage or cable service.
- **ASAP.** For non-meter service agreements that use neither the Bill Period Calendar nor Anniversary methods, the system assumes the end date of a bill segment is the current date. This option is frequently used for service agreements that deal with items (e.g., street lamps, phone booths).
- **Billable Charge.** For non-metered service agreements that exist to levy billable charges, the start AND end dates are defined on the billable charge information that is entered by a user or interfaced from an external system.

The topics in this section discuss each of the above methods.

- Using The Meter Read Date Method
- Using The Bill Period Schedule Method
- Using The Anniversary Method
- Using Billable Charges

Using The Meter Read Date Method

A metered service agreement's SA type does not contain a specific indication that the service agreement is used to charge for metered service. Rather, if the system finds metered service points linked to the service agreement, it knows to use a meter read date as the end date of the bill segment.

Copyright © 2026, Oracle and/or its affiliates.

Using The Bill Period Schedule Method

The bill period schedule method causes the bill segment end date to be calculated using a bill period schedule. The bill period's schedule defines when bill segments are produced and their respective end dates. This method is used when the bill segment needs to fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter).

The service agreement's SA type indicates if the service agreement uses this method, and if so, the specific bill period whose schedule defines the end dates.

> **Note**
>
> **Future and Past End Dates.** It's important to be aware that a bill period schedule can be used to generate end dates that are in the future or in the past.

> **NOT_SUPPORTED**
>
> For more information about bill periods, refer to Designing Bill Periods.

Using The Anniversary Method

The anniversary date method causes the bill segment end date to be calculated based on the first day of service. For example, if a service agreement started on the 16 th of some month, the ongoing bill segments will start on roughly the 16 th of each month. This method is frequently used to bill for garbage or cable service. When using this method, you also define the applicable billing frequency (i.e., monthly, bi-monthly, quarterly, etc.).

The following table contains the bill periods for a service agreement's starting on 23-Feb-99 that uses a monthly frequency.

Start Date	End Date	Billing Days
23-Feb-99	25-Mar-99	30
25-Mar-99	25-Apr-99	31
25-Apr-99	25-May-99	30
25-May-99	25-Jun-99	31
25-Jun-99	25-Jul-99	30
25-Jul-99	25-Aug-99	31
25-Aug-99	24-Sep-99	30
24-Sep-99	24-Oct-99	30
24-Oct-99	24-Nov-99	31
24-Nov-99	24-Dec-99	30
24-Dec-99	24-Jan-00	31
24-Jan-00	23-Feb-00	30

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Future and Past End Dates.** It's important to be aware that anniversary billing can generate end dates that are in the future or in the past. The future-option generates the next anniversary date after the current date. The past-option generates the anniversary date that immediately precedes the current date.

The service agreement's SA type indicates if the service agreement uses this method, and if so, the billing frequency.

> **NOT_SUPPORTED**
>
> For more information about the SA Type attributes that control this method, refer to Setting Up SA Types.

Using Billable Charges

Bill segments for billable charge service agreements have both their start and end dates defined on the respective billable charge.

> **NOT_SUPPORTED**
>
> For more information about billing billable charges, refer to How To Create A One-Time Invoice . For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about interfacing a billable charge from an external system, refer to Uploading Billable Charges .

The service agreement's SA type indicates if the service agreement uses this method, and if so, the billing frequency.

> **NOT_SUPPORTED**
>
> For more information about the SA Type attributes that control this method, refer to Setting Up SA Types.

Preventing Short Bill Segments

Every SA type contains an attribute defining the minimum number of days on a bill segment. Whenever the system attempts to create a bill segment other than the final bill segment, it

Copyright © 2026, Oracle and/or its affiliates.

checks if the number of days is at least as great as the minimum. If not, the bill segment will not be created as part of this bill run. Rather, the system waits until the number of days in the bill segment is at least as large as the minimum. This is true for all methods of bill duration calculation.

Prorating Charges When a Rate is Applied

It is possible for some of the prices that appear on a bill segment to change during the course of the bill period.

NOT_SUPPORTED

Refer to Effective Dates & Price Proration for more information about how charges are prorated during rate application.

Batch Billing

99.9% of all bills are created by the batch billing background process BILLING and require no human intervention before they can be finalized and sent to a customer. The other 0.1% are created by users on-line / real time. This section discusses several important concepts associated with batch billing. If your implementation requires processing of data by CIS Division the BILLING background process provides an optional parameter that allows a CIS Division to be specified as an input parameter.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

- Window Billing And The Bill Cycle Schedule
- Confirming A Batch Of Bills Before Completing Them
- Canceling A Batch Of Bills After They're Complete
- Reopening A Batch Of Bills After They're Complete
- Fixing Errors Detected In Batch Billing
- Completing Pending Bills

Window Billing And The Bill Cycle Schedule

NOT_SUPPORTED

Refer to The Cyclical Billing Process & Window Billing for more information about how an account's bill cycle dictates when a customer is billed.

Copyright © 2026, Oracle and/or its affiliates.

Confirming A Batch Of Bills Before Completing Them

If you're implementing new rates or if something unusual is being introduced that affects billing, you may want to turn off the bill cycle schedule's Freeze and Complete switch. If this switch is off, the system creates bills, it just doesn't freeze and complete them. You can then review the entire batch of bills to make sure they're clean.

- If you find the bills are wrong, correct the source of the error and rerun the bill cycle. The system will remove all incomplete bills and then reproduce them using the corrected information in the system.
- If you find the bills are correct, turn the Freeze and Complete switch on and rerun the cycle. The system will remove all incomplete bills and then reproduce them. Assuming nothing changed, the bill amounts will be the same.

NOT_SUPPORTED

Refer to Setting Up Bill Cycle Schedules for more information.

Canceling A Batch Of Bills After They're Complete

If you need to cancel an entire batch of bills because they were created using faulty data (e.g., the wrong tax rate was defined in the rate), you can. A background process (MASSCNCL) exists for this purpose.

This background process will cancel all the frozen bill segments for the latest run of a given bill cycle's schedule. Optionally, you can cancel bills for a given bill date within the bill cycle's schedule. The cancel reason used on the bill segment is the one marked as the bill cancel reason for mass cancel. Refer to Setting Up Bill (Segment) Cancellation Reasons for more information.

When the cycle is billed again, new bill segments will be created, and the original bill segments and the cancellations will automatically be hidden from the customers.

Reopening A Batch Of Bills After They're Complete

If you need to reopen an entire batch of bills because they were completed prematurely (e.g., payments did not get posted before billing ran, so recent payments were not reflected in the account balances), you can. A background process (MASSROBL) exists for this purpose.

This background process will reopen all the bills for a given bill date for the latest run of a given bill cycle's schedule.

Fixing Errors Detected In Batch Billing

If an "error" bill segment is created by the batch billing process, the system attempts to fix the offending segment each night by regenerating it using the information that exists at that time.

Copyright © 2026, Oracle and/or its affiliates.

Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And, once a bill is error-free, it will be completed and sent to the customer.

Important

Automatic regeneration only works during the account's bill window. If an "error" bill segment is not successfully regenerated on the last night of the account's bill cycle, it will remain in error until the next time the account's cycle runs (unless a user corrects it real time). At that time, the system will generate another bill error indicating the prior bill segment is in error (and then there'll be two bill segments in error).

Completing Pending Bills

If you need to complete pending bills created by the batch billing process, you can. A background process (C1-BLCMP), exists for this purpose.

This background process will complete all pending bills on a given bill date for a given bill cycle's schedule. This process does not delete and regenerate freezable bill segments linked to pending bills.

NOT_SUPPORTED

Refer to Bill Errors for more information.

Billing Financial Transaction Considerations

The topics in this section provide information about the financial impact of a bill segment.

- Billing - Current Balance versus Payoff Balance
- The Source Of GL Accounts On A Bill Segment's Financial Transaction
- Determining the Statistical Quantity

Billing - Current Balance versus Payoff Balance

A bill segment's financial transaction affects a service agreement's payoff balance and/or current balance. In this section, we describe these two balances.

Warning

If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

Copyright © 2026, Oracle and/or its affiliates.

- When Current Balance Equals Payoff Balance
- When Current Balance Differs From Payoff Balance
- Bill Segment Type Controls Which Balance(s) Are Affected

When Current Balance Equals Payoff Balance

For most service agreements, payoff balance and current balance are always the same (or in colloquial speech - the amount the customer thinks they owe equals what they really owe). Let's run through a typical example. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied (i.e., the running balance):

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125	125	125
15-Jan-99	Payment: $150	-25	-25
2-Feb-99	Bill: $175	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200	200	200
15-Mar-99	Payment: $150	50	50
2-Apr-99	Bill: $225	275	275

As you can see, payoff balance and current balance are always in sync.

When Current Balance Differs From Payoff Balance

For some service agreements, payoff balance and current balance differ (or in colloquial speech - the amount the customer thinks they owe differs from what they would owe if they wanted to pay off their account).

Let's run through an example of a customer on a budget to illustrate a classic service agreement where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0

The following table describes the various types of service agreements where these balances may differ.

Type Of Service Agreement	What Payoff Balance Holds	What Current Balance Holds

Copyright © 2026, Oracle and/or its affiliates.

Deposit	Payoff balance holds the amount of deposit you are holding on behalf of the customer's debt. This amount is showed as a credit (with a negative sign) because you truly owe the customer this money.	Current balance holds the amount the customer owes for a deposit.
Charitable contribution	N/A - charitable contribution service agreements never have a payoff balance.	Current balance holds the amount the customer owes for a charitable contribution.
Payment arrangements	Payoff balance holds the total amount of delinquent debt that remains unpaid. When a payment arrangement is started, you transfer the payoff balance from the utility service agreement to the payment arrangement service agreement. Each billing period, the customer pays down this balance.	Current balance holds the amount the customer owes in respect of the billed payment arrangement amounts.
Service agreements on a budget plan	Payoff balance holds the actual amount owed (actual bills minus payments).	Current balance holds the amount the customer owes in respect of the billed budget amount.

Bill Segment Type Controls Which Balance(s) Are Affected

Every bill segment references a bill segment type (the bill segment type comes from its service agreement's SA type). The bill segment type controls how payoff balance and current balance are affected by the bill segment amount. It also controls the algorithm used by the system to calculate the bill segment's bill lines.

NOT_SUPPORTED

Refer to Designing and Defining Bill Segment Types for more information about how bill segment type affects how a bill segment is produced and how its financial transaction is generated.

The Source Of GL Accounts On A Bill Segment's Financial Transaction

A bill segment's financial transaction also contains the double-sided accounting entry that defines how the bill segment affects the general ledger.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Source Of GL Accounts On Financial Transactions for a description of where the system extracts the distribution codes used to construct the GL accounts.

Determining the Statistical Quantity

The statistical quantity for each general entry line in the financial transaction is populated by adding together the statistical quantity for all bill calculation lines for the same distribution code. The bill calculation line indicates a statistical quantity if its corresponding calculation rule has marked the GL statistical quantity setting.

The statistical quantity for a bill calculation line is the prorated billable quantity for that line.

- If the corresponding calculation rule for the bill calculation line is not marked **measures peak**, then the statistical quantity corresponds to the billable quantity for the bill line. This is because billable quantities for SQs that do not represent a peak quantity are prorated when applying the rate.
- If the corresponding calculation rule for the bill calculation line is marked **measures peak**, then the statistical quantity may not correspond to the billable quantity for the bill calculation line. SQs that measure peak are not prorated when applying the rate. However, when determining the statistical amount to send to the general ledger, the quantity is prorated in the same manner as quantities that do not measure peak amounts.

Note

Bill Calc Line Statistical Quantity. The statistical quantity for each bill calculation line is calculated when generating the bill segment and used to populate the statistical quantity for the FT GL entry. The calculated value is not saved in the bill calc line record.

The Source Of Bill Routing Information

When a bill is completed, the system creates a bill routing for each person linked to the account who receives a copy of the bill. The bill routing record contains the information that controls how, where and to whom a bill is sent.

The following points describe how this works:

- Each person who receives a copy of an account's bill is listed on the account's person information.
- A bill routing is created for each such person when the bill is completed.
- The information that appears on the bill routing record is controlled by the **bill route type** specified on the respective account / person information.
- If the bill route type's routing method is email or fax and the system has been configured to use notification preferences for bill routing, a bill routing is created for each active contact

Copyright © 2026, Oracle and/or its affiliates.

preference whose notification type is associated with the **bill route type** specified on the respective account/person information. Otherwise, bill routings are created as follows:

  - If the bill route type indicates the person's bills are routed via fax, the "address line 1" address constituent is populated with the person's fax number. The system knows which of a person's person contacts is a fax number is by the contact routing or which of a person's phone numbers is a fax number by the phone type.. If person contacts are used, the primary fax is selected. If phone is used and the person has multiple fax numbers, one is selected at random.
  - If the bill route type indicates the person's bills are routed via email, the "address line 1" address constituent is populated with the person's Email address. The system knows which of a person's person contacts is an email or knows where to find the person email. If person contacts are used, the primary email is selected.

- If the bill route type indicates the person's bills are routed via the postal service, the address constituents are populated with the address specified on the account / person's **bill address source**. Note, if the person has a seasonal address effective on the business date, the seasonal address will be used regardless of the value of **bill address source**.

> **NOT_SUPPORTED**
>
> Refer to Printing Bills and How To Reprint A Bill (For The Original Recipients or For Someone New) for more information. Refer to Account - Person Information for information regarding how to control who receives a copy of a bill, where the bill is sent, and how the bill is formatted.

Bill Messages

The topics in this section describe how messages are linked to a bill and bill segment.

- The Source Of Bill Messages
- Substituting Field Values Into A Message

The Source Of Bill Messages

When a bill is completed, the system sweeps bill messages from the following sources onto the bill.

> **Note**
>
> **Bill or Bill Segment Messages.** Bill messages will be linked either to the bill or to one its bill segments depending on the source of the bill message code. In other words, messages associated with a service agreement (directly or indirectly) will be linked to the bill segment; messages associated with an account (directly or indirectly) will be linked to the bill.

- All permanent and temporary bill messages linked to an account are linked to the bill when it is completed. Refer to Account - Bill Messages for more information. Note well that all

Copyright © 2026, Oracle and/or its affiliates.

temporary bill messages are removed from the account when they are swept onto a bill (a temporary message is swept onto the next bill produced for the account).

- The system checks if the bill's account's customer class has bill messages. If so, it links all such messages that are effective on the bill date to the bill when it is completed. Refer to Setting Up Customer Class Bill Messages.
- All permanent and temporary bill messages linked to service agreements that contributed bill segments to the bill are linked to their respective bill segment when the bill is completed. Refer to Service Agreement - Miscellaneous for more information. Note well that all temporary bill messages are removed from the service agreement when they are swept onto a bill segment (a temporary message is swept onto the next bill segment produced for the service agreement).
- The system checks if each bill segment's rate has bill messages. If so, it links all such messages that are effective on the bill segment's start date to the bill segment when the bill is completed. Refer to Defining Calculation Rules for more information.
- The system checks if the service provider associated with each sub bill segment has bill messages. If so, it links all such messages that are effective on the bill segment's start date to the bill segment when the bill is completed. Refer to Setting Up Service Providers.

In addition, a user may manually add an ad hoc message to a bill.

And finally, you can develop your own background processes and algorithms that add bill messages to accounts, service agreements, bills and/or bill segments. Refer to Substituting Field Values Into A Bill Message for examples.

Substituting Field Values Into A Message

Many bill messages contain static text (i.e., the message is the same on every bill). However, the system supports messages whose contents are dynamic. For example, consider the bill message Thank you for your charitable contributions in 2000 for $123.12, please keep this bill for tax purposes. This message contains two substitution values (the year and the amount) as is therefore considered dynamic.

Dynamic messages can be implemented as follows:

- Create a bill message code whose Message on Bill contains substitution values. For example, the bill message code to produce the message illustrated above would contain the following Message on Bill - Thank you for your charitable contributions in %1 for %2, please keep this bill for tax purposes.
- Use either of the following methods to link the message code and its substitution parameters to the bill:
  - Create a background process or algorithm to insert a permanent or temporary bill message on the appropriate accounts. Then, when an account's bill is next completed, the system will sweep the message (and its substitution parameters onto the bill). In addition to inserting the appropriate bill message code, your background process / algorithm must also insert the appropriate substitution values. The name of the table in which account messages are inserted is **CI_ACCT_MSG**. The name of the table in which a message's substitution values are inserted is **CI_ACCT_MSG_PRM** (you will insert one row for each substitution field).

    Please see the background process (REACH) for an example of a background process that performs this type of function. Please see the algorithm identified by DEP PIF MSG for an example of an algorithm that performs this type of function.

Copyright © 2026, Oracle and/or its affiliates.

– Rather than adding the dynamic message to account (and then letting the bill completion logic transfer the account message to the bill), you could construct a bill completion algorithm that adds the message code and substitution values on the bill (during bill completion). The name of the table in which bill messages are inserted is **CI_BILL_MSGS**. The name of the table in which a message's substitution values are substituted is **CI_BILL_MSG_PRM** (you will insert one row for each substitution field).

A Bill May Affect More Than Just Customer Balances

The topics in this section provide information about obscure things that may happen when a bill is created, frozen, or canceled.

- Auto Cancel / Rebill If A Service Agreement Is Stopped Retroactively
- Cancel Historical, Estimated Bill Segments When a Real Read Is Found
- Deposits May Be Refunded
- FT Freeze Repercussions

Auto Cancel / Rebill If A Service Agreement Is Stopped Retroactively

If the system detects a bill segment with a bill date after the stop date of the service agreement (because the service agreement was retroactively stopped), the system automatically cancels the bill segment and creates a new one with an end date equal to the service agreement's stop date.

Cancel Historical, Estimated Bill Segments When a Real Read Is Found

A special plug-in spot exists on bill segment type that can be used to automatically cancel historical, estimated bill segments when a “real” read is used on the current bill. This plug-in spot is invoked only when an excessive read (the prior read is more than the current read) situation is detected.

If you wish that the spot be invoked all the time and not only when an excessive read situation is encountered, you must setup the ***Always Call Auto Cancel*** option type of the ***Financial Transaction Options*** Feature Configuration. You will normally do this if you want the type of readings to be the basis of the auto cancellation. For example, you want to cancel estimated bills and rebill distributing usage more accurately across a tiered rate.

These algorithm would typically be specified on bill segment types that calculate bills using meter reads.

> **Note**
>
> This plug-in spot is typically used on bill segment types that calculate bills using meter reads. It is typically not used when usage is requested from Oracle Utilities Meter Data Management.
>
> This algorithm is called by the system when creating bills in batch. It does not get called when a user creates a bill online.

Copyright © 2026, Oracle and/or its affiliates.

Deposits May Be Refunded

NOT_SUPPORTED

Refer to Refunding Deposits for how deposits may be refunded when a deposit service agreement is "final billed".

FT Freeze Repercussions

NOT_SUPPORTED

Refer to Obscure Things That Can Happen for more information about things that can happen when an FT is frozen (and FT's get frozen during billing).

Budget Billing

The topics in this section provide background information about a variety of budget billing issues.

NOT_SUPPORTED

We strongly recommend familiarizing yourself with the topics described in Designing and Defining Budget Plans to fully appreciate how budgets fit into the system's financial architecture.

- Account versus Service Agreement
- How To Set Up A Customer To Participate In A Budget Plan
- The Financial Impact Of Budget Plans
- Budget Billing Background Processes
- Canceling A Budget

Account versus Service Agreement

Customers who pay on a budget plan tend to believe that there is one budget amount applied to their account's bill each month. This is not true. Rather, each service agreement has an

Copyright © 2026, Oracle and/or its affiliates.

individual budget amount (stored in the recurring charge array). The sum of an account's service agreements' budget amount is the account's budget amount.

Why does each service agreement have a unique budget amount? Because all financial accounting in the system is performed at the service agreement level, not for an entire account.

While a unique budget amount is defined for each service agreement, a user can use a simple page to view and change the budget amount for all eligible service agreements linked to an account.

It's important to be aware that some service agreements linked to an account may not be eligible for budget plans. For example, assume you have a customer with a deposit service agreement and one or more utility service agreements. This customer will be billed for their deposit regardless of the utility service agreements' budget amounts. You define whether a service agreement may participate in a budget plan on the service agreement's SA type.

How To Set Up A Customer To Participate In A Budget Plan

If a customer wants to participate in a budget billing plan, simply transfer to Account - Budget and generate a new budget for the customer.

The Financial Impact Of Budget Plans

The only difference between a customer who participates in budget billing and one who doesn't is that budget billing customer have bill segments where payoff amount differs from current amount. Why? Because the payoff amount is the actual amount of the bill. The current amount is the amount the customer is expected to pay (i.e., their budget amount).

Let's run through an example of a customer on a budget to illustrate a service agreement where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0

NOT_SUPPORTED

For more information about current and payoff amounts, refer to Current Balance versus Payoff Balance.

Copyright © 2026, Oracle and/or its affiliates.

Budget Billing Background Processes

The topics in this section describe the background processes that automate management of customer who participate in budget billing plans.

> **Note**
>
> **Budget Processing.** The budget background processes only calculate budget amounts for pending start, active and pending stop SAs whose start and end date (if any) includes the budget date. Additionally, the budget date is used to select the rate schedule that is in effect for an SA if estimating consumption is necessary.

- Budget True Up
- Budget Monitor

Budget True Up

The Budget True Up background process (BUDTRUP) examines all accounts who participate in budget billing plans. If enough time has passed since the customer's budget was last trued up, the system will calculate the account's recommended budget amount. If the recommended amount differs from their existing budget amount by more than a given tolerance percentage, the customer's budget amount (actually, the recurring charge amount on the account's service agreements) will be changed.

> **NOT_SUPPORTED**
>
> The amount of time between true ups and the true up method used to true up is defined on the account's budget plan. The tolerance percentages are defined as parameters of the true-up algorithm that is plugged in on the account's budget plan. Refer to Designing and Defining Budget Plans for all the details.

Budget Monitor

The Budget Monitor background process (BUDMON) examines all accounts that participate in budget billing plans. If the account's existing budget amount is out-of-sync with predefined tolerance percentages as defined by the algorithm plugged into the monitor budget plug-in spot, the system will perform the action indicated by the monitor algorithm. The base product supports highlighting the account on the Budget Review page or updating the account’s budget.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

The tolerance percentages are defined as parameters of the budget monitor algorithm that is plugged in on the account's budget plan. Refer to Designing and Defining Budget Plans for all the details.

Canceling A Budget

The following points describe when a budget is cancelled:

- A user can cancel a budget at any time by using the Cancel Budget action on the Account - Budget page.
- The system may cancel a budget during a collection process (if the collection process contains a Cancel Budget collection event). Refer to Types Of Collection Events for more information.
- When a service agreement is stopped, the system automatically cancels the budget.

When a customer's budget is cancelled, the following events transpire:

- Any recurring charge entry whose effective date is after the cancel date is deleted. This is possible if you are entering a back-dated stop date for the service agreement. For example, perhaps you run the Budget True Up process, which adds a new budget amount to the service agreement (effective today). Then imagine that the customer calls you and informs you that they moved out last week. If you stop the customer's service as of last week, the recurring charge effective today is deleted when the stop is finalized.
- A new recurring charge amount of zero is created for each of the customer's service agreement.
- Each service agreement's current balance is synchronized with its payoff balance (using the sync adjustment code defined on the service agreement's SA type).

Net Energy Metering

The topics in this section provide background information on net energy metering.

- What Is Net Energy Metering?
- Net Energy Metering Service Agreements
- Net Energy Metering True Up
- Setting Up the System for Net Energy Metering
- Net Energy Metering Billing Examples

What Is Net Energy Metering?

Customers in a distributed generation energy market may have renewable energy devices such as solar panels or wind turbines connected to their meters. The energy generated at this service point, as well as any energy used, could possibly be measured by a single meter, with the energy generated being netted against the energy used. This is called net energy metering (NEM). The meter records a positive number when customers use more energy than they generate, and a negative number when they generate more than they consume. These customers may owe a minimum charge each month. Their actual energy charges and

Copyright © 2026, Oracle and/or its affiliates.

generation credits are accrued until the account is adjusted at the end of a specified period, called the true up period. When the account is trued up, the consumption charges and generation credits for the period are combined to determine if the customer owes money to the utility.

Net Energy Metering Service Agreements

As mentioned in the previous section, net energy metering (NEM) customers typically owe a minimum charge each month while their actual energy charges and generation credits are accrued until true up time. This is accomplished by setting up a master service agreement (SA) to hold the monthly minimum charge and a sub service agreement to hold the customer's actual energy charges and generation credits.

For more information about service agreement relationships, refer to Defining SA Relationship Options.

Net Energy Metering True Up

To facilitate the true up process, each true up period will be represented by a sub service agreement. Each sub service agreement will be monitored by a true up monitor (TUM), a service task which is responsible for: performing the true up process at the end of the true up period; reversing true up transactions when cancel/rebills affect a period that has been trued up; synchronizing itself with any changes on its master service agreement or service agreement relationship (e.g., if the service agreement relationship is deleted, the true up monitor will cancel itself); and creating the service agreement relationship and sub service agreement for the next true up period.

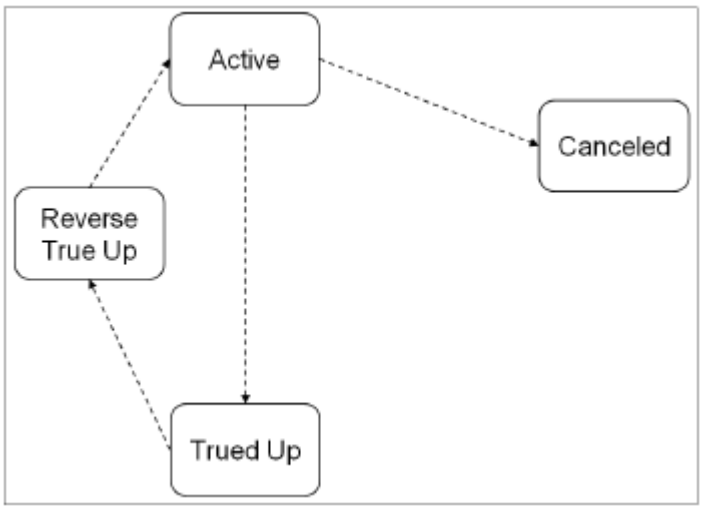
A service agreement creation algorithm on the master service agreement type is responsible for creating the initial sub service agreement and service agreement relationship. Subsequent sub service agreements are created in the true up monitor lifecycle. A service agreement creation algorithm on the sub service agreement type is responsible for creating the true up monitor. A link to the true up monitor service task is available from the characteristics page of the sub service agreement. It is important to note that the true up monitor is not a user-maintained object. End-users will not normally need to access the true up monitor. All that is visible on the true up monitor is its type, status, related sub service agreement and account. As always, relevant financial history will be accessible from the sub service agreement.

- The Lifecycle of a True Up Monitor

The Lifecycle of a True Up Monitor

The following diagram shows the lifecycle of the True Up Monitor business object:

Copyright © 2026, Oracle and/or its affiliates.



Active State: The true up monitor starts out in the Active state. Here the true up monitor is evaluated for true up processing or possible cancellation. The system will perform a true up if either the true up period has come to an end, or the service agreement is final billed.

The amount of time between true ups is defined on the true up task type. See Setting Up the System for Net Energy Metering for a description of how true up task types are configured.

Canceled State: The true up monitor is transitioned to the Canceled state if its service agreement is canceled or deleted. Because the service agreement associated with a true up monitor is a sub service agreement, it is possible that the stop date on the sub service agreement is backdated to a point prior to the start of its associated service agreement relationship. In this case, the true up monitor is also canceled.

Trued Up State: If it is time for true up, the true up monitor is transitioned to the Trued Up state. True up processing involves summing the sub service agreement's bill segments to see if the customer owes any debt (i.e., the energy they generated was not enough to cover the energy they consumed). If the sum is positive, the amount is transferred to the master service agreement for the customer to pay off. If the sum is negative, the sub service agreement balance is brought to zero (the utility keeps the credit). If your implementation wants to issue refunds to customers who generate more energy than they consume, you will need to design your own algorithm for the true up process.

The adjustment types used to transfer or zero out the sub service agreement balance are defined on the true up monitor type. See Setting Up the System for Net Energy Metering for a description of how true up monitor types are configured.

Reverse True Up State: Once the financial true up is done, a new service agreement relationship and sub service agreement are created for the subsequent true up period. If cancel/rebillls occur after the true up process has been performed, the true up monitor is transitioned to the Reverse True Up state. Any adjustments that were previously created are canceled as part of this processing. Thereafter the true up monitor is transitioned back to the Active state, where it waits until bill completion attempts another true up.

Setting Up the System for Net Energy Metering

This section summarizes the various tasks required for setting up your system for net energy metering.

Adjustment Types

The following adjustment types are used at true up time:

Copyright © 2026, Oracle and/or its affiliates.

- **Current Balance True Up.** This is the adjustment type used in syncing the payoff and current amounts on the sub service agreement associated with the true up period. This adjustment type should reference an FT algorithm of **Payoff Amount = 0 / Current Amount = Adj Amount (no GL)**.
- **Transfer To Master.** This is the adjustment type used to transfer the customer's balance from sub service agreement to master service agreement if a positive payoff balance exists on the sub service agreement after true up.
- **Write Down.** This is the adjustment type used to bring the payoff balance of the sub service agreement to zero if a negative payoff balance exists after true up.

Adjustment Type Profile

Create an adjustment type profile that references the adjustment types used in net energy metering billing.

True Up Task Type

To enable billing for net energy metering, your implementation must configure a **True Up Task Type.** This service task type captures the various configuration options that are used by the true up monitor's algorithms:

- **Minimum Days In True Up Period** defines the minimum number of days covered by the bill segments within a true up period before true up processing can occur
- The task type captures the adjustment types used at true up time
- **Reversal Adjustment Cancel Reason** is used on true up adjustments that are canceled as a result of cancel/rebill activity occurring after a true up

Algorithms

Add the following algorithms:

- The sub service agreement and service agreement relationship are initially created by an **SA Creation** algorithm on the master service agreement type. The base package provides algorithm type C1-SAT-SARL for this purpose. An algorithm should be configured with the service agreement relationship type to use.
- Creation of the true up monitor is handled by an **SA Creation** algorithm on the sub service agreement type. The base provides algorithm type C1-SAT-TUM for this purpose. The true up monitor service task type is defined as an algorithm parameter.
- Whenever a net energy metering sub service agreement is canceled, its associated true up monitor should also be canceled. The base provides algorithm C1-CNC-TUM to be plugged in as an **SA Cancel** algorithm on the sub service agreement type to accomplish this.
- The true up monitor is transitioned out of the Active or Reverse True Up state by the billing process. To that end, a **Bill Completion** algorithm needs to be configured on the master service agreement type. This algorithm is responsible for transitioning any true up monitor that is in an Active or Reverse True Up state that is related to the master service agreement. The base provides the algorithm C1-TRN-TUM for this purpose.
- When cancel/rebills occur, financial transactions related to the true up need to be reversed. To accomplish this, an **FT Freeze** algorithm must be plugged in on the sub service agreement type. The base provides algorithm C1-RV-TRUP for this purpose.

For more information on setting up service agreement Relationships, refer to Defining SA Relationship Options. For more information on service agreement type algorithms, refer to SA Type - Algorithms.

Copyright © 2026, Oracle and/or its affiliates.

Your implementation must also decide if the sub service agreement's bill segment financial transactions should impact the general ledger or not. This is controlled by the **Financial Algorithm** on the bill segment type defined for the sub service agreement type. The base provides algorithms C1-C0-GL and C1-C0-NOGL for this purpose. For more information on configuring Bill Segment Types, refer to Designing and Defining Bill Segment Types.

Net Energy Metering Billing Examples

This section contains billing examples for net energy metering. The examples assume the following:

- The customer is billed quarterly using the following tariff:
  - Customer charge of $3/day (billed on master service agreement)
  - $0.10 per kWh (billed on sub service agreement and trued up after 12 month period)
- The GL Accounting distribution codes used in the following tables, such as A/R-NEM and NEM-Liability, are for example only. Your implementation will specify its own distribution codes.
- A True Up Task Type that references the base product NEM business objects with the following configuration:
  - **Minimum Days In True Up Period:** 365
  - **Sync Current Adjustment Type:** NEM SYNC
    This is the adjustment type used to synchronize (make equal) the sub service agreement's current amount with the payoff amount at true up time.
  - **Transfer Adjustment Type:** NEM XFER
    This is the adjustment type used to transfer debt from the sub service agreement to the master service agreement at true up time.
  - **Write Down Adjustment Type:** NEM WDWN
    This is the adjustment type used to write down any remaining amount on the sub service agreement after we attempt to transfer debt to the master service agreement at true up time.

Example 1: Customer Consumes More Than They Generate

The following table outlines the customer's master service agreement financial transactions in 2010. Adjustments created in the true up period are in bold. In this example the customer's net energy results in an outstanding balance of $330 which is transferred from the sub service agreement to the master service agreement, leaving the sub service agreement for that true up period with a zero balance.

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (1/1–3/31) (90 days * $3/day = $270)	**A/R-NEM 270** **NEM-Liability <270>**	**270**	**270**	**270**	**270**
Payment	**Cash 270** **NEM-A/R <270>**	**-270**	**-270**	**0**	**0**

Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (4/1–6/30) (91 days * $3/day = $273)	**A/R-NEM 273** **NEM-Liability <273>**	**273**	**273**	**273**	**273**
Bill Segment (7/1–9/30) (92 days * $3/day = $276)	**A/R-NEM 276** **NEM-Liability <276>**	**276**	**276**	**549**	**549**
Payment	**Cash 549** **A/R-NEM <549>**	**-549**	**-549**	**0**	**0**
Bill Segment (10/1–12/31) (92 days * $3/day = $276)	**A/R-NEM 276** **NEM-Liability <276>**	**276**	**276**	**276**	**276**
NEM XFER true up adjustment (transfer from sub service agreement)	**A/R-NEM 330** **XFER <330>**	**330**	**330**	**606**	**606**

The following table outlines the customer's sub service agreement financial transactions (FTs). Note that bill segment financial transactions affect the customer's pay off balance only. Adjustments created in the true up period are in bold below and reflect the following:

- First, the current balance to the payoff balance is set using the sync adjustment type (NEM SYNC) on the service task type.
- Second, any debt from the sub is transferred to the master service agreement using sync adjustment type on the service task type (NEM XFER).

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (1/1–3/31) (1200kWh * $0.10 = $120)	**A/R-NEM 120** **NEM-Liability <120>**	**120**	**0**	**120**	**0**
Bill Segment (4/1–6/30) (500kWh * $0.10 = $50)	**A/R-NEM 50** **NEM-Liability <50>**	**50**	**0**	**170**	**0**

Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (7/1–9/30) (600kWh * $0.10 = $60)	**A/R-NEM 60** **NEM-Liability <60>**	**60**	**0**	**230**	**0**
Bill Segment (10/1–12/31) (1000kWh * $0.10 = $100)	**A/R-NEM 100** **NEM-Liability <100>**	**100**	**0**	**330**	**0**
NEM SYNC adjustment (sync current to payoff)	**n/a**	**0**	**330**	**330**	**330**
NEM XFER adjustment (transfer to master service agreement)	**XFER 330** **A/R-NEM <330>**	**-330**	**-330**	**0**	**0**

Example 2: Customer Generates More Than They Consume

The following table outlines the customer's master service agreement financial transactions (bill and payment segments are exactly the same as in example 1 above). In this example the customer's net energy results in a credit balance on the NEM sub service agreement which is written off, leaving the sub service agreement for that true up period with a zero balance (credits are not transferred to the master service agreement).

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (1/1–3/31) (90 days * $3/day = $270)	**A/R-NEM 270** **NEM-Liability <270>**	**270**	**270**	**270**	**270**
Payment	**Cash 270** **A/R-NEM <270>**	**-270**	**-270**	**0**	**0**
Bill Segment (4/1–6/30) (91 days * $3/day = $273)	**A/R-NEM 273** **NEM-Liability <273>**	**273**	**273**	**273**	**273**
Bill Segment (7/1–9/30) (92 days * $3/day = $276)	**A/R-NEM 276** **NEM-Liability <276>**	**276**	**276**	**549**	**549**

Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Payment	**Cash 549** **A/R-NEM <549>**	**-549**	**-549**	**0**	**0**
Bill Segment (10/1–12/31) **(92 days * $3/day = $276)**	**A/R-NEM 276** **NEM-Liability <276>**	**276**	**276**	**276**	**276**

The following table outlines the customer's sub service agreement financial transactions (FTs). Note that bill segment financial transactions affect the customer's pay off balance only. Adjustments created in the true up period are in bold below and reflect the following:

- First, the current balance is set to the payoff balance using the sync adjustment type (NEM SYNC) on the service task type.
- Because a credit remains on the sub service agreement, this amount will be written off using the write down adjustment type on the service task type (NEM WDWN).

Event	GL Accounting	Effect On Payoff Amount	Effect On Current Amount	Payoff Balance	Current Balance
Bill Segment (1/1–3/31) **(80kWh * $0.10 = $8)**	**A/R-NEM 8** **NEM-Liability <8>**	**8**	**0**	**8**	**0**
Bill Segment (4/1–6/30) **(-120kWh * $0.10 = $-12)**	**A/R-NEM <12>** **NEM-Liability 12**	**-12**	**0**	**-4**	**0**
Bill Segment (7/1–9/30) **(-80kWh * $0.10 = $-8)**	**A/R-NEM <8>** **NEM-Liability 8**	**-8**	**0**	**-12**	**0**
Bill Segment (10/1–12/31) **(100kWh * $0.10 = $10)**	**A/R-NEM 10** **NEM-Liability <10>**	**10**	**0**	**-2**	**0**
NEM SYNC true up adjustment (sync current to payoff)	**n/a**	**0**	**-2**	**-2**	**-2**
NEM WDWN true up adjustment (write down credit)	**NEM A/R 2** **E - W/O <2>**	**2**	**2**	**0**	**0**

For more information about current and payoff amounts, refer to Billing - Current Balance versus Payoff Balance.

Copyright © 2026, Oracle and/or its affiliates.

Using Billable Charges For Pass Through / Convergent Billing

The term "pass through" billing is used to describe the practice of receiving charges calculated by third parties and presenting them on the customer's bill along with your own charges. "Pass through" billing is implemented in the system using Billable Charges.

The following points provide information to help you decide the most appropriate way to implement "pass through" billing given your specific requirements:

- Customers with pass through charges will need a separate service agreement to hold the pass through charges. We refer to this type of service agreement as a "billable charge" service agreement.

> **Note**
>
> A service agreement's SA type controls whether a service agreement can have billable charges linked to it. Specifically, the SA type must have a "special role" of Billable Charge.

- A billable charge service agreement holds the billable charges until the customer is next billed. At that time, a separate bill segment will be created for each unbilled billable charge linked to the billable charge service agreement.
- While it is not required, we recommend creating a separate billable charge service agreement for each type of pass through charge. For example, if you can receive "pass through" charges from the local water company and the local cable company, we'd recommend having two billable charge service agreements for each customer.
- As described under We Bill For Them, you can setup the system to automatically create billable charge service agreements when you start regular service for your customers. If you take this approach, you can also have the system create the payable transaction to pay the 3rd party. Refer to Pay At Bill Time vs. Pay At Pay Time for more information.
- You can interface billable charges using the Billable Charge Interface. The interfaced charges can consist of any of the following:
- Pre-calculated bill lines that will be presented "as is" to the customer.
- Service quantities that are used by the system to calculate the charges on behalf of the third party. For example, the 3rd party may pass through the number of kilowatt-hours and you calculate the bill lines for them. Refer to Uploading Consumption Rather Than Bill Lines for more information.
- Meter read details that you want printed on the bill. Note, the base package bill segment creation algorithm simply sweeps billable charge read details onto the related bill segment's read detail. This algorithm does NOT use these reads to derive service quantities. This is intentional as most implementations that import reads also import service quantities and therefore deriving service quantities is unnecessary (and also error prone). If your implementation needs to derive service quantities from these reads, you can develop your own bill segment creation algorithm to do just this.
- All of the above. You might interface both bill lines and service quantities if the third party calculated the charges but did not calculate the taxes. The dollar amount for which taxes will be calculated would be defined using a service quantity. Refer to Calculating Taxes On Uploaded Charges for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You cannot interface interval data via the billable charge interface. Rather, you would need to interface the interval data using the standard interval data set interface and then let the system calculate the charges for you.

- The bill lines on a billable charge can fit into any of the following categories:
- Memo-only (i.e., don't affect the general ledger). A bill line that is memo-only contain information that is purely informational. For example, the 3rd party might pass the start and end meter reads that they used to calculate the charges. You can use memo-only bill lines to show this information on the customer's bill. If a bill line is not memo-only (i.e., it affect the general ledger), a distribution code must also be specified to define how the lines affects the general ledger.
- Show on the customer bill. It is possible to create bill lines that affect the general ledger, but are not shown on the customer's bill. This is an unusual practice, but it happens.
- Summary / detail. Each bill line has an indication of whether it is a summary or detail line. This only impacts bill presentation.

> **Note**
>
> The above indicators can be defaulted onto a billable charge line by using Billable Charge Line Types when you interface the lines into the system.

- Characteristics (i.e., user-defined fields) can be associated with billable charge lines. You might populate this information if you are interfaced information that may be useful during bill presentation or for reporting purposes.
- If service quantities are specified on a billable charge, they are saved on the bill segment created when the billable charge is billed. This means that they system can support creation of a graph of historical usage on the bill, as well as other queries on customer usage. In addition, it might be possible to event estimate consumption based on these service quantities (but such a plug-in is not supplied with the base package).

> **Note**
>
> The system does not support passing in items (e.g., street lamps). This could be a problem if you need to perform ratable calculations for the 3rd party based on, say, the number of street lights. However, if you require ratable, street light pass through billing, you can calculate a service quantity for each type of street light and save this information on the billable charge. The associated rate would then have a price per service quantity rather than a price per item type.
>
> The system does not support sending in raw meter reads with billable charges. This consumption needs to be passed in as a service quantity. However, you can pass in meter read details to be printed on the bill segment.

- Cancel / rebill for billable charges is quite different than for normal bill segments. A billable charge bill segment can be cancelled, and this will reverse the financial effects of the billable charge. But without new information from the source, there is no way to rebill the

Copyright © 2026, Oracle and/or its affiliates.

customer. Therefore, if the original charges were incorrect, the source system would send both a reversal of the charges and a newly revised set of information. These could be passed as two separate billable charges or they could combined on a single billable charge.

- In terms of the customer information setup, while a service agreement that holds billable charges does not need a service point, there are advantages to having this information in the system. For one thing, if jurisdictional taxes are being calculated by the system, the definition of the jurisdiction(s) is held on the service point or premise. Also, the link to the service point defines the service address, facilitates customer lookup, causes information to appear in Control Central's trees, etc.
- For most other functionality, the billable charge service agreement supports the same functionality as normal service agreements. This includes budget billing, payment distribution, credit & collection tracking and events, current & payoff balances, movement of old debt onto payment arrangements, etc.

> **NOT_SUPPORTED**
>
> For more information about billing billable charges, refer to How To Create A One-Time Invoice. For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about creating sub service agreements to hold billable charges, refer to We Bill For Them. For more information about interfacing a billable charge from an external system, refer to Uploading Billable Charges.

Printing Bills

The contents of this section describe the technical implementation of both an online and batch bill production.

- Bill Routings Are Created For Each Recipient Of A Bill
- Bill Route Types Control The Information Merged Onto Bills
- Technical Implementation Of Online Bill Display
- Technical Implementation Of Printing Bills In Batch
- Reproducing The Bill Print Flat File
- How To Reprint A Specific Bill
- Who Gets A Copy Of A Bill?
- Final Bills and Bill Print
- The Bill Extract Records
- XML Bill Extracts

Bill Routings Are Created For Each Recipient Of A Bill

Refer to The Source Of Bill Routing Information for a description of how the system constructs the information used to route a bill to one or more recipients.

Copyright © 2026, Oracle and/or its affiliates.

Bill Route Types Control The Information Merged Onto Bills

Every bill routing record references a Bill Route Type.

The bill route type controls the following for interacting with a document composition application:

> **Note**
>
> CIS Division can override the extract algorithm and/or the background process found on the bill route type. The system processes first check the bill's account's CIS Division for an override specific for the bill route type.

- It contains an algorithm that is responsible for extracting the information merged onto your bills. Algorithms of this type are called under the following scenarios:
  - The background process that interacts with your bill print software calls these algorithms.
  - If your bill print software has the ability to construct a real-time image of a bill, you can plug-in an Online Bill Display algorithm on the Installation Record. This type of algorithm will call the bill extract algorithm for an appropriate bill route type to extract the information that is merged onto the bill. Refer to Technical Implementation Of Online Bill Display for the details.
- It contains the ID of the background process responsible for interacting with your bill print software in batch.

The bill route type controls the following for interacting with a reporting engine, such as BI Publisher:

- It contains an algorithm that is responsible for returning the appropriate report code and other information in order to extract the information to be merged onto your bills. Typically, the reporting engine itself is responsible for extracting and formatting the data. Algorithms of this type are called under the following scenarios:
  - The background process that interacts with your reporting engine calls these algorithms.
  - If your reporting engine has the ability to construct a real-time image of a bill, you can plug-in an Online Bill Display algorithm on the Installation Record. This type of algorithm will call the bill extract algorithm for an appropriate bill route type to return the appropriate report code. Refer to Technical Implementation Of Online Bill Display for the details.
- It contains the ID of the background process responsible for interacting with your reporting engine in batch.

Technical Implementation Of Online Bill Display

Users can view an image of any bill that is sent to a customer on Bill - Main if you plug-in an Online Bill Display construction algorithm on the Installation Record

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Some display algorithms rely on the bill extract algorithm on each Bill Route Type

The system can make a call to an external system to retrieve a bill image. Many bill print vendors support this functionality.

The system provides an Installation – Online Bill Display algorithm type (C1GENBILLURL) that can generate a dynamic URL.

- The algorithm of this algorithm type supports dynamic variables such as the bill ID and static variable such a vendor or implementation supplied security key.
- The algorithm constructs a URL including generating HASH or security keys.
- The system displays the retrieved image in a separate browser session
- Refer to the algorithm type description for information about this algorithm type.

The system can interact with either document composition applications, or a reporting engine (such as BI Publisher).

The system provides an Installation – Online Bill Display algorithm type (C1-BL-DISP) that interacts with Documaker to render an online image of your bill. The following points describe what takes place when clicking **Display Bill**.

- The algorithm based on this algorithm type calls the bill extract algorithm for an appropriate bill route type (as determined by the algorithm).
- The bill extract algorithm will construct the information that appears on the bill (refer to algorithm type C1-BLEX-XML for further details).
- The constructed information is passed to document composition software that will reconstruct the bill's image in a PDF format.
- The system displays the PDF in a separate browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

> **Note**
>
> Refer to The Bill Extract Records for examples of the record layouts passed to the document composition software.

The system also provides an Installation – Online Bill Display algorithm type (C1-ONLN-CR) that interacts with BI Publisher to render an online image of your bill. The following points describe what takes place when clicking **Display Bill**:

- The algorithm based on this algorithm type calls the bill extract algorithm for an appropriate bill route type (as determined by the algorithm).
- The bill extract algorithm returns the appropriate report code to the Online Bill Display algorithm, which in turn returns the report code to the bill page (refer to algorithm type C1-BLEX-BIP for further details). This algorithm is not actually extracting data because when you use BI Publisher reports, the report itself is responsible for extracting and formatting the data.
- A user exit on the bill page invokes BI Publisher and displays the bill in a separate browser session.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Bill Print in BI Publisher for examples of the data extracted for the report.

Technical Implementation Of Printing Bills In Batch

The batch process that extracts bill related information reads all bill routing records in a given run number that are marked with its batch control ID

The base package example of this process (POSTROUT) simply calls the bill extract algorithm on the routing record's route type. The algorithm formats the information to be passed to the bill print software or reporting engine. Refer to Bill Route Types Control The Information On Bills for more information.

> **Note**
>
> **If your software doesn't support online bill display.** The algorithm that formats bill extract records that's plugged in on bill route type serves two purposes: 1) it interacts with the online bill display algorithm plugged into the installation record to support online images of a bill, and 2) it interacts with the background process to support printing your bills in batch. If your bill print software does not support the rendering of bill images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that works with your bill print software to print bills in batch (and then specify this batch process on your bill route type).

Reproducing The Bill Print Flat File

You can reproduce the flat file at any time. Simply request the download process and specify the run number associated with the historic run.

How To Reprint A Specific Bill

Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for instructions describing how to reprint a specific bill.

Who Gets A Copy Of A Bill?

A copy of a bill is sent to every individual / business specified in the bill's routing details.

There are two ways in which a bill routing detail can be created:

- When a bill is completed, the system creates a routing detail for every person linked to the account that wants to receive a copy of the bill (as specified on Account - Person Information). Refer to The Source Of Bill Routing Information for more information.

Copyright © 2026, Oracle and/or its affiliates.

- After a bill is completed, you may insert a bill routing record. Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for more information.

Final Bills and Bill Print

If a bill is produced for an account where all of the accounts SAs are stopped, closed or cancelled; the bill is flagged as being the final bill on the flat file produced by the postal routing process (refer to Bill Record - 0100 for the details of this switch).

If a bill is considered to be a final bill and the amount owing for the entire account is less than the final bill threshold amount on the installation record, the bill will be skipped (i.e., it won't appear on the flat file produced by the postal routing process). Note, this logic also suppresses bills from being produced when a payment is received AFTER the final bill and the account's balance falls beneath the installation record's final bill threshold amount. Refer to Installation Options - Billing for more information.

The Bill Extract Records

Numerous different types of records are interfaced to the document composition application. However, they all share a common structure for bills:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the document composition application bill template.
- The next 244 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed bill appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted bills are printed.
- The next 3 bytes are a filler that is available for future use.
- The remaining bytes contain the **Bill Information** that appears on the printed bill. The type of information differs for each type of record.

The topics in this section describe each component.

- Printer Keys (Record Types)
- Sort Key
- Mailing Key
- Bill Information

Printer Keys (Record Types)

The following table defines each printer key (i.e., record type) on the bill print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
0100	Bill record	Required. 1 per bill.
0150	Account Balance Record	Required. 1 per Special Role.
0300	Current charge record	1 per current charge.

Copyright © 2026, Oracle and/or its affiliates.

0600	Payment record	1 per payment.
0900	Premise record	1 per premise.
1100	Service agreement record	1 per SA linked to the premise.
1300	Bill segment record	1 per bill segment linked to the SA.
1500	Bill calculation header	1 per bill calculation header linked to the bill segment.
1700	Bill calc line	1 per bill calculation line linked to the bill calculation header
1900	End of record type 1500	1 for each bill calculation header
2100	Read detail	1 for each meter read detail linked to the bill segment
2400	Service quantity detail	1 for each service quantity detail linked to the bill segment
2700	Item detail	1 for each item detail linked to the bill segment
2840	Bill segment messages	1 for each message detail linked to the bill segment
2900	End of record type 1300	1 for each bill segment
3100	Consumption history (for graph)	1 for each historical consumption period that appears on the graph
3400	Adjustment detail	1 for each adjustment
3420	Adjustment Calc Line	1 or more per adjustment
3600	End of record type 1100	1 for each service agreement
3700	End of record type 0900	1 for each premise
3900	Bill message detail	1 for each message
9999	End of record type 0100	1 for each bill

Sort Key

The following table defines the structure of the sort key that appears on each bill print record. Please note that different components are filled in for each printer key.

Field Name	Format	Source/Value/Description
Bill Print Intercept	A8	This is the user ID, if any, who reviews the bill (as specified on the account record)
Bill ID	A12	
Sequence Number	N2	The sequence number of the account / person information associated with this bill routing information.

Copyright © 2026, Oracle and/or its affiliates.

Copy Number	N2	The number of copies desired by the account / person associated with this bill routing information.
Bill record group	A2	10 Bill 20 Payment 30 Current charge 40 Message 50 Non-Premise 55 Non-Premise end 60 Premise 70 Bill end
Filler	A2	
Premise State	A6	Only applicable for Premise based records.
Premise City	A90	Only applicable for Premise based records.
Premise Address 1	A64	Only applicable for Premise based records.
Premise Id	A10	Only applicable for Premise based records.
Premise record group	A2	10 Premise record 20 SA level record 30 Premise end record
SA Print Priority	A2	
SA Id	A10	
SA Group	A2	For Premise based records: 10 SA 20 Histogram 30 Current charge 40 Adjustment 50 SA end record For Non-Premise based records: 10 SA 20 Current Charge 30 Adjustment 40 SA end record
Bill Segment End Date	A10	If the record is related to an adjustment, this is the adjustment creation date. If the record is for histogram detail, this contains the histogram detail number.
Bill Segment Id	A12	If the record is related to an adjustment, this is the adjustment ID.

Copyright © 2026, Oracle and/or its affiliates.

Bill Segment Group	A2	10 Bill segment 20 Bill segment calculation group 30 Bill segment read detail 40 Bill segment SQ detail 50 Bill segment item detail 55 Bill segment message 60 Bill segment end
Bill Calc Header Sequence	N2	For records related to read detail, SQ detail, item detail or bill message detail, this field contains the related sequence number for that record.
Bill Calc Header Group	A2	10 Bill calculation header 20 Bill calculation line 30 Bill calculation end
Line Seq	N3	Defines order of rows for similar record types
Filler	47	Only applicable for non-premise based records. (This makes up for the premise address related fields that are not applicable above).

Mailing Key

The following table defines the structure of the mailing key that appears on each bill print record.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on the bill routing record.

Bill Information

The topics in this section describe the information that appears on each on each bill print record.

- Global Extract Information Record (0010)
- Bill Record (0100)
- Account Balance Record (0150)
- Current Charge Record (0300)
- Payment Record (0600)
- Premise Record (0900)
- Service Agreement Record (1100)
- Bill Segment Record (1300)

Copyright © 2026, Oracle and/or its affiliates.

- Bill Calc Header Record (1500)
- Bill Calc line Record (1700)
- End Bill Calc Header Record (1900)
- Bill Segment Read Record (2100)
- Bill Segment SQ Record (2400)
- Bill Segment Item Record (2700)
- Bill Segment Message Record (2840)
- End Bill Segment Record (2900)
- Histogram Record (3100)
- Adjustment Record (3400)
- Adjustment Calc Lines Record (3420)
- End Service Agreement Record (3600)
- End Premise Record (3700)
- Message Record (3900)
- End Bill Record (9999)
- Address Sub-Record

Global Extract Information Record (0010)

This is a "header" record that contains information about the extract run as a whole.

> **Note**
>
> Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file.

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
BILL_CNT	N10	Count of bills on this extract. Multiple copies of a bill are counted multiple as well.
EXTRACT_DTTM	A26	System time of extraction. Also updated in bill routing table.
BILL_ID_START	A12	This is only used when a bill is produced as a result of an online request (from Bill - Main). It contains the Bill Id to extract.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
BILL_ID_END	A12	This is only used when a bill is produced as a result of an online request (from Bill - Main). It contains the Bill Id to extract.
THREAD_NBR	N10	This field is only used if bills are extracted in batch. It contains the thread number in which the bill was extracted.
THREAD_CNT	N10	This field is only used if bills are extracted in batch. It contains the total number of threads that were extracted.

Top of Page

Bill Record (0100)

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_BILL
ACCT_ID	A10	CI_BILL
BILL_DT	D (A10)	CI_BILL
DUE_DT	D (A10)	CI_BILL
COPY_NBR	N2	
NBR_BILL_COPIES	N2	CI_BILL_RTG
BILL_PRT_INTERCEPT	A8	CI_BILL_RTG
BILL_FORMAT_FLG	A2	CI_BILL_RTG
REPRINT_SW	A1	CI_BILL_RTG
CASH_ONLY_SW	A1	
CUR_BAL	S13.2	CI_BILL_SA
CUR_CHG_AMT	S13.2	CI_FT
CUR_CORR_AMT	S13.2	CI_FT
CUR_ADJ_AMT	S13.2	CI_FT
CUR_PAY_AMT	S13.2	CI_FT
CUR_PREV_BAL	S13.2	CI_FT
TOT_BAL	S13.2	CI_BILL_SA
TOT_CHG_AMT	S13.2	CI_FT
TOT_CORR_AMT	S13.2	CI_FT
TOT_ADJ_AMT	S13.2	CI_FT
TOT_PAY_AMT	S13.2	CI_FT
TOT_PREV_BAL	S13.2	Computed during extract.
CURRENCY_CD	A3	CI_BILL_SA
DELINQUENT_AMT	S13.2	Computed during extract.
CUST_PO_ID	A20	CI_BILL_RTG

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
ACCT_ENTITY_NAME	A254	CI_PER_NAME. Account Entity Name. Might include prefix/ suffix from Account/Person if available.
ENTITY_NAME	A254	CI_BILL_RTG. Mailing Entity Name.
ADDRESS_SBR	A1241	CI_BILL_RTG. Refer to its components in a later section.
FINAL_BILL_SW	A1	Computed during extract. Contains a value of Y if this bill is considered to be a "final" bill.
SCHED_EXTRACT_DT	D (A10)	CI_APAY_CLR_STG. This is the date that the automatic payment will be downloaded.
ENTITY_NAME1	A254	CI_BILL_RTG. Mailing Entity Name1.
ENTITY_NAME2	A254	CI_BILL_RTG. Mailing Entity Name2.
ENTITY_NAME3	A254	CI_BILL_RTG. Mailing Entity Name3.
ACCT-CUR-SYMBOL	A4	Currency symbol for the account's currency code.

Top of Page

Account Balance Record (0150)

Field Name	Format	Source/Value/Description
SPECIAL-ROLE-CD	A2	SA Type Special Role flag
SPECIAL-ROLE –DESCR	A30	Special Role description
CUR-AMT	S13.2	Derived from CI_FT (Include ALL non-canceled/non-closed SAs for the account/debt class, even those that do not have current charges)
TOT- AMT	S13.2	Derived from CI_FT (Include ALL non-canceled/non-closed SAs for the account/debt class, even those that do not have current charges)

Top of Page

Current Charge Record (0300)

Field Name	Format	Source/Value/Description
DEBT-CLASS-CD	A4	Debt Class of SA Type
DEBT-CLASS-DESCR	A30	Debt Class Description

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CUR-CHARGE-AMT	S13.2	Derived from CI_FT (Only the current charge bill segments.)
TOT-CHARGE-AMT	S13.2	Derived from CI_FT (Only the current charge bill segments.)

Top of Page

Payment Record (0600)

Field Name	Format	Source/Value/Description
FT TYPE	A2	Pay or Pay Cancel ('PS' or 'PX')
PAY-DT	D (A10)	Derived from CI_FT
CAN_RSN_CD	A4	Derived from CI_PAY, Only for cancellation
CAN-RSN-DESCR	A30	Derived from CI_PAY_CAN_RSN
CUR-PAY-AMT	S13.2	Derived from CI_FT.
TOT-PAY-AMT	S13.2	Derived from CI_FT.

Top of Page

Premise Record (0900)

Field Name	Format	Source/Value/Description
NON_PREM_SW	A1	Y/N switch to indicate (Y) special (non-premise) premise record. This is for non-premise service agreements such as charity or payment arrangement.
PREM_ID	A10	From CI_SA or CI_BSEG
ADDRESS_SBR	A1241	Service Address (Address of Char Premise) from CI_PREM. Refer to its components in a later section.

Top of Page

Service Agreement Record (1100)

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_SA
OLD_ACCT_ID	A36	CI_SA
CUST_READ_FLG	A2	CI_SA
SIC_CD	A4	CI_SA
SIC_DESCR	A30	CI_SIC
BUSINESS_UNIT	A10	CI_SA
SA_TYPE_CD	A8	CI_SA

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
SA_TYPE_DESCR	A30	CI_SA_TYPE
SA_TYPE_DFLT_DESCR	A80	CI_SA_TYPE
SVC_TYPE_CD	A30	CI_SA_TYPE
SVC_TYPE_DESCR	A30	CI_SVC_TYPE
REV_CL_CD	A8	CI_SA_TYPE
CHAR_PREM_ID	A10	CI_SA
SA_REL_ID	A10	CI_SA_REL
SA_REL_TYPE_CD	A8	CI_SA_REL
SA_REL_TYPE_DESCR	A30	CI_SA_REL_TYPE
SPR_CD	A12	CI_SA_REL
SPR_DESCR	A50	CI_SPR
SPR_ENTITY_NAME	A254	CI_PER_NAME (this is the service provider's person's name)

Top of Page

Bill Segment Record (1300)

Field Name	Format	Source/Value/Description
START_DT	D (A10)	CI_BSEG
END_DT	D (A10)	CI_BSEG
CLOSING_BSEG_SW	A1	CI_BSEG
EST_SW	A1	CI_BSEG
BILL_CYC_CD	A4	CI_BSEG
SQ_OVERRIDE_SW	A1	CI_BSEG
ITEM_OVERRIDE_SW	A1	CI_BSEG
BSEG_STAT_FLG	A2	CI_BSEG
CAN_RSN_CD	A4	CI_BSEG
CAN_RSN_DESCR	A30	CI_BILL_CAN_RSN
CUR_AMT	S13.2	CI_FT
TOT_AMT	S13.2	CI_FT

Top of Page

Bill Calc Header Record (1500)

Field Name	Format	Source/Value/Description
START_DT	D (A10)	CI_BSEG_CALC
END_DT	D (A10)	CI_BSEG_CALC
SETID	A10	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
CALC_AMT	S13.2	CI_BSEG_CALC

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
DESCR_ON_BILL	A254	CI_BSEG_CALC

Top of Page

Bill Calc line Record (1700)

Field Name	Format	Source/Value/Description
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_AMT	S13.2	CI_BSEG_CALC_LN
EXEMPT_AMT	S13.2	CI_BSEG_CALC_LN
UOM_CD	A30	CI_BSEG_CALC_LN
TOU_CD	A30	CI_BSEG_CALC_LN
SQI_CD	A30	CI_BSEG_CALC_LN
BILL_SQ	S12.6	CI_BSEG_CALC_LN
DESCR_ON_BILL	A254	CI_BSEG_CALC_LN

Top of Page

End Bill Calc Header Record (1900)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Bill Segment Read Record (2100)

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_BSEG_READ
BADGE_NBR	A16	D1_DVC_IDENTIFIER
UOM_CD	A30	CI_BSEG_READ
TOU_CD	A30	CI_BSEG_READ
RDG_DEC_POS	N1	CI_REG. Decimal positions of the reads.
START_REG_READING	S9.6	CI_BSEG_READ
START_READ_DTTM	T (A26)	CI_BSEG_READ
START_ESTIMATE_SW	A1	CI_REG_READ
END_REG_READING	S9.6	CI_BSEG_READ
END_READ_DTTM	T (A26)	CI_BSEG_READ
END_ESTIMATE_SW	A1	CI_REG_READ
MSR_QTY	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	N3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CONSUM_SUB_FLG	A2	CI_REG
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
REGISTER_CONST	S6.6	CI_BSEG_READ
QTY_DEC_POS	N1	CI_UOM. Decimal positions of the final quantity.
FINAL_UOM	A4	CI_BSEG_READ
FINAL_TOU	A8	CI_BSEG_READ
FINAL_REG_QTY	S12.6	CI_BSEG_READ
SQI_CD	A30	CI_BSEG_READ
FINAL_SQI_CD	A8	CI_BSEG_READ

Top of Page

Bill Segment SQ Record (2400)

Field Name	Format	Source/Value/Description
UOM_CD	A30	CI_BSEG_SQ
TOU_CD	A30	CI_BSEG_SQ
SQI_CD	A30	CI_BSEG_SQ
SQ_DEC_POS	N1	CI_BSEG_SQ
BILL_SQ	S12.6	CI_BSEG_SQ

Top of Page

Bill Segment Item Record (2700)

Field Name	Format	Source/Value/Description
START_DT	D (A10)	CI_BSEG_ITEM
END_DT	D (A10)	CI_BSEG_ITEM
ITEM_CNT	S9.2	CI_BSEG_ITEM
ITEM_DESCR	A60	CI_ITEM_TYPE

Top of Page

Bill Segment Message Record (2840)

Field Name	Format	Source/Value/Description
BILL_MSG_CD	A4	CI_BSEG_MSG
MSG_ON_BILL	A254	CI_BILL_MSG_L

Top of Page

End Bill Segment Record (2900)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Histogram Record (3100)

Field Name	Format	Source/Value/Description
UOM_CD	A30	CI_SA_TYPE
TOU_CD	A30	
SQI_CD	A30	
SQ_DEC_POS	N1	CI_UOM. Decimal positions of SQ.
SQ_CNT	N2	Count of SQ (maximum of 13.)
MAX_BILL_SQ	S12.6	Maximum Bill SQ from list. This is derived from CI_BSEG_READ.
SQ_DT (1:13)	D (A10)	Date (for label of the SQ). This is derived from CI_BSEG.
BILL_SQ (1:13)	S12.6 (1:13)	13 occurrences of QTY, one for each period ending on the SQ_DT. This is derived from CI_BSEG_READ.

Top of Page

Adjustment Record (3400)

Field Name	Format	Source/Value/Description
ADJ_DT	D (A10)	CI_FT
FT_TYPE_FLG	A2	CI_FT
CUR_AMT	S13.2	CI_FT
TOT_AMT	S13.2	CI_FT
ADJ_TYPE_CD	A8	CI_ADJ
ADJ_TYPE_DESCR	A30	CI_ADJ_TYPE
DESCR_ON_BILL	A254	CI_ADJ_TYPE
CAN_RSN_CD	A4	CI_ADJ
CAN_RSN_DESCR	A30	CI_ADJ_CAN_RSN

Top of Page

Adjustment Calc Lines Record (3420)

Field Name	Format	Source/Value/Description
APP_IN_SUMM_SW	A1	Appears in Summary Y/N switch derived from CI_ADJ_CALC_LN

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
RS_CD	A8	Rate Schedule derived from CI_ADJ_CALC_LN
CALC_AMT	S13.2	Calculated Amount derived from CI_ADJ_CALC_LN
EXEMPT_AMT	S13.2	Exempt Amount derived from CI_ADJ_CALC_LN
UOM_CD	A30	Unit of Measure derived from CI_ADJ_CALC_LN
TOU_CD	A30	Time of Use derived from CI_ADJ_CALC_LN
SQI_CD	A30	SQI derived from CI_ADJ_CALC_LN
BILL_SQ	S12.6	Billable Service Quantity derived from CI_ADJ_CALC_LN
DESCR_ON_BILL	A254	Description on Bill derived from CI_ADJ_CALC_LN
APP_IN_SUMM_SW	A1	Appears in Summary Y/N switch derived from CI_ADJ_CALC_LN

Top of Page

End Service Agreement Record (3600)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End Premise Record (3700)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Message Record (3900)

Field Name	Format	Source/Value/Description
BILL_MSG_CD	A4	CI_BILL_MSGS
MSG_PRIORITY_FLG	A2	CI_BILL_MSG
INSERT_CD	A8	CI_BILL_MSG
MSG_ON_BILL	A254	CI_BILL_MSG

Top of Page

End Bill Record (9999)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Address Sub-Record

The address sub-record that appears in both the bill record (0100) and the premise record (0900) is composed of the following:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A254	Refer to The Source Of Bill Routing Information for a description of conditions when this field will contain a fax number or an Email address rather than the first line of a postal address (this only occurs for a bill record (0100)).
ADDRESS2	A254	
ADDRESS3	A254	
ADDRESS4	A254	
CITY	A90	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A90	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

XML Bill Extracts

The system includes a sample bill extract algorithm that produces an XML document to interface to a document composition application. This is an alternative to the fixed flat file format described above. For additional details, see the algorithm type **Create XML bill extract records** (C1-BLEX-XML).

Algorithms of this type construct the XML file records that contain the information that appears on a printed bill. Each record in the extract is constructed using an extendable data area and may be extended by an implementation using the Post Processing Script parameter. The following steps should be followed to include additional elements in the extract:

- Create a data area that includes the additional elements for the record you wish to extend. This data area must reference the base product data area used in the extract. For example, to extend the bill record, create a data area that references data area C1-BillPrintBillRecord as its extended data area. See the C1-BLEX-XML algorithm type for a complete list of the data areas available.

Copyright © 2026, Oracle and/or its affiliates.

- Create a service script to retrieve the additional elements. This service script must be defined as the Post Processing Script on your algorithm and its schema must include the main bill extract data area (C1-BillPrintRecord).

For an example of how to extend the XML bill extract to include additional elements, see the CI_BillPrintXMLExtract export bundle included in the demo data.

Sequential Bill Numbers

Every bill is uniquely identified by its bill ID. Bill ID is a random, system-assigned number. In some locales, bills have an additional identifier known as a "sequential bill number". Sequential bill numbers are system-assigned, sequential numbers. For example, the first bill produced by the system is assigned a sequential bill number of 1, the next bill is assigned a value of 2, etc. An additional requirement is that no gaps may exist between the sequential bill numbers.

The topics in this section describe how sequential bill numbers are implemented in the system.

- Sequential Numbers Are Only Assigned If The Installation Option Is Turned On
- Pending Bills Do Not Have A Sequential Bill Number
- When / How Is The Sequential Bill Number Assigned
- Additional Functionality for Sequential Bill Numbering

Sequential Numbers Are Only Assigned If The Installation Option Is Turned On

Sequential bill numbers will only be assigned if you have selected Sequential Bill Numbers as the Alternative Bill ID Option on the Installation record.

Pending Bills Do Not Have A Sequential Bill Number

Bills are assigned a sequential bill number after they are completed. This means that pending bills will not have a sequential bill number. This is intentional as pending bills can be deleted and if a deleted bill had a sequential bill number, gaps would result. Refer to Bill Lifecycle for more information about a bill's state transition lifecycle.

When / How Is The Sequential Bill Number Assigned

> **Warning**
>
> This section is a technical explanation of how sequential bill numbers are assigned. It contains background information for those who wish to understand the rationale of the design.

The batch billing background process BILLING takes advantage of parallel processing to allow multiple bills to be produced concurrently. In addition, we allow groups of bills to be committed in "chunks" (based on the commit frequency assigned to the billing background process).

Copyright © 2026, Oracle and/or its affiliates.

These two facts make it impossible to allocate sequential bill numbers during batch billing (because back outs could result in gaps and parallel processing could result in duplicates). This is why sequential bill numbers are allocated as follows:

- A sequential bill number is allocated when a bill is completed online (by clicking the **Complete** button on Bill - Main).
- If you submit the batch billing process with the following parameters, sequential bill numbers will be allocated when bills are completed:
- **Batch thread number** must equal 1.
- **Batch thread count** must equal 1.
- **Override maximum records between commits** must equal 1.
  The above parameters mean that bills are not being produced in parallel and they are not being committed in "chunks". This allows the system to allocate sequential bill numbers when the bills are completed.
- If you do not run batch billing single-threaded, you must schedule the Assign Sequential Bill Numbers background process (ASSGNSBN) to run immediately after batch billing. This single-threaded process assigns sequential bill numbers to completed bills.
- The following measures are in place as a fail-safe:
- If the bill print download background process (POSTROUT) encounters a bill without a sequential bill number, it will abort with an appropriate error message.
- Users are not allowed to delete a completed bill with a sequential bill number. Note, deletion of completed bills is typically prohibited due to the presence of frozen bill segments. However, it's possible to create a bill without frozen bill segments and additional logic exist to inhibit the deletion of such bills if they have sequential bill numbers.

Additional Functionality for Sequential Bill Numbering

The following points describe additional features that are enabled when the Installation record indicates sequential bill numbers are used:

- The bill's number appears in "bill info". "Bill info" is the concatenated string of information that summarizes a bill on many pages in the system.
- The Bill Search page includes an additional search option to search by bill number.
- The match type algorithm (DSO BILL-ID) that allows payments to be matched to a given bill id will try to match the payment to a bill with a matching sequential bill number.

Document Numbers

Every bill is uniquely identified by its bill ID. Bill ID is a random, system-assigned number. In some locales utilities may be required to assign and print an additional document number. For example, in Argentina a utility must obtain authorization from the national tax agency before they are allowed to print bills at a local office. Once authorized, the utility is assigned a printing authorization code valid for a period. The printing authorization code and expiration, as well as an official document number are printed on the bill at the location where the bill is issued.

The topics in this section describe how document numbers are implemented in the system.

- Document Numbers Are Only Assigned If the Installation Option Is Turned On

Copyright © 2026, Oracle and/or its affiliates.

- Pending Bills Do Not Have a Document Number
- When/How Is the Document Number Assigned
- Additional Functionality Exists When Document Numbers Are Used

Document Numbers Are Only Assigned If the Installation Option Is Turned On

Document numbers will only be assigned if you have selected Document Numbers as the Alternative Bill ID Option on the Installation record. In addition, the Document Number (C1–DOC-NBR) and Document Number Details (C1–DOCNBR-DT) algorithms must be enabled on the Installation record. These algorithms contain the logic used by the system to assign a document number to a bill.

The logic used to assign a document number can be quite complex. For example, the document number may depend on the issuing center, the bill's document type (credit memo, invoice or debit memo), the account's VAT category and the CIS division. Depending on your implementations complexity, you may choose one of the following possible configurations for document numbers:

- Populate the Document Number, Document Number Details and Document Sequence Number algorithms on the Installation record. The system will behave as follows:
  - During online bill completion, a document type is derived and captured on the bill along with other required bill characteristics and a document number is assigned
  - During batch billing, a document type is derived and captured on the bill along with other required bill characteristics. Assigning a document number is deferred to the Assign Bill Document Numbers background process. The idea here is that all complex logic is executed during the multi-threaded batch billing process leaving the minimal amount of work for the single-threaded batch process that assigns the document number.
- Populate the Document Number and Document Number Details algorithms on the Installation record. Here, the system assigns a document number during online bill completion, but skips the assignment of document type and number during batch billing. The Assign Bill Document Numbers background process is responsible for deriving the bill's characteristics (document type, document classification, etc.) and assigning a document number.

Pending Bills Do Not Have a Document Number

Similar to sequential bill numbers, bills are assigned a document number after they are completed. This means that pending bills will not have a document number. Refer to Bill Lifecycle for more information about a bill's state transition lifecycle.

When/How Is the Document Number Assigned

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> This section is a technical explanation of how document numbers are assigned. It contains background information for those who wish to understand the rationale of the design.

- An installation algorithm assigns a document number to a bill.

Similar to sequential bill numbers, document numbers are allocated as follows:

- A document number is allocated when a bill is completed online (by clicking the **Complete** button on Bill - Main).
- The Assign Bill Document Numbers background process (C1-BNBAS) should be scheduled to run immediately after batch billing. This single-threaded process assigns document numbers to completed bills.
- The logic used to assign a document number resides in the installation options Document Number (C1–DOC-NBR) algorithm.
- The following measures are in place as a fail-safe:
- If the billing download process (POSTROUT) encounters a bill without a document number, it will abort with an appropriate error message.
- Users are not allowed to delete a completed bill with a document number. Note, deletion of completed bills is typically prohibited due to the presence of frozen bill segments. However, it's possible to create a bill without frozen bill segments and additional logic exist to inhibit the deletion of such bills if they have sequential bill numbers.

Additional Functionality Exists When Document Numbers Are Used

The following points describe additional features that are enabled when the Installation record that document numbers are used:

- The Bill Search page includes an additional search option to search by Document Number and Document Type.
- The bill's document number and type appear in "bill info". "Bill info" is the concatenated string of information that summarizes a bill on many pages in the system.
- The match type algorithm (C1–PDOV-PYBN) that allows payments to be matched to a given bill id will try to match the payment to a bill with a matching document number and type.

Digital Signatures

In some locales, billing software is required to generate and store digital signatures for each bill. Digital signatures are used to secure bills that are stored in the database, and the digital key for the bill will be known only to the vendor.

CCB supports digital signatures with the following features:

- An Encryption feature configuration must be configured with an algorithm info option for the key alias and signing algorithm

Copyright © 2026, Oracle and/or its affiliates.

- The installation algorithm, C1-BILLSIG, generates the digital signature for a bill and captures this on the bill itself. The algorithm also stores an alias of the private/public key pair on the Bill as a characteristic. See the installation algorithm, C1-BILLSIG, for a description of how the signature is generated.
- Bills are assigned a digital signature when they are completed.
  - The Assign Bill Document Numbers background process (C1-BNBAS) is responsible for generating digital signatures for a batch of bills. This batch job should be scheduled to run immediately after batch billing. See the description of the batch control and parameter settings for generating digital signatures.
  - A digital signature is also generated when a bill is completed online (by clicking the **Complete** button)

> **Note**
>
> The digital signatures features are only appropriate for customers who use document number in generating bill signatures.

For an example of the objects and instructions used for implementing this feature, see the CI_ BilllDocNumber & DigitalSignature export bundle included in the demo data.

For details about setting up the required keystore for this feature, see the Installation Guide.

Writing Off Bills

If your organization practices open-item accounting and your credit and collections rules revolve around collecting on unpaid bills, you can set up the system to automatically write-off bills.

If you've enabled this functionality, Bill - Main will contain a button that a user can click to manually write off a bill. A corresponding button exists that a user can click to reverse a written-off bill.

> **NOT_SUPPORTED**
>
> Refer to Writing Off Bills for a description of how to set up the system to allow both automated and manual bill write off.

Idiosyncratic Manual Bill Cancellation

If a bill's account is associated with a Customer Class that has a Cancel Bill algorithm, the system invokes this algorithm before the bill is displayed to determine if the bill is "cancellable". If the algorithm indicates the bill is cancellable, a button appears on Bill - Main. When clicked, the Cancel Bill algorithm is invoked again to cancel the bill. This functionality is meant to support unique cancellation needs required by some implementations; the sample base-package algorithm is empty.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Bills

A bill is used to communicate changes in a customer's financial obligations to the customer. The topics in this section describe how to maintain bills.

> **Note**
>
> **The system creates most bills behind-the-scenes.** Most bills are created by the system when it processes the bill cycle schedule. You should only have to access the bill pages to correct a bill or to add a bill real-time. For information about how the system creates bills, refer to The Big Picture of Billing.

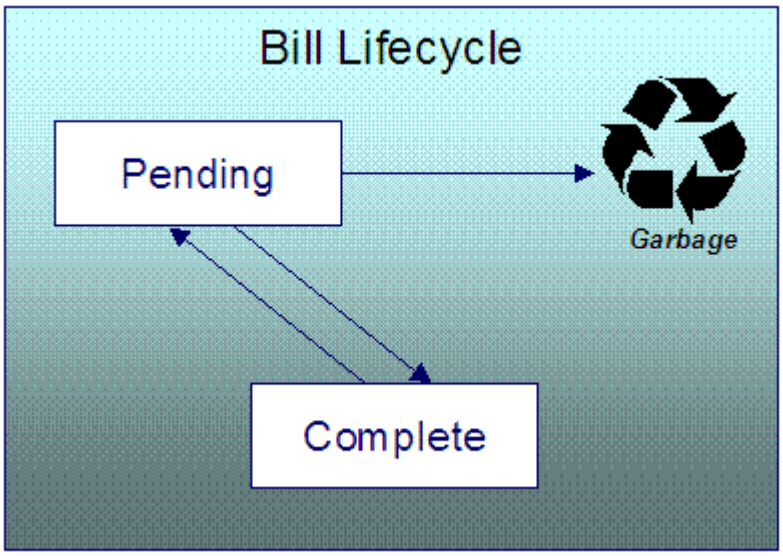
It's important to be aware that there are very few fields that are directly modifiable by a user. To modify most fields on a bill, you have to change source information (e.g., service agreement, rate, meter read) and then regenerate the bill. For example, if you want to change a bill's amount, you must cancel or add bill segments; you cannot change the bill's amount by modifying the bill amount field. Refer to How To for step-by-step instructions that explain how to perform common bill maintenance functions.

- Bill Lifecycle
- Bill - Main Information
- Bill - Bill Segments
- Bill - Bill Routings
- Bill - Bill Messages
- Bill - Characteristics
- How To
- Financial Transactions On A Bill

Bill Lifecycle

The following diagram shows the possible lifecycle of a bill.

Copyright © 2026, Oracle and/or its affiliates.



> **Warning**
>
> This explanation only makes sense in the context of the page used to maintain bills. Refer to Bill - Main Information for the details.

A bill is initially saved in the Pending state. You may create one or more bill segments for the account's service agreements at this point. Refer to How To for information about generating bill segments for the bill.

A bill becomes Complete when it is ready to be sent to the customer. Completing a bill triggers many things to occur. Refer to the section below for information about what happens when a bill is completed.

When you complete a bill, several things may happen:

- Pre-completion algorithms associated with the account's service agreements' SA types are executed. The system executes these algorithms first in the order of the billing processing sequence on each service agreement's SA type then in the order of the algorithm's sequence. These are not executed for the completion of credit notes.
- Pre-completion algorithms associated with the account's customer class are executed.
- The bill's due date is calculated. This is equal to the bill date plus the number of days defined on the account's customer class. If the resultant date is not a workday, the due date is set to the next workday. Note, this due date can be overridden if an override algorithm exists on the account's customer class.
- The bill's routing information is set up using Account - Person Information.
- If the Freeze At Bill Completion installation option has been turned on, freezable bill segments and adjustments linked to the account are frozen. Note, only freezable adjustments whose adjustment type indicates Freeze At Bill Completion will be frozen at this time. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for more information.
- If a final bill segment of a deposit service agreement is detected, the deposit is refunded to the customer using the refund method algorithm plugged-in on the deposit SA's deposit class. Refer to Deposit Refund for more information. This step is skipped during the completion of credit notes, or if the bill is linked to an off cycle bill generator.

Copyright © 2026, Oracle and/or its affiliates.

- Bill completion algorithms associated with the account's service agreements' SA types are executed. The system executes these algorithms first in the order of the billing processing sequence on each service agreement's SA type then in the order of the algorithm's sequence.
- Bill completion algorithms associated with the account's customer class are executed.
- Bill messages are amalgamated from various sources and linked to the bill. Refer to The Source Of Bill Messages for more information.
- Other financial transactions that have been frozen since the last bill and are marked to show on bill are linked to the new bill
- For open-item customers adjustments and corrections are linked
- For correction notes or credit notes, the bill cancellation financial transactions are linked
- For off cycle bill generator bills:
- Adjustments linked to the off cycle bill generator are linked to the bill
- If the off cycle bill generator is linked to a service agreement, then all non-payment financial transactions associated with the SA are linked to the bill
- For all other situations, payments, adjustments and corrections are linked assuming the adjustments are not linked to an off cycle bill generator
- All financial transactions that don't already have a user-defined aging date will be dated with the current date. In other words, they start aging from the date the bill is completed.
- If the account pays automatically and the installation's Autopay Creation Option is to Create At Bill Completion, an automatic payment is created. Refer to APAY-CREATE for more information about the auto pay creation algorithm. This step is skipped during the completion of credit notes.
- If the account's customer class indicates the account is an open-item customer, a match event is created. The bill's FTs and the automatic payment's FTs are linked to it. Refer to Payments and Match Events for more information about match events. This step is skipped during the completion of credit notes.
- If the account pays automatically and the installation's Autopay Creation Option is to Create On Extract Date, the bill is stamped with the date when the automatic payment is to be created. Refer to APAYCRET, the background process that creates the automatic payment, and APAYDSFR, the background process that distributes and freezes the automatic payment, for more information. This step is skipped during the completion of credit notes.
- The bill's late payment date is calculated. This is equal to the due date plus the grace days defined on the account's customer class. If the resultant date is not a workday, the LPC date is set to the next workday. This step is skipped during the completion of credit notes.
- The account is marked for review by the account debt monitor on its due date plus a grace period defined on its customer class. This step is skipped during the completion of credit notes.
- If the account's customer class indicates the account is an open-item customer, the system will create a match event if the new charges are offset by other financial transactions. Refer to Payments and Match Events for more information about match events. This step is skipped during the completion of credit notes.
- The bill's status becomes Complete.
- Post bill completion algorithms associated with the account's customer class are executed.

Copyright © 2026, Oracle and/or its affiliates.

If the system cannot complete the bill (for whatever reason), the bill remains Pending and an error message is shown on the main bill page. After correcting the cause of the problem, attempt to complete the bill again.

A complete bill may be changed back to pending using the Reopen button on Bill - Main Information. You would reopen a bill when

- Add more bill segments to a completed bill. Refer to the How To section for information about linking bill segments for a bill.
- Fine-tune the payments, adjustments, and corrections that were linked to a completed bill. Refer to the How To section for detailed instructions.

When you're happy with the bill, you can complete it again.

> **Note**
>
> **Cannot reopen historical bills.** You may only reopen an account's most recent bill (except if this bill is linked to an off cycle bill generator) because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.
>
> **Automatic payments.** If an automatic payment was created when the bill was completed and it has already been interfaced to the financial institution, you cannot reopen the bill. If the automatic payment exists, but it has not yet been interfaced, the system will automatically cancel the payment when you reopen the bill.
>
> **Certain bill completion algorithms prevent reopening.** If a bill contains a service agreement whose SA type has a "transfer accounts receivable to service provider" algorithm, the system will not allow the bill to be reopened. Refer to If You Deal With TBFU Service Providers You Can't Reopen Bills for more information about this bill completion algorithm.

You may delete a Pending bill from the database. You may not delete a pending bill if: a) there are frozen bill segments linked to the bill, or b) if financial transactions were linked to the bill (and this can only happen if the bill had been previously completed).

In addition to removing the bill, the system will also remove unfrozen bill segments linked to the bill.

Bill - Main Information

Open **Menu**, and then **Financial**, and then **Bill**, and then **Search** to maintain core bill information.

The following discussion describes the fields on this page. Refer to How To for a description of how to perform common bill maintenance functions.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance, etc.). If sequential bill numbering has been enabled, the bill's number also appears. If document numbering has been enabled, the assigned document number and document type will also appear. If this bill is a correction note or credit note, a message will appear indicating this along with the ID of the original bill. If the batch billing

Copyright © 2026, Oracle and/or its affiliates.

background process generates the bill, the bill information includes the bill cycle and window start date. Bills generated manually, i.e., off-schedule, do not have this information.

Bill ID is the system-assigned unique identifier of the bill.

Account ID identifies the customer who is responsible for the bill. The name of the main customer on the account appears adjacent.

If this bill is a credit note or correction note, a message appears adjacent to the account information indicating that this bill is a credit note or correction note for another bill; this hyperlink will drill you to the original bill. If a credit note or correction note has been issued for one or more of this bill's bill segments, a message appears indicating such. **Bill Status** is the bill's status. Refer to Bill Lifecycle for the potential values and how to handle a bill when it exists in a given state. Adjacent to the bill status is the **Display Bill** button, which will display an online image of the bill when clicked. Refer to How To Display A Bill On-line for more information.

If the bill is associated with an off cycle bill generator, a message appears adjacent to the bill status indicating this and the hyperlink will take the user to the related off cycle bill generator.

> **Note**
>
> You can only use the **Display Bill** button if your system has been configured to display an on-line image; otherwise, a message indicating that the service is not available will appear. This option can only be configured by your technical staff. The system comes with configurations to display bill images using a document composition application such as *BI Publisher* or to retrieve a bill image from an external source such as a bill print vendor. This feature can also work with other applications.

Due Date is the date on which the bill is due. A bill's due date is calculated as follows:

- Add the due days specified on the account's customer class to the bill date.
- The resulting date is the due date unless it's not a workday. In this case, the due date is set to the next workday (workdays are defined in the installation options tables).
- If the account's customer class has an override due date calculation algorithm, the due date may be overridden. Refer to Setting Up Customer Classes for more information.

> **Note**
>
> **Late payment charges.** Late payment charges are calculated on the due date plus the grace period. Refer to Defining Late Payment Charge Options for more information.

Bill Date is the business date that was used when the bill was completed. For example, if batch billing starts on 11:55 pm on 1-Jan-2000 and a bill is produced at 1:00 am on 2-Jan-2000, this bill's bill date would be 1-Jan-2000.

Create Date/Time is the date and time on which the bill was originally created.

Completion Date/Time is the date and time on which the bill was completed.

If the customer pays automatically and the automatic payment has not yet been created, a brief message appears highlighting the date and amount of the future automatic payment. In

Copyright © 2026, Oracle and/or its affiliates.

addition, a button appears called **Stop Autopay**, that, when clicked, causes the associated automatic payment to stop. You might click this button if the customer indicates that they will send in a payment. After you click the **Stop Autopay** button, a message appears highlighting the stoppage. Refer to How And When Are Automatic Payments Created for more information about how to setup the system to defer the creation of automatic payments until the future automatic payment extract date.

In the **Message** area, a brief error message appears if there's a problem with the bill. The message area is suppressed if there are no problems with the bill.

Click the magnifying button to view the long explanation. The long explanation will provide information about the cause of the error (and how to fix it).

The **Late Payment Charge Details** section contains information about how the system determines when and whether to levy late payment charges:

- **Eligible for Late Payment Charge** can be checked if a charge should be levied for late payment. This field is defaulted based on the customer's customer class. If you want to excuse a bill from late payment charge processing uncheck this box.
- **Late Payment Charge Date** is the date on which the system will levy a late payment charge if payment is not received. This date is calculated when the bill is completed. It is equal to the bill's due date plus the account's customer class's grace days. If this date does not fall on a workday, the date is set to the next available workday when the bill is completed. If you want to manually delay the onset of late payment charge processing, extend this date into the future.

> **NOT_SUPPORTED**
>
> **Late payment charges.** Late payment charges are calculated on the due date plus the grace period. Refer to How Late Payment Charges Get Calculated for more information.

Bill Summary summarizes the financial impact of the bill. The information that appears differs depending on whether the account is a balance-forward or open-item account. If the account is a balance-forward customer, the following information appears:

- **Previous Period's Balance** is the balance from the account's last bill.
- **Total Payments** is the total amount of frozen or canceled payment segment financial transactions linked to this bill. Frozen payment segments appear as negative numbers (decreasing the amount owed by the customer), while canceled payment segments appear as positive numbers (increasing the amount owed). You can drill down on this balance to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Payments From A Completed Bill for a description of how to perform common maintenance functions.
- **Total Adjustments** is the total amount of frozen or canceled adjustment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Adjustments From A Completed Bill for a description of how to perform common maintenance functions.
- **Total Bill Corrections** is the total amount of canceled and / or rebilled bill segment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total.

Copyright © 2026, Oracle and/or its affiliates.

- **Total Current Billing Charges** is the total amount of frozen bill segment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total.
- **Ending Balance For This Period** is the sum of the five amounts above. It is the amount owed by the customer as of the end of the bill period.

If the account is an open-item customer, the following information appears under **Bill Summary**:

- **New Charges** is the total amount of frozen bill segment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total.
- **Adjustments** is the total amount of frozen or canceled adjustment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Adjustments From A Completed Bill for a description of how to perform common maintenance functions.
- **Corrections** is the total amount of canceled and / or rebilled bill segment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total.
- **Total** is the sum of the above amounts. You can drill down on this value to view the financial transactions that contribute to this total.

If the account is an open-item customer, the following information appears under **Match Summary** (note, this section is not displayed for balance-forward accounts):

- X **Balanced Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to balanced match events.
- X **Unbalanced Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to unbalanced match events.
- X **Disputed Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to disputed match events.
- X **Unmatched Item(s)** contains the count and total amount of financial transactions linked to this bill that are not linked to any match event.

Total Changes After Completion only appears if bill segments that appeared on the original bill have been subsequently canceled or rebilled. The amount displayed is the net effect of all cancels and rebills. Note, you can use the **Remarks** column in the bill segments grid (to the right) to see which bill segments have been cancelled / rebilled.

If 25 or fewer bill segments are linked to this bill, the grid in the lower right portion of the page contains one row for every bill segment linked to the bill.

The following points describe each column in the grid:

- The **Bill Segment** column contains a concatenation of each bill segment's division, SA type, status and bill period. The division and SA type come from the bill segment's service agreement.
- The **Current Amount** column contains the bill segment's effect on the service agreement's current balance.
- The **Status** column shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values.
- The **Remarks** highlight special situations. The following remarks may appear:

Copyright © 2026, Oracle and/or its affiliates.

- Rebill after completion appears if the bill segment is a rebill that was created after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
- Rebill prior completion appears if the bill segment is a rebill that was created before the bill was sent to the customer.
- Canceled after completion appears if the bill segment was canceled after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
- Canceled prior completion appears if the bill segment was canceled before the bill was sent to the customer.
- If the bill segment's status is Error or Incomplete, this column contains an estimated value of this bill segment. The estimated value is calculated by computing an average daily amount from the service agreement's frozen bill segments in the last 365 days. This average is then applied to the number of days in this bill segment's bill period. If the bill segment doesn't have a bill period (rare, but possible), the historical average is not displayed.
- If the bill segment's status is Error, this column also contains the error message.
- Rebilled due to correction appears if a correction note was generated for the bill segment. This remark appears only if you have the Correction Note Billing Option specified in the Installation record.
- Rebilled due to correction. Canceled after completion appears for a correction note that spawns another correction note. This remark appears only if you have the Correction Note Billing Option specified in the Installation record.

If more than 25 bill segments are linked to this bill, the grid in the lower right portion of the page contains a summary of the various bill segments linked to the bill.

The following points describe each column in the grid:

- The **Status** column shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values. Click on the hyperlink to transfer to the Bill Segments tab where the associated bill segments can be viewed.
- The **Total Bill Segments** column contains the number of bill segments linked to the bill with this **Status**. If the status is Error or Incomplete, this column contains an estimated value of these bill segments. The estimated value is calculated by computing an average daily amount from frozen bill segments in the last 365 days for each service agreement. This average is then applied to the number of days in each bill segment's bill period. Please note, an historical average cannot be computed if an Error or Incomplete bill segment does not have a bill period (rare, but possible).
- The **Total Amount** column contains the sum of the current amount of these bill segments.

At the bottom of the scroll is shown the **Total Generated Charge**. This represents the total amount of the bill segments. This amount may differ from the Total Current Billing Charges when:

- The bill is not Complete. A bill segment's financial effect is not shown on the bill until the bill is completed.
- A bill segment has been canceled / rebilled since the bill was completed.

The topics that follow describe each of the actions that appear in the **Bill Segment Action** and **Bill Action** areas. Refer to How To for a description of typical business processes that use these buttons.

- Generate

Copyright © 2026, Oracle and/or its affiliates.

- Freeze
- Cancel Frozen
- Freeze / Complete
- Complete
- Delete
- Reopen
- Credit Note
- Correction Note
- Write Off
- Reverse Write Off
- Cancel Bill

Generate

The **Generate** button causes a bill segment to be created for each service agreement linked to the account. The system generates bill segments in the order of the billing processing sequence on each service agreement's SA type.

This button is enabled when you are in add mode (i.e., you are not displaying an existing bill) and you have selected an **Account ID**.

When clicked, the **Generate** window opens.

You must specify the following parameters in the **Generate** window to generate the bill segments:

- **Cutoff Date** is the last possible day of each bill segments bill period. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** defines the financial period to which the new bill segments' financial transactions are booked. The current date defaults when the window opens.
- Check **Allow Estimates** if the system is allowed to estimate consumption if it can't find a meter read.
- If the service agreements use meter reads, use the **Billing Option** to control how the system retrieves meter reads that appear on the bill. Choose Use Cut off Date if the meter reads must exist on the **Cutoff Date** (defined above). Choose Use Schedule if you want the service agreement's service point(s) service cycle schedule to control the meter reads that are used (the schedule effective on the cutoff date is used to select the meter reads).

After specifying the parameters, click **Calculate** to create a new bill and new bill segments for the account. You see a summary of the bill segments in the lower right portion of the page.

Freeze

Clicking **Freeze** causes all freezable bill segments to become frozen. Refer to Bill Segment Lifecycle for more information about freezing.

This button is enabled if:

Copyright © 2026, Oracle and/or its affiliates.

- the Freeze At Will **Bill Segment Freeze Option** on the installation option has been selected AND
- freezable bill segments exist AND
- there are no error, incomplete or pending cancel bill segments.

> **Note**
>
> **Freezing a day or more after generation.** If, during freezing, bill segments in closed accounting periods are detected, a pop-up appears in which you can specify a new accounting date; this date is updated onto the offending bill segments. This only happens if the accounting period closes before you freeze the bill segments (the accounting date is stamped on a bill segment when it's generated).

If, after freezing, you're ready to send the bill to the customer click Complete to finalize the bill.

> **Note**
>
> **If problems are detected after freezing.** A bill segment may not be changed after it is frozen. All subsequent changes must occur by canceling the frozen bill segment and creating a new bill segment. Refer to How To Cancel A Bill Segment and How To Cancel / Rebill A Bill Segment for more information.

Cancel Frozen

Clicking **Cancel Frozen** causes all frozen bill segments to become canceled. Refer to Bill Segment Lifecycle for more information about cancellation. This button is disabled if the bill is written off.

> **Note**
>
> **Pending cancels are not affected.** Clicking **Cancel Frozen** does not affect bill segments that are pending cancel. To make a pending cancel bill segment canceled, transfer to Bill Segment - Main and click Cancel.

If you need to cancel a specific bill segment, either transfer to the next tab and cancel the bill segment in question or follow the instructions under How To Cancel A Bill Segment.

This button is enabled if frozen bill segments are linked to the bill.

When clicked, the **Cancel Frozen Bill Segments** window opens.

In the **Cancel Frozen Bill Segments** window, you must specify the following parameters to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.

Copyright © 2026, Oracle and/or its affiliates.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments.

> **Note**
>
> **There is no Undo.** Unlike cancellations performed on Bill Segment - Main, there is no Undo action. This transaction moves the pending cancel bill segments to the cancel state immediately; whereas the bill segment transaction lets you examine the cancellation before you commit it. If you cancelled bill segments by mistake, you must generate and freeze new bill segments.

If the bill is pending, the cancellation causes the bill segments to be suppressed on the version of the bill sent to the customer (but they remain in the database for audit and financial reporting purposes).

If the bill is complete and you do not Reopen the bill, the financial impact of the canceled bill segments appears on the next bill created for the account (under Bill Corrections).

If the bill is complete you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a frozen bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system marks the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

Freeze / Complete

The **Freeze / Complete** button performs a variety of functions:

- all freezable bill segments (including rebills) become frozen
- all pending cancel bill segments become canceled
- all freezable adjustments whose adjustment type indicates Freeze At Bill Completion become frozen
- the bill is finalized and marked for transmission to the customer

Refer to Bill Lifecycle for information about what happens during bill completion. Refer to Bill Segment Lifecycle for more information about freezing.

This button is enabled if:

- the Freeze At Bill Completion Bill Segment Freeze Option on the installation option has been selected AND
- the bill is pending AND
- there are no bill segments or there are bill segments and all are freezable, pending cancel, frozen or canceled AND
- there are no error or incomplete bill segments.

When clicked, the **Complete** window opens.

If the **User May Override Bill Date** option has been enabled on the Installation Record, you may override the **Bill Date** prior to completion. Otherwise, the **Bill Date** is the current date and cannot be changed. The current date defaults when the window opens.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Freezing a day or more after generation.** If, during freezing, bill segments in closed accounting periods are detected, a pop-up appears in which you can specify a new accounting date; this date is updated onto the offending bill segments. This will only happen if the accounting period closes before you freeze the bill segments (the accounting date is stamped on a bill segment when it's initially generated).

After specifying the parameter, click **Calculate** to freeze and complete the bill.

Complete

The **Complete** button causes a bill to be finalized and marked for transmission to the customer. Refer to Bill Lifecycle for information about what happens during bill completion.

This button is enabled if:

- the Freeze At Will Bill Segment Freeze Option on the installation option has been selected AND
- the bill is pending AND
- there are no bill segments or there are bill segments and all are frozen or canceled.

When clicked, the **Complete** window opens.

If the **User May Override Bill Date** option has been enabled on the Installation Record, you may override the **Bill Date** prior to completion. Otherwise, the **Bill Date** is the current date and cannot be changed. The current date defaults when the window opens. If completing a correction note, the user is prompted to enter a **Correction Note Date instead of a bill date** in the **Complete** window.

After specifying the above parameters, click **Calculate** to complete the bill.

Delete

The **Delete** button deletes a bill and its bill segments.

This button is enabled if:

- the bill is pending AND
- all bill segments are incomplete, freezable or in error.

Reopen

The **Reopen** button causes a bill to be reopened. Refer to Bill Lifecycle for information about why you would reopen a bill.

This button is enabled if:

- the bill is complete AND

Copyright © 2026, Oracle and/or its affiliates.

- the bill is not written off AND
- this is the most recent bill produced for the account AND
- the bill is not linked to an off cycle bill generator AND
- reopening hasn't been prohibited for the bill. Refer to Bill Lifecycle for information about conditions that can prevent a bill from reopening.

After reopening a bill, it returns to the pending state and you can make changes to the account's financial transactions (e.g., add payments, adjustments, and bill segments). After which, you click Complete or Freeze / Complete to finalize the bill. Only one of these buttons is shown - the specific one depends on how your organization has set the **Bill Segment Freeze Option** on the installation record.

Credit Note

The **Credit Note** button is used to cancel ALL bill segments and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND
- the bill is not written off AND
- The bill has frozen bill segments.

To create credit notes for all frozen bill segments, click the **Credit Note** button without selecting any bill segments. A system message asks for confirmation that credit notes will be created for all frozen bill segments.

To create credit notes for specific frozen bill segments, select just those bill segments.

When clicked, the **Credit Note** window opens.

You must specify the following parameters in the **Credit Note** window to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details appear. The current date defaults when the window opens.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments and create a credit note.

It should be noted that if this operation is successful, the new "negative bill" is complete. It is ready for transmission to the customer and for interface to the general ledger.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If you want to issue a credit note for a specific bill segment.** Rather than cancel all bill segments on a bill, you can create a credit note for a specific bill segment on a bill. To do this, drill down on the bill segment in question and then click the Credit Note button on the bill segment page. You can also issue a credit note for selected bill segments on the next tab by selecting the desired bill segments and clicking the **Credit Note** button.

Correction Note

The **Correction Note** button is used to cancel and rebill ALL bill segments and create a new bill that contains the cancellation and rebill details. We refer to the resultant bill as a correction note. Refer to Correction Notes - Presenting Correction Notes on a Separate Bill for more information.

This button is enabled if:

- the **Correction Note** Billing Option is specified on the Installation record AND
- the bill is complete AND
- there is at least one frozen bill segment on the bill or all of the bill's bill segments are frozen or cancelled AND
- there are no bill segment financial transactions that have been transferred out of the system.

When clicked, the **Correction Note** window opens.

You must specify the following parameters in the **Correction Note** window to correct the bill segment:

- **Cancel Reason** defines why you are performing the correction.
- **Cutoff Date** is the last possible day of each bill period for a bill segment. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** defines the financial period that the financial transactions are booked for a bill segment. The current date defaults when the window opens.
- Check **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- Use the **Billing Option** to define how the system looks for meter reads:
  - Select Use Schedule if the system should select reads based on the SA's SP measurement cycle schedule.
  - Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
- If you prefer to have the system use the consumption details used on the original bill segments, check **Use Old Read/Item/SQ**. If this switch is on, neither **Allow Estimates** nor **Billing Option** is relevant.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The **Use Old Read/Item/SQ** switch is not available if the bill's SA references an SA type with a special role flag of Bill Determinants Required.

- After specifying the parameters, click **Calculate** to correct the bill segments and create a correction note. The Bill Page opens and displays the Correction Note.

Once you complete a correction note, it cannot be reopened.

After you create a correction note, the **Undo Correction Note** button is activated.

The button is enabled if:

- the **Correction Note** Billing Option is specified on the Installation record AND
- the bill is a correction note AND
- the bill is not linked to an off cycle bill generator AND
- all of the bill segments for the correction note are freezable.

> **Note**
>
> **If you want to issue a correction note for a specific bill segment.** You can create a correction note for a specific bill segment on a bill. To do this, drill down on the bill segment in question and then click the **Correction Note** button on the bill segment page. You can also issue a correction note for selected bill segments by selecting the desired bill segments and clicking the **Correction Note** button.

Write Off

Refer to Writing Off Bills for a description of when the **Write Off** button appears and what happens if the button is clicked.

Reverse Write Off

Refer to Writing Off Bills for a description of when the **Reverse Write Off** button appears and what happens if the button is clicked.

Cancel Bill

Refer to Idiosyncratic Manual Bill Cancellation for a description of when the **Cancel Bill** button appears and what happens if the button is clicked. Note that if the bill is linked to an off cycle bill generator, this button is not enabled.

Bill - Bill Segments

Copyright © 2026, Oracle and/or its affiliates.

You can use this page to view all or selected bill segments linked to a bill. In addition, you can also perform certain maintenance functions on this page (see the details below).

Open **Menu**, and then **Financial**, and then **Bill**, and then **Search** and navigate to the **Bill Segments** page to view this information.

> **Note**
>
> If the bill has more than 25 bill segments and you don't use the go to buttons in the bill segment grid on the Main tab to drill over to this information, the search criteria are intentionally left blank in order to avoid retrieving all bill segments (with the resultant slow response times). You must therefore use the **SA Filter** and the **Status Filter** to define the type of bill segments that should be retrieved. See the Description of page below for more information about this page's search criteria.

The Description of page section below describes the fields on this page. Refer to How To for a description of how to perform common maintenance functions.

Description of page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance, etc.).

Bill ID is the system-assigned unique identifier of the bill.

> **Note**
>
> If the bill has more than 25 bill segments and you don't use the hyperlinks in the bill segment grid on the Main tab to drill over to this page, the **Filters** are intentionally left blank in order to avoid retrieving all bill segments (with the resultant slow response times). You must therefore use the **SA Filter** and the **Status Filter** to define the type of bill segments that should be retrieved.

The area beneath **Bill Info** provides you with options that control which bill segments appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements whose bill segments appear in the grid. The following options are available:
- Address. Use this option to restrict bill segments to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you do not wish to restrict bill segments based on service agreement attributes.
- Geographic Type. Use this option to restrict bill segments to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.
- SA Type. Use this option to restrict bill segments to those whose service agreements are linked to a given **CIS Division** and **SA Type**.
- Use **Status Filter** to restrict the bill segments based on their status. The following options are available:
- All. This option shows all bill segments regardless of status.

Copyright © 2026, Oracle and/or its affiliates.

- Refer to Bill Segment Lifecycle for the various status values.

Don't forget to click the search button after changing the filters.

The **Select All** / **Clear All** buttons are used to select bill segments if you plan on issuing any of the mass update actions at the bottom of the page. These buttons are enabled if either of the following conditions is true:

- All bill segments that appear in the grid are in the Error, Freezable and/or Incomplete states.
- All bill segments that appear in the grid are in the Frozen state

Refer to the description of the "mass update" actions below for more information.

> **Note**
>
> **20 bill segments at a time.** Clicking **Select All** selects the first 20 bill segments in the grid. If more than 20 bill segments exist, you must select them in batches.

The grid that follows contains the bill segments that match your search criteria. The following information appears in the grid:

- Select box. See the mass update description immediately above for conditions under which this checkbox can be used to select bill segments for mass update actions.
- The **Bill Segment** column contains a concatenation of the bill segment's division, SA type, status and bill period. The division and SA type come from the bill segment's service agreement. Click the hyperlink to transfer to Bill Segment - Main Information. On this page you can perform maintenance functions (e.g., cancel/rebill, delete, cancel, etc., depending on the segment's status) on the bill segment in question.
- The **Current Amount** column contains the bill segment's effect on the service agreement's current balance.
- The **Status** shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values.
- The **Remarks** column highlights special circumstances associated with the bill segment.
- Rebill after completion appears if the bill segment is a rebill that was created after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
- Rebill prior completion appears if the bill segment is a rebill that was created before the bill was sent to the customer.
- Canceled after completion appears if the bill segment was canceled after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
- Canceled prior completion appears if the bill segment was canceled before the bill was sent to the customer.
- If the bill segment's status is Error or Incomplete, this column contains an estimated value of this bill segment. The estimated value is calculated by computing an average daily amount from the service agreement's frozen bill segments in the last 365 days. This average is then applied to the number of days in this bill segment's bill period. If the bill segment doesn't have a bill period (rare, but possible), the historical average is not displayed.

Copyright © 2026, Oracle and/or its affiliates.

- If the bill segment's status is Error, this column also contains the error message.
- Rebilled due to correction note appears if the correction note was generated for the bill segment. This remark is generated only if you have Correction Note specified in the Installation record.
- Rebilled due to correction. Canceled after completion appears for a correction note that spawns another correction note. This remark is generated only if you have Correction Note specified in the Installation record.
- The **Service Quantity** and **UOM** (unit of measure) columns contain the amount of consumption that was billed. The UOM is defined on the service agreement's SA type.
- The **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

This transaction has sophisticated logic that can be used to perform "mass updates" to the bill segments that appear in the grid (using the buttons at the bottom of the page). The following points describe these mass update actions:

- Generate (Bill - Bill Segments)
- Freeze (Bill - Bill Segments)
- Delete (Bill - Bill Segments)
- Cancel/Rebill/Freeze (Bill - Bill Segments)
- Cancel (Bill - Bill Segments)
- Credit Note (Bill - Bill Segments)
- Correction Note (Bill - Bill Segments)

Generate (Bill - Bill Segments)

The **Generate** button is used to delete and recreate one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the Error, Freezable and/or Incomplete state.

Note that you can only select a bill segment for regeneration if ALL bill segments in the grid are in the Error, Freezable and/or Incomplete states. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Generate** window opens.

You must specify the following parameters in the **Generate** window to regenerate a new bill:

- **Accounting Date** defines the financial period to which the new bill segments' financial transactions will be booked. The current date defaults when the window opens.
- Check **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it can't find a meter read.
- If the service agreements use meter reads, use the **Billing Option** to control how the system retrieves meter reads that appear on the bill. Choose Use Cut off Date if the meter reads must exist on the **Cutoff Date** (defined above). Choose Use Schedule if you want the service agreement's service point(s) measurement cycle schedule to control the meter reads that are used.

After specifying the parameters, click **Calculate** to regenerate the selected bill segments. A summary of what the system does is presented next.

Copyright © 2026, Oracle and/or its affiliates.

Freeze (Bill - Bill Segments)

The **Freeze** button is used to freeze one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the Freezable state AND the installation option indicates users are allowed to "freeze at will".

Note that you can only select a bill segment for freezing if ALL bill segments in the grid are in the Freezable state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Freeze** window opens.

Accounting Date is used to define the financial period to which the frozen bill segments' financial transactions are booked. The current date defaults when the window opens.

After specifying the above parameters, click **Freeze** to freeze the selected bill segments. A summary of what the system does is presented next.

Delete (Bill - Bill Segments)

The **Delete** button is used to delete one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the Error, Freezable and/or Incomplete state.

Note that you can only select a bill segment for deletion if ALL bill segments in the grid are in the Error, Freezable and/or Incomplete states. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

If the system is successful in deleting the selected bill segment(s), a confirmation window summarizes what the system just did (i.e., it shows the number of bill segments that were deleted).

Cancel/Rebill/Freeze (Bill - Bill Segments)

If problems are detected with a frozen bill segment, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you'll probably need to fix the cause of the problem. The following table describes common problems and how to fix them.

Problem	How To Fix It
The wrong rate was used	Go to Service Agreement - Rate Info and correct the rate.
The wrong meter was connected to the service point	
A contract rider was applied incorrectly	Go to Service Agreement - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Service Agreement - Rate Info and correct the tax exemption information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multi-Cancel/Rebill Saves Time.** If the cause of the problem impacts many bill segments related to a service agreement, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once. Also note that for service agreements that require bill determinants from Oracle Utilities Meter Data Management, this transaction may only be used to cancel / rebill a single bill segment at a time.
>
> **Credit Notes**. Using the cancel/rebill functionality causes the cancel and rebill details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the **Rebill** functionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

The **Cancel/Rebill/Freeze** button is used to cancel, rebill and freeze one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the Frozen state AND the bill is not written off.

Note that you can only select a bill segment for cancel/rebill/freeze if ALL bill segments in the grid are in the Frozen state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Cancel/Rebill/Freeze** window opens.

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** defines the financial period to which the canceled and new bill segments' financial transactions are booked. The current date defaults when the window opens.
- Check **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- Use the **Billing Option** to define how the system looks for meter reads:
  - Select Use Schedule if the system should select reads based on the SA's SP measurement cycle schedule.
  - Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
  - If you prefer to have the system use the consumption details used on the original bill segments, check **Use Old Read/Item/SQ**. If this switch is on, neither **Allow Estimates** nor **Billing Option** is relevant.

> **Note**
>
> The **Use Old Read/Item/SQ** switch is not available if the bill's SA references an SA type with a special role flag of Bill Determinants Required.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> If the cause of the rebill is an incorrect rate and the new rate requires service quantities that were not calculated when the bill was originally calculated, do not turn on **Use Old Read/Item/SQ**. Why? Because you want the system to derive new service quantities during the rebill process.

After specifying the parameters, click **Calculate** to cancel, rebill and freeze the selected bill segments.

> **Note**
>
> **Errors.** Bill segment errors may occur when you cancel / rebill the bill segments. For example, if you change the SA's rate to a rate that requires a unit-of-measure that doesn't exist on the SA's meter, a bill segment error is generated. If this occurs, you must go to Bill Segment - Main for the pending cancel bill segment and use the **Undo** action to remove the offending bill segment (and restore the original segment to the Frozen state). Alternatively, you can drill down on the Error bill segment and use the **Regenerate** action after correcting the cause of the problem.
>
> **There is no Undo.** Unlike cancel / rebills performed on Bill Segment - Main, there is no Undo action. This is because this transaction freezes the newly created bill segments after it cancel/rebills; whereas the bill segment transaction lets you examine the new bill segment before you freeze it. If you made a mistake, simply correct the cause of the mistake and then cancel / rebill again. The erroneous financial transactions are automatically suppressed on the next bill produced for the customer.

A confirmation window summarizes what the system just did (i.e., it shows the number of bill segments that were deleted / generated).

Cancel (Bill - Bill Segments)

The **Cancel** button is used to cancel one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the Frozen state AND the bill is not written off.

Note that you can only select a bill segment for cancellation if ALL bill segments in the grid are in the Frozen state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Cancel** window opens.

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** is used to define the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.

After specifying the above parameters, click **Calculate** to cancel the selected bill segments.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **There is no Undo.** Unlike cancellations performed on Bill Segment - Main, there is no Undo action. This transaction moves the pending cancel bill segments to the cancel state immediately; whereas the bill segment transaction lets you examine the cancellation before you commit it. If you cancelled bill segments by mistake, you must generate and freeze new bill segments.

Credit Note (Bill - Bill Segments)

The **Credit Note** button is used to cancel selected bill segments and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND
- if you select at least one row from the bill segments grid that's in the Frozen state AND
- the bill is not written off.

Note that you can only select a bill segment for this action if ALL bill segments in the grid are in the Frozen state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Credit Note** window opens.

You must specify the following parameters in the **Credit Note** window to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details appear. The current date defaults when the window opens.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments and create a credit note.

It should be noted that if this operation is successful, the new "negative bill" is complete. Refer to Bill Lifecycle for information about the steps that take place during completion. It is ready for transmission to the customer and for interface to the general ledger.

> **Note**
>
> **If you want to issue a credit note for an entire bill.** Rather than cancel a specific bill segment, functionality exists that allows you to create a credit note for ALL bill segments on a bill. To use this feature, display the bill in question and then click the Credit Note button on the bill page.

Copyright © 2026, Oracle and/or its affiliates.

Correction Note (Bill - Bill Segments)

The **Correction Note** button is used to cancel and rebill bill segments and create a new bill that contains the cancellation and rebill details. We refer to the resultant bill as a correction note. Refer to Correction Notes - Presenting Correction Notes on a Separate Bill for more information. This button is enabled if:

- the **Correction Note** option is specified on the Installation record AND
- the bill is complete AND
- the bill segment is frozen AND
- there are no bill segment financial transactions that have been transferred out of the system.

When clicked, the **Correction Note** window opens.

You must specify the following parameters in the **Correction Note** window to correct the bill segment:

- **Cancel Reason** defines why you are performing the correction.
- **Cutoff Date** is the last possible day of each bill segments bill period. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** defines the financial period to which the corrected bill segments' financial transactions are booked. The current date defaults when the window opens.
- Check **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- Use the **Billing Option** to define how the system looks for meter reads:
  - Select Use Schedule if the system should select reads based on the SA's SP measurement cycle schedule.
  - Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
- If you prefer to have the system use the consumption details used on the original bill segments, check **Use Old Read/Item/SQ**. If this switch is on, neither **Allow Estimates** nor **Billing Option** is relevant.

> **Note**
>
> The **Use Old Read/Item/SQ** switch is not available if the bill's SA references an SA type with a special role flag of Bill Determinants Required.

- After specifying the parameters, click **Calculate** to correct the bill segments and create a correction note. The Bill Page opens and displays the Correction Note.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If you want to issue a correction note for an entire bill.** You can create a correction note for ALL bill segments on a bill. To use this feature, display the bill in question on the Bill page and click the **Correction Note** button.

Bill - Bill Routings

This page is used to view the recipients of a bill. Refer to Printing Bills for a discussion of how this information is used to produce bill images.

This page is also used to request a new copy of a bill. Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for a description of how to do this.

> **Note**
>
> **No routing information until completion.** A bill has no routing details until it is completed. At completion time, the system creates a routing detail for every person linked to the account that receives a copy of its bills. Refer to The Source Of Bill Routing Information for more information about how this information is constructed.

Use **Menu**, and then **Financial**, and then **Bill**, and then **Search** and navigate to the **Bill Routings** page.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance). If sequential bill numbering has been enabled, the bill's number also appears.

Bill ID is the system-assigned unique identifier of the bill.

The **Bill Routing Information** scroll contains an entry for each recipient of the bill. This information is disabled after the bill has been routed to the recipient. The remainder of this discussion describes the fields that can be defined for each recipient of the bill.

Person ID identifies the person who was originally associated with the routing record. You may change the **Person ID** to any person in the system as this information is only used to default the recipient's name and address information, below.

The **Bill Routing Parameters** area contains information explaining how and when the bill is routed to the recipient. This information is gray when the rill routing information has been extracted or if a bill is reopened:

- **Bill Route Type** is the method used to transmit the bill to the customer. If the **Person ID** is linked to the account, the bill route type defaults from the person's Account - Person Information.
- **Sequence** is the system-assigned identifier assigned to this bill routing.
- **Batch Control** is the identifier of the batch of bills in which the recipient's copy was downloaded. The batch process is defined on the **Bill Route Type** or CIS Division.

Copyright © 2026, Oracle and/or its affiliates.

- **Extract Date/Time** is the date and time the bill was downloaded. This information only appears after the download has commenced.
- **Format** indicates if the customer should receive a Detailed or a Summary bill. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **Reprint** is checked if a user created this routing. This switch is populated by the system and is always protected.
- Turn on **Do Not Extract** if the bill should not be routed to the individual for whatever reason.
- **Copies** indicates the number of copies of the bill the person receives. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **PO ID** is the purchase order Id the customer wants printed on their copy of the bill. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **Intercept** is the User ID of the individual who wants to review the bill before it is sent to the customer. This field is automatically populated when the account has a Bill Print Intercept specified (intercept information is defined on Account - Main Information). For a one-time intercept, a user can enter their ID in this field before the bill, or reprint, has been extracted.

> **Note**
>
> Adding a User ID to the Intercept field will not automatically populate the Bill Print Intercept field on the Account's Main page; nor, will it affect future bill routings.

The **Mailing Address Information** area contains the recipient's name and address.

- If the recipient has an override mailing name , **Name1**, **Name2** and **Name3** contain the person's override mailing name. Otherwise, **Name1** contains the recipient's primary name (this name might also include a prefix or suffix if the related Account - Person relationship has these fields defined).
- The remaining fields are the recipient's address. Refer to The Source Of Bill Routing Information for information about how the system constructs the recipient's address. Note:
- If the recipient's address is a mailing address (as opposed to fax or email), the **Country** associated with the address controls the fields that appear in the address. Refer to Defining Countries for more information on the address constituents.
- If a user manually adds a routing record, the mailing address information defaults from the information on the current routing record.
- If a user specifies a **Person ID** that is linked to the bill's account, the **Bill Route Type** defined on the Account - Person record is defaulted.
- If a user specifies a **Person ID** that is not linked to the bill's account, the **Bill Route Type** on the installation record is defaulted. Note, if the bill route indicates the bill is routed via the postal service, the system will only be able to default the person's address if the person has an override mailing address on their correspondence information.
- If the user changes the **Bill Route Type** after the initial default, the system populates the address information as described under The Source Of Bill Routing Information.
- The Validate button on the bill page will only be available if you have set the Allow Address Validation option type to Y on the General System Configuration feature configuration. Once clicked, the Address Validation Script on the General System Configuration feature

Copyright © 2026, Oracle and/or its affiliates.

configuration will be invoked to validate the entered address. Refer to Implementing Address Validation for more information.

Bill - Bill Messages

The Messages page is a grid containing one row for every message that appears on the bill. Open **Menu**, and then **Financial**, and then **Bill**, and then **Search** and navigate to the **Bill Messages** page to view this information.

> **Note**
>
> **No messages until completion.** A bill has no messages until it is completed (unless you insert them manually). At completion time, the system assembles messages from the various message sources. Refer to The Source Of Bill Messages for information about these sources.
>
> **The bill segment(s) may also have messages.** Be aware that only account-oriented messages are linked to a bill. There may also be service agreement-oriented messages linked to the bill's bill segments. Refer to Bill - Bill Messages for information about the page on which service agreement-oriented messages are displayed. Refer to The Source Of Bill Messages for information about the various message sources and whether each is linked to a bill or a bill segment.
>
> **Bill messages may contain dynamic information.** Refer to Substituting Field Values Into A Bill Message for more information.

The **Description of Page** section below describes the fields on this page. Refer to How To Add Ad Hoc Messages To A Bill for a description of how to perform common maintenance functions.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance). If sequential bill numbering has been enabled, the bill's number also appears.

Bill ID is the system-assigned unique identifier of the bill.

The following columns are displayed in the grid:

- **Bill Message** is the code that identifies the bill message.
- **Message On Bill** is the message associated with the code.
- **Priority** is the bill messages priority (on the printed bill).
- **Insert Code** defines if the bill message causes an insert in the envelope.

Bill - Characteristics

The Characteristics page is a grid containing one row for each characteristic linked to the bill. Bill characteristics are purely informational.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can only choose characteristic types defined as permissible on a bill record. Refer to Setting Up Characteristic Types and Their Values for more information.

Open **Menu**, and then **Financial**, and then **Bill**, and then **Search** and navigate to the **Characteristics** page to view this information.

Description of Page

The following fields display on the grid:

- **Characteristic Type** must be a type of characteristic defined as permissible on a bill record.
- **Sequence** Controls the order in which characteristics of the same type are displayed.
- **Characteristic Value** indicates the value of the characteristic if the characteristic type is pre-defined or foreign key reference.

How To

The topics in this section describe how to perform common bill maintenance functions.

- How To Create A Bill For All Service Agreements Linked To An Account
- How To Create A Bill For A Specific Service Agreement
- How To Create a Bill with no Bill Segments
- How To Create A One-Time Invoice
- How To Correct A Bill Segment That's In Error
- How To Correct A Bill That's In Error
- How To Cancel A Bill Segment
- How To Cancel / Rebill A Bill Segment
- How To Create A Credit Note
- How To Create A Correction Note
- How To Create An Off Cycle Bill Generator
- How To Override The Items Billed On A Specific Bill Segment
- How To Remove Unwanted Adjustments (or Payments) From A Completed Bill
- How To Add An Adjustment (or Payment) To A Completed Bill
- How To Reprint A Bill (For The Original Recipients or For Someone New)
- How To Add Ad Hoc Messages To A Bill
- How To Remove A Completed Bill
- How To Display A Bill On-line

Copyright © 2026, Oracle and/or its affiliates.

How To Create A Bill For All Service Agreements Linked To An Account

99.9% of all bills are created by the batch bill process and require no human intervention before they are finalized and sent to a customer. The other 0.1% are created by users on-line / real time. This section explains how to create the 0.1%.

The following points describe how to use this page to create a new bill:

- Navigate to the Bill - Main Information page:
- If you have already selected the customer (say on Control Central), select **Go To Bill**, and then **Add** option on the account context menu. context menu. This will transfer you to this page with the respective account populated.
- If you haven't selected the account, use Menu to navigate to this page (**Menu**, and then **Financial**, and then **Bill**, and then **Add**) and then select the **Account ID** using the adjacent search button.
- Click the Generate button to create a new bill.
- You will see a summary of the bill segments in the lower right portion of the page. If there are errors, refer to How To Correct A Bill Segment That's In Error. If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:
- If this option is set to Freeze At Bill Completion, click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, click the Complete button.

How To Create A Bill For A Specific Service Agreement

Most bills contain a bill segment for every service agreement linked to the account. If you want to produce a bill that contains bill segments for a subset of service agreements (e.g., you want to create a one-time invoice for an account's billable charge service agreement), follow these steps:

- Navigate to the Bill - Main Information page:
- If you have already selected the customer (say on Control Central), select **Go To Bill**, and then **Add** option on the account context menu. context menu. This will transfer you to this page with the respective account populated.
- If you haven't selected the account, use Menu to navigate to **Menu**, and then **Financial**, and then **Bill** , and then **Add,** and then select the **Account ID** using the adjacent search button.
- Click the save button to create a bill without bill segments for the account.
- Select Go To Bill Segment + option on the Bill's context menu to transfer to Bill Segment - Main.
- Select the **SA ID** of the service agreement for which you want to create a bill segment.
- Click the Generate button to generate a bill segment.
  - If an error exists, refer to How To Correct A Bill Segment That's In Error.
  - If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:

Copyright © 2026, Oracle and/or its affiliates.

- If this option is set to Freeze At Bill Completion, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

How To Create a Bill with no Bill Segments

You may wish to generate a bill that contains no bill segments. For example, perhaps your customer is billed quarterly but pays monthly. During the months where no charges are calculated, you may choose to send a "bill" that does not show new charges, but includes the payments received. To produce a bill with no bill segments, follow these steps:

- Navigate to the Bill - Main Information page:
- If you have already selected the customer (say on Control Central), select **Go To Bill**, and then **Add** option on the account context menu. context menu. This will transfer you to this page with the respective account populated.
- If you haven't selected the account, use Menu to navigate to this page (**Menu**, and then **Financial**, and then **Bill**, and then **Add**) and then select the **Account ID** using the adjacent search button.
- Click the save button to create a bill without bill segments for the account.
- At this point either the **Complete** or **Freeze / Complete** button (based on the value of Bill Segment Freeze Option on installation options) becomes enabled. Click the applicable button for your installation to complete the bill. At this point, all the steps described for completing a bill in Bill Lifecycle are performed, including sweeping onto the bill any payments received since the last bill was generated.

How To Create A One-Time Invoice

The following points describe the steps necessary to create and bill a one-time invoice:

- If the customer to be invoiced doesn't already have an appropriate billable charge service agreement, create one. The easiest way to do this is from Start/Stop Service.
- Use Maintaining Billable Charges to add a billable charge for the service agreement. The billable charge contains the invoice lines and amounts. The system will produce the bill for this charge when the account's bill cycle next executes. If you can't wait until that time, proceed to the next step.
- If you created a new billable charge service agreement in step 1, you must activate it before billing can bill it. To do this, transfer to Service Agreement - Main Information and click the activate button.
- Refer to How To Create A Bill For A Specific Service Agreement for instructions describing how to create a bill with a bill segment for the billable charge service agreement.

How To Correct A Bill Segment That's In Error

A bill segment will exist in the Error state if a problem was encountered during the generation of a bill segment. To fix an error bill segment you must correct the cause of the error and then regenerate the bill segment. Refer to Bill Segment Errors for background information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Error segments created during batch billing.** If the error bill segment was created as part of the batch bill process, the system attempts to fix the offending segment(s) each night during the account's bill window by regenerating it. Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And once a bill is error-free, it will be completed and sent to the customer. Refer to Batch Billing for more information.

Correcting the cause of a bill segment error can be difficult or trivial; we hope that the error messages (along with their long descriptions) provide a good description of how to resolve the issue.

After you have corrected the cause of the error, you should regenerate the bill segment. There are two ways to do this:

- You can use the Generate action on Bill - Bill Segment to regenerate one or more bill segments. We recommend using this method if the cause of the error affected many bill segments because you can regenerate many bill segments at a time on this page. Also, be aware that this transaction shows the estimated value of error bill segments so you can gauge whether to invest a lot of time in correcting the problem. For example, if the bill segment's estimated value is $10,000, it's probably worth correcting the problem before sending the bill out. However, if the bill segment's value is small, you may decide to simply delete the error bill segment and send the bill without it (the system will attempt to create a bill for the deleted period the next time the account is billed).
- You can use the Generate action on Bill Segment - Main to regenerate an individual bill segment. We'd recommend using this method if the problem is bill segment-specific and you need to examine the information that is snapshot on the bill segment to resolve the error (e.g., you need to look at the meter reads and service quantities).

> **Note**
>
> **Drilling Into Bill Segment Errors.** When you use the bill segments in error To Do list to work through errors, you will be transferred to Bill Segment - Main with the bill segment displayed.

After all errors have been resolved on a bill, return to Bill - Main. The **Bill Segment Freeze Option** on the installation option controls the next step:

- If this option is set to Freeze At Bill Completion, click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, click the Complete button.

How To Correct A Bill That's In Error

Besides errors on bill segments, there may also be errors detected when the system attempts to complete a bill. For example, if the system cannot find a mailing address, the bill will be in error (as opposed to one of its bill segments). Refer to Bill Errors for background information.

Copyright © 2026, Oracle and/or its affiliates.

To correct such an error:

- Navigate to Bill - Main.
- Look at the bill's error message. Push the view button to view the long explanation that will provide suggestions as to the cause of the error (and how to fix it). Transfer to the appropriate page to fix the problem.
- After fixing the cause of the error, return to Bill - Main. The **Bill Segment Freeze Option** on the installation option controls the next step:
- If this option is set to Freeze At Bill Completion, click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, click the Complete button.

How To Cancel A Bill Segment

If a bill segment was frozen and it should never have been created in the first place, the bill segment must be canceled. There are several ways to do this:

- You can use the Cancel Frozen action on Bill - Main to cancel all frozen bill segments linked to the bill. We recommend using this method if you need to cancel everything on a bill.
- You can use the Cancel action on Bill - Bill Segments to cancel selected bill segments. We recommend using this method if you need to cancel multiple bill segments.
- You can use the Cancel action on Bill Segment - Main to cancel an individual bill segment. We'd recommend using this method if you need to examine the information on the bill segment before canceling it.

If the related bill is pending, the cancellation will cause the bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is complete and you do not Reopen the bill, the financial impact of the canceled bill segment(s) will appear on the next bill created for the account (under Bill Corrections).

If the related bill is complete you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a frozen bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

How To Cancel / Rebill A Bill Segment

If a bill segment was frozen but it contains incorrect information, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you must correct the cause of the problem. The following table describes common problems and where to go to fix them.

Problem	How To Fix It

Copyright © 2026, Oracle and/or its affiliates.

The wrong rate was used	Go to Service Agreement - Rate Info and correct the rate.
The wrong meter was connected to the service point	
A contract rider was applied incorrectly	Go to Service Agreement - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Service Agreement - Rate Info and correct the tax exemption information.

After correcting the cause of the problem, you're ready to cancel and rebill the offending bill segment(s). There are several ways to do this:

- You can use the Cancel / Rebill / Freeze action on Bill - Bill Segments to cancel, rebill and freeze selected bill segments. We recommend using this method if you need to cancel / rebill multiple bill segments due to a pervasive problem.
- You can use the Rebill action on Bill Segment - Main to cancel and rebill an individual bill segment. We'd recommend using this method if you need to examine the information on the bill segment before and after canceling it. If you use this method, you must also use the Freeze action to freeze the newly created bill segment.
- If the cause of the problem impacts many bill segments related to a service agreement, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once. Also note that for service agreements that require bill determinants from Oracle Utilities Meter Data Management, this transaction may only be used to cancel / rebill a single bill segment at a time.

If the related bill is pending, the cancellation will cause the erroneous bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is complete and you do not Reopen the bill, the financial impact of the canceled bill segment(s) and their rebill bill segment(s) will appear on the next bill created for the account (under Bill Corrections).

If the related bill is complete you may be able to Reopen the bill and then Complete it. By doing this, you can suppress an erroneous frozen bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel / rebill the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

How To Create A Credit Note

A credit note is used to cancel bills segments and present the results of this cancellation on a separate bill. This separate bill is referred to as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Installation option must be turned on.** If your organization creates credit notes, be sure to select the Credit Note Billing Option on the Installation Options. If you don't select this option, the credit note buttons will not be enabled.

Before creating a credit note, you must first determine whether a single bill segment must be canceled or whether all bill segments for a bill must be canceled:

- If you need to issue a credit note for the entire bill, navigate to the Bill page and click the Credit Note button.
- If you need to issue a credit note for a specific bill segment, navigate to the Bill Segment page and click the Credit Note button.

> **Note**
>
> **Open-item customers**. If any of the bill segments are associated with an open-item customer class AND the original bill segment is an open item, the match event will be created and balanced with the original FT and the cancellation FT, assuming you have the appropriate FT freeze algorithm plugged-in on the customer class.
>
> **Issuing a credit note for a specific bill segment**. You may also issue a credit note for an individual bill segment. When you create a credit note for an individual bill segment, that bill segment is canceled and a new credit note bill is created for that single bill segment's cancellation information.

How To Create A Correction Note

The **Correction Note** button is used to cancel and rebill bill segments and present the cancellation and rebill details on a separate bill. Refer to Correction Notes - Presenting Correction Notes on a Separate Bill for more information.

> **Note**
>
> **Installation option must be turned on.** If your organization creates correction notes, be sure to select the Correction Note Billing Option on the Installation Options. If you don't select this option, the correction note buttons will not be enabled.

Before creating a correction note, you must first determine whether the note should impact a single bill segment or whether all bill segments for a bill must be corrected:

- If you need to issue a correction note for the entire bill, navigate to the Bill page and click the Correction Note button.
- If you need to issue a correction note for a specific bill segment, navigate to the Bill Segment page and click the Correction Note button.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Open-item customers**. If any of the bill segments are associated with an open-item customer class AND the original bill segment is an open item, the match event will be created and balanced with the original FT and the cancellation FT, assuming you have the appropriate FT freeze algorithm plugged in on the customer class.
>
> **Issuing a correction note for a specific bill segment**. You can also issue a correction note for an individual bill segment. When you create a correction note for an individual bill segment, that bill segment is canceled and a new correction note bill is created for that single bill segment's cancellation information.

How To Create An Off Cycle Bill Generator

An off cycle bill generator is used to create an off cycle bill for an account. You control the financial transactions that are linked to the bill. To create a bill using the base off cycle bill generator business object, follow these steps:

- With an account in context, select **Account Context Menu**, and then **Go To Off Cycle Bill Generator +.** *(See* context menu *for more information about Context Menus.)* This will transfer you to this portal with the respective account populated.
- Select the adjustments, billable charges and/or service agreements that you wish to include in the bill.
- Enter the required bill information to be used when creating the bill; this includes the Cutoff Date, Accounting Date, Allow Estimates switch, Billing Option and Bill Date.
- Click the save button.
- Click the Complete button to transition the off cycle bill generator to the complete state and create the bill. Any errors encountered while attempting to create the bill will be captured in the off cycle bill generator's log.
- Refer to Off Cycle Bill Generator for more information.

How To Override The Items Billed On A Specific Bill Segment

The How the System Requests How Much Was Consumed used by the customer is one of the first steps that happens when a bill segment is created. 99.9% of the time, the system does a perfect job in this effort. However, there are rare circumstances when, due to missing or incorrect data, the quantity calculated by the system is not correct.

To help you deal with these rare situations, a "back door" exists that allows a bill segment's billable quantities to be overridden. This truly is a "back door" because the proper way to fix the amount of a commodity is to correct the cause of the error and then cancel / rebill the bill segment.

This section describes how to override the total number of items billed on a bill segment. You would do this in the rare situation when you cannot get the item snapshot to reflect the amount of consumption you need to bill.

- Display the error or freezable bill segment on Bill Segment - Main.

Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> You can only override consumption on a bill segment that is in Error or Freezable. This means you must generate a bill segment before you can override the item details.

- Transfer to the Bill Segment - Item Details tab.
- Insert one row in the item grid (click the + button) for each item you need to add. You may also change item amounts that were populated by the system when it initially generated the bill.
- Save the bill segment.
- Generate the bill segment.
- If the bill segment still looks wrong, repeat the above points until the bill segment is correct.
- The **Bill Segment Freeze Option** on the installation option controls the next step:
- If this option is set to Freeze At Bill Completion, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

How To Remove Unwanted Adjustments (or Payments) From A Completed Bill

If the system (or a user) has created adjustments that have been swept onto a recently completed bill but you don't want them to appear on the bill (e.g., the system generated a late payment charge but you want to forgive it) perform the following steps:

- Cancel the adjustment (Adjustment - Main Information).
- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will see that an adjustment that was swept onto the bill when it was originally completed has been canceled and it will mark the adjustment for suppression on the printed bill.

You can remove unwanted payments using an analogous approach - Cancel the payment, Reopen the bill, Complete the bill.

> **Note**
>
> **Cannot reopen historical bills.** You may only reopen an account's most recent bill because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.

How To Add An Adjustment (or Payment) To A Completed Bill

If you want to add an adjustment to a completed bill perform the following steps:

Copyright © 2026, Oracle and/or its affiliates.

- Add and freeze the adjustment (Adjustment - Main Information).
- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will sweep the recently created financial transactions onto itself.

You can add a payment to a bill using an analogous approach - Add the payment, Reopen the bill, Complete the bill.

> **Note**
>
> **Cannot reopen historical bills.** You may only reopen an account's most recent bill because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.

How To Reprint A Bill (For The Original Recipients or For Someone New)

A copy of a bill is sent to every individual / business specified in the bill's routing details. The system creates the routing details when a bill is completed. Refer to The Source Of Bill Routing Information for more information about how this information is compiled during bill completion.

If you want to reprint a bill OR if you want to send a bill to someone other than the original recipient(s):

- Navigate to the Bill - Main page.
- Transfer to the Bill - Bill Routings tab.
- Insert a new row (click the + button). Note well, you can only insert information on a bill that is Complete.
- Specify the name and address to which the bill should be sent. If you want to send the bill to someone other than the original recipient, simply specify the individual's name and address. If the recipient is a person in the system, you can select their person ID and let the system default this information for you.
- Save the bill.

> **Note**
>
> If a bill is Complete but the print process hasn't yet executed, you don't have to insert a new routing row. Rather, you can simply change the name and address on the original routing details.

How To Add Ad Hoc Messages To A Bill

If you want to add a message to a completed bill:

- Navigate to the Bill - Main page.

Copyright © 2026, Oracle and/or its affiliates.

- Transfer to the Bill - Bill Messages tab.
- Insert a new row (click the + button). Note, you can only insert a message on the most recent bill that is Complete.
- Specify the desired message code.
- Save the bill.
- Reprint the bill as per How To Reprint A Bill (For The Original Recipients or For Someone New).

NOT_SUPPORTED

Refer to The Source Of Bill Messages for more information about the messages the system automatically links to a bill when the bill is completed.

How To Remove A Completed Bill

You can't physically remove a completed bill from the system. However, you can remove all financial transactions from the bill (and therefore end up with a zero value bill). You can also suppress the bill from being printed. To do this, perform the following steps:

- If the bill has bill segments, cancel each bill segment. Refer to How To Cancel A Bill Segment for instructions describing how to do this.
- If the bill has payment amounts or adjustment amounts that were entered incorrectly, navigate to the payment and/or adjustment pages and cancel the payments / adjustments.
- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Transfer to the Bill - Bill Routings tab. Turn on the Do Not Print switch for every routing entry.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will sweep the canceled segments onto it and the net effect should be zero current charges.

Note

Cannot reopen historical bills. You may only reopen an account's most recent bill because recompleting the bill causes the ending balance to change, and we don't want this to happen to historical bills.

If the bill has payment amounts or adjustment amounts that are valid, but you don't want them to appear on this bill, navigate to the **Menu**, and then **Financial**, and then **Bill**, and then **Search** and navigate to the **Financial Details** page. Drill over to each respective financial transaction by clicking the drill down button. Clicking this button transfers you to Financial Transaction - Main. On this page, clear out the value of the Bill ID. Clearing out the Bill ID unlinks the financial transaction from the bill. When the system next completes a bill for the account, it will find the unbilled financial transaction and link it to the bill.

Copyright © 2026, Oracle and/or its affiliates.

How To Display A Bill On-line

There are multiple ways to display a bill. Each requires the use of the Installation Option — Online Bill Display plug-in spot.

- One method is to make a call to retrieve a bill image from an external source. Many bill print vendors support this method and support REST or web service calls to request the bill image, often in the PDF format. The system is delivered with an algorithm that can construct a URL to retrieve a bill image.
- Another method uses software to render the bill in a format suitable for displaying. You must configure the system to invoke this software; otherwise, you will get a message indicating that the service is not available. This option can only be configured by your technical staff. The system can be configured to work with different bill production software, and is delivered with an algorithm that can display a bill on-line using BI Publisher or a document composition application. With a document composition application, the bills are presented in PDF format.

To view a bill on-line, click the **Display Bill** button. For any method that present the bill as a PDF, you must have Adobe® Acrobat® Reader installed.

Financial Transactions On A Bill

Open **Menu**, and then **Financial Query**, and then **Financial Transactions On A Bill** to view this information.

> **Note**
>
> You can also open this page using the go to buttons that prefix the financial transaction summaries on Bill - Main.

Description of page

Bill Id is the system-assigned unique identifier of the bill.

Account ID is the bill's account.

The area beneath **Account ID** provides you with options that control which financial transactions appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
  - Address. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code.
  - All. Use this option if you do not wish to restrict financial transactions based on service agreement attributes.
  - Geographic Type. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.

Copyright © 2026, Oracle and/or its affiliates.

  - SA ID. Use this option to restrict financial transactions to those of a specific service agreement.
  - SA Type. Use this option to restrict financial transactions to those whose service agreements are linked to a given **CIS Division** and **SA Type**.
- Use **FT Type Filter** to restrict the type of financial transactions that appear in the grid. The following options are available:
  - Adjustments. This option shows all financial transactions associated with adjustments.
  - All. This option shows all financial transactions.
  - Corrections. This option shows all financial transactions associated with corrections (i.e., cancel / rebills that have occurred after the bill was completed).
  - Current Charges. This option shows all financial transactions associated with bill segments (that aren't Corrections).
  - Payments. This option shows all financial transactions associated with payments.
- Use **Match Event Status Filter** to restrict the financial transactions based on the status of their match event. This filter only appears if the bill's account is an open-item customer. The following options are available:
  - All. This option shows all financial transactions.
  - Balanced. This option shows all financial transactions whose match event is balanced.
  - Disputed. This option shows all financial transactions whose match event is disputed.
  - Unbalanced. This option shows all financial transactions whose match event is unbalanced.
  - Unmatched. This option shows all financial transactions that are not linked to a match event.

> **Warning**
>
> Don't forget to click the search button after changing the filters or after selecting a new Bill ID.

The grid that follows contains the financial transactions that match your search criteria. The following information is displayed:

- **Match Event Status** shows the status of the financial transaction's match event. This column only appears if the bill's account is an open-item customer.
- **FT Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown. Click on the hyperlink to transfer to Financial Transaction - Main. On this page, you can change certain aspects of the FT in question. Refer to How To Remove Unwanted Adjustments (or Payments) From A Completed Bill for a description of how to perform common maintenance functions.
- **Accounting Date** is the date the system uses to determine the financial transaction's accounting period in your general ledger.
- **Current Amount** contains the financial transaction's effect on the service agreement's current balance.
- **Payoff Amount** contains the financial transaction's effect on the service agreement's payoff balance. The **Payoff Amount** will be dim if it equals the **Current Amount**.

Copyright © 2026, Oracle and/or its affiliates.

- **Show on Bill** indicates if information about the financial transaction appears on the customer's bill.
- **SA Information** contains a summary of the respective service agreement.
- **Financial Transaction ID** is the system-assigned unique identifier of the financial transaction.

At the bottom of the page is a summary of the financial transactions that match the search criteria.

Maintaining Bill Segments

A bill typically contains one bill segment for every active service agreement linked to its account. A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

The actions taken to create a bill segment are dependent on the service agreement's SA type. For example, a service agreement that charges for metered service is produced as follows:

- The system requests for the amount or usage consumed (or billing determinants) from Oracle Utilities Meter Data Management (e.g., the number of lamps, the number of gallons of water, the peak kW, etc.). Refer to How the System Requests How Much Was Consumed for the details.
- The service agreement's rate is applied to the amount consumed to calculate how much the customer owes for the service. A bill segment and a financial transaction are generated to reflect the results of this calculation. Refer to How Rates Affect The Information On Bill Segments for the details. Refer to Effective Dates & Price Proration for information about how the system prorates changes to rates and prices during a bill period.

> **Note**
>
> **SA type controls everything.** The above points explain how a bill segment is produced for a **bill determinants required** service agreement. It's important to be aware that the system has other bill segment creation algorithms. For example, if the service agreement charges for a deposit, the system neither amalgamates consumption nor does it apply a rate. You may be wondering how the system knows which method to use? It uses the bill segment type on the service agreement's SA type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

It's important to be aware that there are very few fields that are directly modifiable by a user. To modify most fields on a bill segment, you have to change source information (e.g., service agreement, rate, meter read) and then regenerate the bill segment. For example, if you want to change the rate used to calculate a bill segment, you must change the rate history on the respective service agreement and then cancel / rebill the bill segment. Refer to How To for step-by-step instructions that explain how to perform common bill segment maintenance functions.

- Bill Segment Lifecycle
- Bill Segment - Main Information
- Bill Segment - Read Details

Copyright © 2026, Oracle and/or its affiliates.

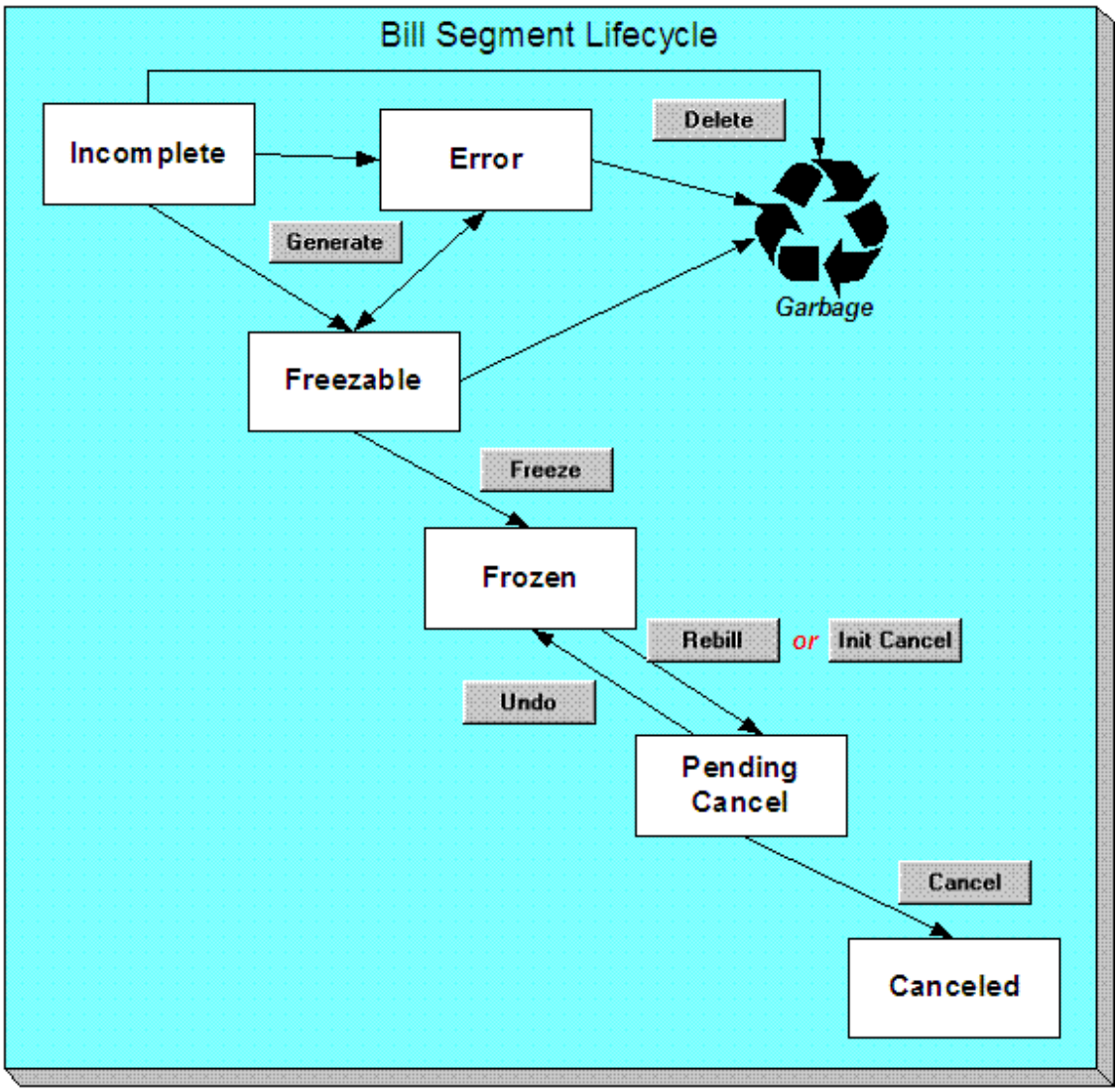
- Bill Segment - Item Details
- Bill Segment - SQ Details
- Bill Segment - Calc Lines
- Bill Segment - Financial Details
- Bill Segment - Bill Segment Messages
- Bill Segment - Sub Bill Segments
- Bill Segment - Bill Segment Portal
- Bill Segment - Usage
- Financial - Bill Segment Calculation Line Characteristics

Bill Segment Lifecycle

The following diagram shows the possible lifecycle of a bill segment.

> **Warning**
>
> This diagram only makes sense in the context of the page used to maintain bill segments. Refer to Bill Segment - Main Information for the details.

Copyright © 2026, Oracle and/or its affiliates.



Bill segments are initially created in the Incomplete state. Bill segments in this state don't have bill lines or a financial transaction; they are simply a stub awaiting generation.

Click **Generate** to generate the bill segment's bill lines and its financial transaction.

- If the system cannot generate the bill segment (for any number of reasons), the bill segment is moved to the Error state. You may regenerate the bill segment after correcting the source of the error (by clicking **Generate**). You may also delete such a bill segment.
- If the system successfully generates a bill segment, the bill segment becomes Freezable. If the bill segment is incorrect, correct the cause of the problem (e.g., fix the meter read, change the service agreement's rate) and regenerate the bill segment (by clicking **Generate**).

Click the delete button to physically remove an Incomplete, Error or Freezable bill segment from the database. Note, batch billing will delete unfrozen bill segments when it attempts to create bill segments for an account that has unfrozen bill segments. Also note, if you've plugged-in the appropriate pre-completion algorithm, the system will delete bill segments in error on the last night of the bill cycle. Refer to DEL-BSEG for more information.

Click **Freeze** to freeze a bill segment and its financial transaction. After doing this, the bill segment's state becomes Frozen and the bill segment may now appear on a customer's bill.

Copyright © 2026, Oracle and/or its affiliates.

Also note, when a bill segment is frozen, its financial details will be interfaced to the general ledger the next time the GL interface executes.

You may not change a bill segment once it is frozen. However, you have two options if you want to change a frozen bill segment's financial impact:

- If you need to remove the financial effects of a bill segment, click **Init Cancel**. Clicking this button causes a new financial transaction to be generated and linked to the bill segment. This financial transaction reverses the financial effects of the original bill segment. The bill segment becomes Pending Cancel after the system successfully generates this financial transaction. At this point, you have two options:
- Click **Cancel** to freeze the cancellation. Clicking this button causes the bill segment to move to the Canceled state. Once canceled, the bill segment cannot be uncanceled or changed.
- Click **Undo** to return the pending cancellation to the Frozen state.
- Click **Rebill** if you need to recalculate the bill segment because something has changed since it was originally created (e.g., the service agreement's rate was changed, a new read was entered). Clicking this button causes the following events to take place:
- The original bill segment is moved to the Pending Cancel state.
- A new bill segment is created in the Freezable (or Error) state.

At this point, you have two options.

- Click **Freeze** to make the new bill segment Frozen and make the original bill segment Canceled.

> **Note**
>
> **Multi-Cancel/Rebill Saves Time.** If you need to rebill many bill segments related to a service agreement (e.g., because many bill segments were created using the wrong rate), you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once. Also note that for service agreements that require bill determinants from Oracle Utilities Meter Data Management, this transaction may only be used to cancel / rebill a single bill segment at a time.
>
> Using the cancel and cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, then you will need to use Credit Notes. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

- Click **Undo** to delete the new bill segment and return the original bill segment to the Frozen state.

Bill Segment - Main Information

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To for a description of how to perform common bill maintenance functions.

The Main page contains core bill segment information. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** to view this information.

Note

Most bill segments are created by a background process. Refer to A High Level Overview Of The Bill Creation Process for information describing how the system creates a bill segment.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

Account ID references the bill segment's account.

Current Amount is the bill segment's effect on the service agreement's current balance. **Payoff Amount** is only shown if it differs from current amount. Refer to Billing - Current Balance versus Payoff Balance for more information.

Bill ID is the system-assigned unique identifier of the bill on which the bill segment appears. A concatenation of its bill date, status, due date and amount is displayed.

SA ID contains information about the bill segment's service agreement. A concatenation of the service agreement's division, SA type, start option (if any), status and start date is displayed.

If the quotation module is not turned off, you can generate simulated bill segments when you create a quote. You can view simulated bill segments by drilling down on the desired bill segment from Quote - Quote Details. When a simulated bill segment is displayed on this page, the **Billing Scenario ID** that controlled the bill segment's billing period (and consumption) is displayed. Refer to Creating Quotes And Quote Details for more information.

Period is the start and end dates of the bill segment.

Usage Period is the period used by Oracle Utilities Meter Data Management to determine the correct interval data to process for the bill segment. This field appears when the bill segment is for an SA that has Bill Determinants Required for the special role flag.

Bill Cycle displays the bill cycle and window start date of the bill segment's bill. This information is only populated on bills when they are generated on-schedule by the batch billing background process. Bills generated manually, i.e. off-schedule, do not have this information.

Status is the bill segment's status. Refer to Bill Segment Lifecycle for the potential values and how to handle a bill segment when it exists in a given state.

If the **Closing** switch is on, this is the final bill segment for the service agreement (i.e., the service agreement has been stopped and this is the final bill segment).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The closing switch is not set for sub SA bill segments nor billable charge SA bill segments.

If the **Estimate** switch is on, consumption on this bill segment has been estimated by the system.

If the **SQ Override** switch is on, a user has overridden the service quantities (i.e., the amount of consumption for which the customer is charged is not consistent with the customer's meter reads). This switch is also set if a user has overridden the calculation/audit read details.

If the **Item Override** switch is on, a user has overridden the type and/or number of items (i.e., the type/number of items for which the customer is charged is not consistent with the customer's service point information). Refer to How To Override The Items Billed On A Specific Bill Segment for more information.

If this bill segment was created for a sub service agreement where we calculate the charges (i.e., not for a "pass through" billing situation):

- **SA Rel ID** shows the unique identifier of the record that defines the relationship between the sub service agreement and the master service agreement (this record shows the service provider associated with the bill segment). Click the go to button to transfer to this record.
- **Master Bseg ID** shows the unique identifier of the master bill segment whose bill triggered the creation of this sub service agreement. This bill segment's consumption was used to calculate the charges on this bill segment.

Premise is the address of the main premise associated with the service agreement (as defined on the service agreement's characteristic premise). It's important to note that the service agreement may cover service at multiple premises, but only one may be defined as the main premise.

Create Dt/Tm is the date and time on which the bill segment was created or regenerated.

In the **Message** area, a brief error message appears if there's a problem with the bill segment. The message area is suppressed if there are no problems with the bill.

Push the magnifying glass button to view the long explanation. The long explanation will provide information about the cause of the error (and how to fix it).

If the bill segment is a rebill of a canceled bill segment, a reference to the original bill segment is displayed. Click the adjacent go to button to transfer to the bill segment that was canceled.

If the bill segment was canceled and rebilled by another segment, a reference to the new bill segment is displayed. Click the adjacent go to button to transfer to the new bill segment that superseded the canceled segment.

The topics that follow describe each of the actions on that appear in the **Bill Segment Actions** area. Refer to How To for a description of typical business processes that use these buttons.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Mass update actions.** This page allows you to work on individual bill segments. If you want to update many bill segments linked to a bill at once, try using the mass update actions available on Bill -Bill Segments.

- Generate (Bill Segment)
- Delete (Bill Segment)
- Freeze (Bill Segment)
- Rebill (Bill Segment)
- Init Cancel (Bill Segment)
- Undo (Bill Segment)
- Cancel (Bill Segment)
- Credit Note (Bill Segment)

Generate (Bill Segment)

Click the **Generate** button to create a new a bill segment.

This button is enabled when either of the following conditions is true:

- You are in add mode (i.e., you are not displaying an existing bill segment) and you have selected an **Account ID**, **Bill ID** and **SA ID**.
- You are displaying a bill segment that is Incomplete, Error, or Freezable. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Generate** window opens.

You must specify the following parameters to generate a new bill segment:

- **Cut-off Date** is the last possible day of the bill segment's bill period. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** is used to define the financial period to which the new bill segment's financial transaction will be booked. The current date defaults when the window opens.
- Turn on **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- If the service agreement uses meter reads, use the **Billing Option** to control how the system retrieves meter reads that appear on the bill. Choose Use Cut off Date if the meter reads must exist on the **Cutoff Date** (defined above). Choose Use Schedule if you want the service agreement's service point(s) measurement cycle schedule to control the meter reads that are used.

After specifying the above parameters, click the **Calculate** button to create a new bill segment for the bill.

If an error exists, refer to How To Correct A Bill Segment That's In Error.

Copyright © 2026, Oracle and/or its affiliates.

If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:

- If this option is set to Freeze At Bill Completion, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
- If this option is set to Freeze At Will, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

Delete (Bill Segment)

The **Delete** button deletes a bill segment.

This button is enabled if you are displaying a bill segment that is Incomplete, Error, or Freezable. Refer to Bill Segment Lifecycle for more information about these status values.

Freeze (Bill Segment)

Clicking the **Freeze** button causes a freezable bill segment to become frozen. Refer to Bill Segment Lifecycle for more information about freezing.

This button is enabled if:

- the Freeze At Will **Bill Segment Freeze Option** on the installation option has been selected AND
- the bill segment is freezable.

> **Note**
>
> **Freezing a day or more after generation.** If the bill segment is linked to a closed accounting period, a pop-up appears in which you can specify a new accounting date. This will only happen if the accounting period closes before you freeze the bill segment (the accounting date is stamped on a bill segment when it's initially generated).
>
> **If problems are detected after freezing.** A bill segment may not be changed after it is frozen. All subsequent changes must occur by canceling the frozen bill segment and creating a new bill segment. Refer to How To Cancel A Bill Segment and How To Cancel / Rebill A Bill Segment for more information.

Rebill (Bill Segment)

If problems are detected with a frozen bill segment, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you'll probably need to fix the cause of the problem. The following table describes common problems and where to go to fix them.

Copyright © 2026, Oracle and/or its affiliates.

Problem	How To Fix It
The wrong rate was used	Go to Service Agreement - Rate Info and correct the rate.
The wrong meter was connected to the service point	
A contract rider was applied incorrectly	Go to Service Agreement - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Service Agreement - Rate Info and correct the tax exemption information.

> **Note**
>
> **Multi-Cancel/Rebill Saves Time.** If the cause of the problem impacts many bill segments related to a service agreement, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once. Also note that for service agreements that require bill determinants from Oracle Utilities Meter Data Management, this transaction may only be used to cancel / rebill a single bill segment at a time.
>
> **Credit Notes**. Using the cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the **Rebill** functionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

The **Rebill** button causes an existing bill segment to be canceled, and a new bill segment to be created.

This button is enabled when you display a Frozen bill segment AND the bill segment's bill is not written off AND the bill segment's bill is not linked to an off cycle bill generator. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Rebill** window opens.

You must specify the following parameters to cancel and rebill a new bill segment:

- **Cancel Reason** defines why the bill segment is being canceled.
- **Cutoff Date** is the last day of the new bill segment. The cutoff date on the bill segment defaults when the window opens.
- **Accounting Date** is used to define the financial period to which the canceled and new bill segments' financial transactions will be booked. The current date defaults when the window opens.
- Turn on **Allow Estimates**to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- Use the **Billing Option** to define how the system looks for meter reads. Select Use Schedule if the system should select reads based on the SA's SP's measurement cycle schedule. Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
- Turn on **Use Old Read/Item/SQ for Regen** if you want the system to use the consumption details on the original bill segment when it calculates the new bill segment.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The **Use Old Read/Item/SQ** switch is not available if the bill's SA references an SA type with a special role flag of Bill Determinants Required.

Click the **Calculate** button to cancel the original bill segment and create a new bill segment.

At this point, there are now two bill segments - the original bill segment is in the state of Pending Cancel, the new bill segment is in the Freezable (or Error) state. If you want to back out, click the go to button to return to the original bill segment and then click Undo (this will delete the new bill segment and return the original bill segment to the Frozen state). If the new bill segment looks correct, click Freeze to move the original bill segment to the Canceled state and the new bill segment to the Frozen state.

Init Cancel (Bill Segment)

The **Init Cancel** button causes the first step of the bill segment cancellation process to be executed. You'd click this button if a frozen bill segment should never have been created (i.e., you want to remove the financial impact of a bill segment from a customer's balance).

This button is enabled when you display a Frozen bill segment AND the bill segment's bill is not written off. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Cancel** window opens.

You must specify the following parameters to cancel and rebill a new bill segment:

- **Cancel Reason** defines why the bill segment is being canceled.
- **Accounting Date** is used to define the financial period to which the canceled bill segment's financial transactions will be booked. The current date defaults when the window opens.

Click the **Calculate** button to cancel the original bill segment.

At this point, the bill segment is in the state of Pending Cancel. If you want to back out, click Undo; the bill segment will be returned to the Frozen state. Click Cancel to move the bill segment to the Canceled state.

> **Note**
>
> **Credit Notes**. Using the cancel functionality will cause the cancel details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the cancel functionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

Undo (Bill Segment)

The **Undo** button returns a Pending Cancel bill segment to the Frozen state. If a Pending Cancel bill segment was created during a Cancel / Rebill, the newly created Freezable bill segment is also deleted.

Copyright © 2026, Oracle and/or its affiliates.

This button is enabled when you display a Pending Cancel bill segment. Refer to Bill Segment Lifecycle for more information about this status.

Cancel (Bill Segment)

Clicking the **Cancel** button causes a Pending Cancel bill segment to become Canceled. Refer to Bill Segment Lifecycle for more information about cancellation.

This button is enabled if a Pending Cancel bill segment is displayed AND the bill segment's bill is not written off.

When clicked, the **Cancel** window opens.

You must specify the following parameters to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions will be booked. The current date defaults when the window opens.

After specifying the above parameters, click the **Calculate** button to cancel the frozen bill segments.

If the related bill is pending, the cancellation will cause the bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is complete and you do not Reopen the bill, the financial impact of the canceled bill segment will appear on the next bill created for the account (under Bill Corrections).

If the related bill is complete you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a frozen bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

Credit Note (Bill Segment)

The **Credit Note** button is used to cancel a bill segment and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

> **Note**
>
> **If you want to issue a credit note for an entire bill.** Rather than cancel a specific bill segment, functionality exists that allows you to create a credit note for ALL bill segments on a bill. To use this feature, display the bill in question and then click the **Credit Note** button on the bill page.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND

Copyright © 2026, Oracle and/or its affiliates.

- the bill segment is frozen AND
- the bill segment's bill is not written off.

When clicked, the **Credit Note** window opens.

You must specify the following parameters to cancel the frozen bill segment:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segment's financial transaction will be booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details will appear. The current date defaults when the window opens.

After specifying the above parameters, click the **Calculate** button to cancel the frozen bill segment and create a credit note.

It should be noted that if this operation is successful, the new "negative bill" will be complete (meaning that it will be ready for transmission to the customer and for interface to the general ledger).

Bill Segment - Read Details

The Read Details page contains information about the meter reads that were used to derive the bill segment's metered consumption. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Read Details** page to view this information.

> **Note**
>
> Refer to How The System Determines How Much Was Consumed for how the system assembles this information.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To for a description of how to perform common bill maintenance functions.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The meter read grid is a snapshot of the meter reads used to calculate the bill segment's consumption. One row exists for every register linked to every meter linked to every service point linked to the service agreement. This information is for audit purposes only and may not be modified. The following information is displayed in the grid:

- **Final Unit of Measure** is the unit of measure of the resultant billing quantity.
- **Final Time of Use** is the time-of-use code of the resultant billing quantity.
- **Final Service Quantity ID** is the service quantity identifier of the resultant billing quantity.

Copyright © 2026, Oracle and/or its affiliates.

- **Register Quantity** is the resultant billing quantity. See Measured Quantity below for more information.
- **Service Point ID** is the unique identifier of the service point at which the meter was located.
- **Meter Configuration ID** is the unique identifier of the meter's configuration.
- **Sequence** is the sequence number of the register on the meter.
- **Unit of Measure** is the register's unit of measure.
- **Time of Use** is the register's time-of-use code.
- **SQI** is reserved for future use.
- **Start Register Reading** is the start register reading. If the register is consumptive, (as opposed to subtractive) the word N/A - Consumptive is displayed, as the start read is not relevant. Click the Go to button to drill down into the meter read.
- **End Register Reading** is the end register reading. Click the Go to button to drill down into the meter read.
- **Measured Quantity** is the resulting measured quantity.

> **Note**
>
> **Measured quantity isn't necessarily the difference between the start and end read.** The measured quantity is equal to the Measured Quantity * Register Constant * Installation Constant * Use Percent.
>
> **Measured quantity isn't necessarily the quantity that's billed.** If the rate used to calculate the bill segment has a pre-processing calculation group, this consumption will be transformed further. For example, a cubic foot of gas can be transformed by a calculation rule to a therm of gas. When calculation rules convert measured consumption, they do NOT overwrite the measured quantity. Rather, they populate the Final UOM, Final TOU, Final SQI, and Register Quantity with the final billable quantity. Refer to All Calculation Rule-Based Rates Share a Common Structure for more information about pre-processing calculation groups.

- **Register Constant** is the constant on the register multiplied by the installation constant on SP Meter History. The resulting value is then used in calculating Measured Quantity.
- **How To Use** is the register's how to use code.
- **How To Use (SP)** is the SA/SP's how to use code.
- **Use Percent** is the SA/SP's usage percentage.
- **Measures Peak Qty** indicates if the unit of measure is one that measures peak consumption.
- **Start Read Date/Time** is the date and time of the start meter read. If the register is consumptive, (as opposed to subtractive) nothing will be displayed.
- **End Read Date/Time** is the date and time of the end meter read.
- **Start Reg Read ID** is the unique identifier of the start register read. If the register is consumptive, (as opposed to subtractive) nothing will be displayed.
- **Read Type** is the read type of the start meter read.
- **End Reg Read ID** is the unique identifier of the end register read.

Copyright © 2026, Oracle and/or its affiliates.

- **Read Type** is the read type of the end meter read.
- **Multiplier** is the composite multiplier that converts the measured quantity to the registered quantity.

> **Note**
>
> The multiplier is calculated by dividing the Register Quantity by the Measured Quantity (if the Measured Quantity is 0, then the result is 0). This is not stored on the database. There is a chance that the value is not precise due to round-off.

Bill Segment - Item Details

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Override The Items Billed On A Specific Bill Segment for a description of how to override this information.

The Item Details page contains information about the items located at the service agreement's service points. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Item Details** page to view this information.

> **Note**
>
> Refer to How the System Requests How Much Was Consumed for how the system assembles this information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The item details grid is a snapshot of the items that existed at the service agreement's service points at the time the bill segment was created. One row exists for every type of item linked to every service point linked to the service agreement.

This information may be modified on a bill segment in the Error or Freezable state. Insert one row in the item grid (click the + button) for each item you need to add. You may also change item amounts that were populated by the system when it initially generated the bill by simply overwriting the information.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> The proper way to fix the number and type of items on a bill segment is to correct the service point / item information and then regenerate the bill segment. Overriding the item details on a bill segment is a last resort that should only be used when you cannot fix the service point / item information (for whatever reason).

The following information is displayed in the grid:

- **Seq No** is the system-assigned unique identifier of the item detail row.
- **Item Type** is the type of item.
- **Item ID** is the unique identifier of the item (this is only displayed if a badged item is linked to the service point; non-badged items don't have unique identities).
- **Start Date** is the first day in the bill segment's bill period that the item was linked to the service point.
- **End Date** is the last day in the bill segment's bill period that the item was linked to the service point.
- **Item Count** is the number of items of this type that were linked to the service agreement's service points.
- If the item's item type indicates the system must estimate consumption (e.g., lamps), **UOM** is the unit of measure of the estimated consumption. **Daily Service Quantity** is the item's estimated amount of daily consumption.

> **NOT_SUPPORTED**
>
> Refer to Defining Device Options for more information about item types and defining the estimated consumption amount for items.

Bill Segment - SQ Details

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page.

The SQ Details page contains information about the How the System Requests How Much Was Consumed that will be priced by the service agreement's rate. In addition to showing SQ details, this page shows any Calculation / Audit Read Details and Calculation / Audit Usage Details that are linked to the bill segment. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **SQ Details** page to view this information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

Copyright © 2026, Oracle and/or its affiliates.

The **Service Quantity** details grid is a snapshot of the service quantities amassed from:

- The bill segment's usage request.
- Those items for which the system estimates consumption. Refer to Defining Device Options for more information.
- Billable charges that supply service quantities.

One row exists for every unique combination of unit of measure (UOM), time-of-use (TOU) code, and service quantity identifier (SQI) associated with the service agreement.

This information may be overridden on a bill segment in the Error or Freezable state. Insert one row in the service quantity grid (click the + button) for each UOM/TOU/SQI you need to add. You may also change amounts that were populated by the system when it initially generated the bill by simply overwriting the information.

> **Warning**
>
> The proper way to fix the SQ details on a bill segment is to correct the cause of the error (e.g., correct the meter reading). Overriding the SQ details on a bill segment is a last resort that should only be used when you can't fix the cause of the problem. Refer to How To Cancel / Rebill A Bill Segment for ways to fix common problems.

The following information is displayed in the grid:

- **Unit of Measure** is the unit of measure of the service quantity.
- **Time of Use** is the time-of-use of the service quantity.
- **SQI** is the service quantity identifier of the service quantity.
- **Initial Service Quantity** is the initial quantity amassed by the system before application of the rate's pre-processing calculation group(s), if any exists.
- **Billable Service Quantity** is the service quantity that will be priced by the service agreement's rate. This amount differs from the initial service quantity when the rate's pre-processing calculation rule(s) manipulate consumption.

The **Calculation/Audit Read Details** section contains Calculation / Audit Read Details. This section only appears if any of the following conditions are true:

- The service agreement has a special role of Billable Charge or Bill Determinants Required.
- The bill segment has one or more reads with a usage flag of Calculation/Audit .

The following information may be defined for each calculation / audit read:

- **Final Unit of Measure** is the final unit of measure of the calculation / audit read.
- **Final Time of Use** is the final time-of-use code of the calculation / audit read.
- **Final SQI** is the final service quantity identifier of the calculation / audit read.
- **Start Date/Time** is the date and time of the calculation / audit read.
- **End Date/Time** is the date and time of the calculation / audit read.
- **Quantity** is the quantity of the calculation / audit read. This would typically contain the amount that was billed during the **Start Date/Time** through the **End Date/Time** for the **Final Unit of Measure**, **Final Time of Use** and **Final SQI**.
- **Unit of Measure** is the initial unit of measure of the calculation / audit read (if any).

Copyright © 2026, Oracle and/or its affiliates.

- **Time of Use** is the initial time-of-use code of the calculation / audit read.
- **SQI** is the initial service quantity identifier of the calculation / audit read.
- **Start Quantity** is the "start" quantity (if any). This value would typically only be displayed if the calculation / audit read is subtractive (i.e., you must subtract the end quantity from the start quantity to determine billable consumption).
- **End Quantity** is the "end" quantity (if any). This value would typically only be displayed if the calculation / audit read is subtractive (i.e., you must subtract the end quantity from the start quantity to determine billable consumption).
- **Measured Quantity** is the resulting measured quantity (if any). This value would typically only be displayed if the calculation / audit read is subtractive (i.e., you must subtract the end quantity from the start quantity to determine billable consumption). It should contain the difference between the **Start Quantity** and the **End Quantity**.
- **Constant** is the constant of the measuring device that was multiplied by the **Measured Quantity** to derive the calculation / audit read's **Quantity**.
- **How To Use** is a code that indicates if the "calculation / audit" read was is considered to be additive, subtractive, peak or check.
- **Use Percent** is the percentage of the total consumption that was billed.
- **Meas Peak Qty** indicates if the unit of measure is one that measures peak consumption.
- **Multiplier** is the ratio of the **Quantity** and the **Measured Quantity**. This value only appears if a non-zero value exists in both of these fields.
- **Seq No** is the sequence number (if any).
- **Service Point** is the unique identifier of the service point at which the readings were calculated (if applicable).

This information may be overridden on a bill segment in the Error or Freezable state. Insert one row in the calculation / audit details grid (click the + button) for each entry you need to add. You may also information populated by the system when it initially generated the bill by simply overwriting the information.

> **Warning**
>
> The proper way to fix the calculation/audit details on a bill segment is to correct the cause of the error. Overriding the calculation/audit details on a bill segment is a last resort that should only be used when you can't fix the cause of the problem.

> **NOT_SUPPORTED**
>
> Refer to Calculation / Audit Read Details for more information about this section.

The **Calculation/Audit Usage Details** grid contains usage service quantities for the bill segment. This section appears when the service agreement has a special role of Bill Determinants Required. The following information is displayed in the grid:

- **Final Unit of Measure** is the final unit of measure of the calculation / audit usage.
- **Final Time of Use** is the final time-of-use code of the calculation / audit usage.

Copyright © 2026, Oracle and/or its affiliates.

- **Final Service Quantity ID** is the final service quantity identifier of the calculation / audit usage.
- **Start Date/Time** is the date and time of the calculation / audit usage.
- **End Date/Time** is the date and time of the calculation / audit usage.
- **Quantity** is the quantity of the calculation / audit usage. This would typically contain the amount that was billed during the **Start Date/Time** through the **End Date/Time** for the **Final Unit of Measure**, **Final Time of Use** and **Final SQI**.
- **Sequence** is the sequence number.
- **Service Point ID** is the unique identifier of the service point at which the usage was calculated.

NOT_SUPPORTED

Refer to Calculation / Audit Usage Details for more information about this section.

Bill Segment - Calc Lines

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To Cancel / Rebill A Bill Segment for instructions describing how to regenerate this information if the bill segment is incorrect.

The Calc Lines page contains information about the bill calculation lines produced when the system generates the bill segment. These lines are the source of the details printed on the customer's bill. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Calc Lines** page to view this information.

NOT_SUPPORTED

Most service agreements use a rate to calculate the information that appears on this page. Refer to How Rates Affect The Information On Bill Segments for more information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The Bill Segment Header (**Bill Seg Hdr)** scroll contains one entry for every version of the rate that was used to calculate the bill segment's lines. Because most bill segments use a single rate version calculation group, you typically only see one entry in the scroll.

The following information is displayed above the bill calculation line grid.

Copyright © 2026, Oracle and/or its affiliates.

- **Sequence** will be 1 unless multiple versions of the rate were in effect during the bill period. If there was more than one effective rate version calculation group, there will be a separate set of calc lines for each version applied. The lines that apply to the first part of the bill period would have a Sequence of 1, the lines for the next part of the bill period would have a Sequence of 2 and so on.
- **Start Date - End Date** is the portion of the bill segment period that the calculation details apply. .
- **Amount** is the sum of the bill calculation lines in the grid.
- **Desc on Bill** is summary information about the calculation details that will be printed on the customer's bill.
- **Rate Version** is the ID of the rate version calculation group and its effective date. This information only appears if a rate was used to calculate the charges.

The calculation lines grid is a snapshot of how the system calculated the bill segment amount. One row exists for every calculation involved in this process. This information is for audit purposes only and may not be modified. The following information is displayed in the grid:

- A drill down button appears in the **Calc Line Char** column if the bill calculation line has characteristics. The **Calc Line Char** column only appears if at least one of the calculation lines has a characteristic.
- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the bill line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the bill line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **Appears in Summary** switch defines if this line's amount also appears on a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column. The system automatically populates this switch as follows:
  - Bill segments created by applying a rate have this switch turned on if the corresponding calculation rule is summarized on a summary calculation rule
  - Bill segments created from billable charges use the value of this switch as defined on the billable charge. Refer to Maintaining Billable Charges for more information.
- **Unit of Measure** (UOM) is the unit of measure of the service quantity priced on the calculation line.
- **Time of Use** (TOU) is the time-of-use code of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity will be different from the measured consumption is there are pre-processing calculation group in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.
- **Calculation Rule** is the rule that calculated the line. This is only populated if the new rate engine calculated the charges.

Copyright © 2026, Oracle and/or its affiliates.

- **Measures Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt.
- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the line was calculated using a bill factor (because only bill factors use characteristic values). Refer to An Illustration Of A Bill Factor And Its Characteristics for more information.

Bill Segment - Financial Details

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To Cancel / Rebill A Bill Segment for instructions describing how to regenerate this information if it is incorrect.

The Financial Details page contains information about the financial effects of the bill segment. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Financial Details** page to view this information.

Note

Refer to The Source Of GL Accounts On A Bill Segment's Financial Transaction for how the system compiles this information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The **Bill Segment FT** scroll contains one entry for every financial transaction (FT) associated with the bill segment. All bill segments have an FT that shows the impact of the bill segment on the general ledger. A bill segment may have an additional FT (for a maximum of two) if it is cancelled. The second FT shows the impact of the cancellation on the general ledger.

The following financial transaction information is displayed above the general ledger distribution grid.

- **FT Type** displays the type of financial transaction (FT).
- **Show on bill** indicates if information about the FT appears on the customer's bill.
- **FT ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the Financial Transaction - Main page. On this page you can change certain aspects of

Copyright © 2026, Oracle and/or its affiliates.

the FT in question. Refer to How To Remove Unwanted Adjustments (or Payments) From A Completed Bill for a description of how to perform common maintenance functions.

- The posted by information that appears adjacent to the FT ID shows the date on which the FT was frozen and the user ID of the user who performed the operation.
- The "Linked to bill on" message appears when the financial transaction is linked to a bill. A FT is linked to a bill the next time a bill is completed for the account. Therefore, if a user cancels a bill segment, the FT associated with the cancellation will not be linked to a bill until the next bill is completed (although the financial effect of the cancellation takes place immediately). Click the adjacent go to button to transfer to the bill on which the FT appears.
- **Current Amount** contains the FT's effect on the service agreement's current balance.
- **Payoff Amount** contains the FT's effect on the service agreement's payoff balance. This field is only visible when the payoff amount differs from the current amount.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Arrears Date** is the date the FT starts aging.

The journal details grid is a snapshot of the distribution codes used to generate the GL accounts that are affected by the FT. This information is for audit purposes only and may not be modified. The following information is displayed in the grid:

- **Seq** is the system-assigned unique identifier of the distribution line.
- **Distribution Code** is the distribution code used to generate the GL account affected by the distribution line. Drill over to the FT if you need to see the actual GL account.
- **Amount** is the amount debited / credited (credits are shown as negative values).
- **Statistics** and **Statistics Amount** are only specified when a statistical quantity is associated with the distribution line. For example, if the line is associated with a charge for the total number of kilowatt-hours consumed, the statistics code will be the kWh and the statistics amount will be the number of kilowatt-hours associated with the charge.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the distribution line was derived from bill calculation lines that were calculated using a bill factor (because only bill factor's use characteristic values). Refer to An Illustration Of A Bill Factor And Its Characteristics for more information.
- If you have configured your installation options to indicate that fund accounting is practiced, the description of the **Fund** associated with this distribution code is displayed.

Bill Segment - Bill Segment Messages

The Messages page is a grid containing one row for every message that appears on the bill segment. Open **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Bill Segment Messages** page to view this information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **No messages until completion.** A bill segment has no messages until it is completed (unless you insert them manually). At completion time, the system assembles messages from the various message sources. Refer to The Source Of Bill Messages for information about these sources.
>
> **The bill may also have messages.** Be aware that only service agreement-oriented messages are linked to a bill segment. There may also be account-oriented messages linked to the bill. Refer to Bill - Bill Messages for information about the page on which account-oriented messages are displayed. Refer to The Source Of Bill Messages for information about the various message sources and whether each is linked to a bill or a bill segment.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The following columns are displayed in the grid:

- **Msg Code** is the code that identifies the bill message.
- **Message On Bill** is the message associated with the code.
- **Priority** is the bill messages priority (on the printed bill).
- **Insert Code** defines if the bill message causes an insert in the envelope.

Bill Segment - Sub Bill Segments

The Sub Bill Segments page contains the sub bill segments that are associated with the master SA's bill segment. Open this page using **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Sub Bill Segment** page.

> **Note**
>
> Sub bill segments are only linked to the master SA's bill segment when the master SA has an SA relationship with a We Bill For Them - Rate Ready or an It's Us service provider. Refer to The Rate Ready Calculation Method for the details.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, SA type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The following information is displayed in the grid.

- Click the adjacent **Go To** button to transfer to the bill segment.
- The **Bill Segment** column contains a concatenation of the bill segment's business unit, SA type, status, bill period and amount. The business unit and SA type come from the bill segment's service agreement.

Copyright © 2026, Oracle and/or its affiliates.

- The **Current Amount** column contains the bill segment's effect on the sub service agreement's current balance.
- The **Status** column contains the bill segment's current status.

Bill Segment - Bill Segment Portal

Select **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Bill Segment Portal** page to view additional information associated with a Bill Segment.

Depending on your implementations requirements, you may define additional zones to appear on this Portal

The contents of this section describe the zones that are available on this portal page.

General Information

Bill Segment ID, **Bill Seg Info** are displayed.

Bill Segment - Usage

This Usage portal is available when the bill segment has associated usage requests. It displays all usage requests related to the bill segment. Select **Menu**, and then **Financial**, and then **Bill Segment**, and then **Search** and navigate to the **Usage** page to view bill segment's usages.

General Information

Bill Segment ID and Bill Segment Info are displayed.

Bill Segment Usage

The following information is displayed in the Bill Segment Usage zone:

- The **Create Date/Time** of the usage request.
- The **Usage** column contains a concatenation of the bill segment's division, SA type, status, and period of the usage request. The usage information links to the Usage Portal for the usage request.
- The **Usage ID** for the usage request.

Financial - Bill Segment Calculation Line Characteristics

This page displays the characteristics that are linked to a specific bill segment calculation line. The information on this page is for audit purposes only and cannot be changed.

Open this page by clicking the **Go To Calc Line Characteristic** button on the calculation lines grid of the Bill Segment - Calc Lines page.

Description of Page

Account is the account to which the bill segment is linked. The name of the main person on the account appears next to the account ID.

The **SA** is the service agreement to which the bill segment applies. Basic information about the service agreement appears adjacent.

Copyright © 2026, Oracle and/or its affiliates.

Premise shows the service agreement's characteristic premise (i.e., the primary premise associated with the service agreement).

The **Bill Seg ID** and **Bill Seg Header** displays the identification of the bill segment and header associated with the calculation line characteristic.

Calc Line displays the calculation line number with which the characteristics are associated.

The **Description on Bill** displays the description of the calculation line that appears on the customer's bill.

The characteristics grid displays the **Characteristic Type**s and the **Characteristic Value**s associated with the calculation line.

> **NOT_SUPPORTED**
>
> For more information about characteristics, refer to Setting Up Characteristic Types and Their Values.

Off Cycle Bill Generator

The Off Cycle Bill Generator portal may be used to create an off cycle bill for an account where you control which financial transactions are linked to the resultant bill. For example, you can create a bill for a specific adjustment or billable charge. Navigate using **Menu**, and then **Financial**, and then **Off Cycle Bill Generator**, and then **Add.**

- Off Cycle Bill Generator Query
- Off Cycle Bill Generator Portal

Off Cycle Bill Generator Query

Use the query portal to search for an existing off cycle bill generator. Once an off cycle bill generator is selected, you are navigated to the maintenance portal to view and maintain the selected record.

Off Cycle Bill Generator Portal

This portal appears when an off cycle bill generator has been selected from the Off Cycle Bill Generator Query portal.

The topics in this section describe the base-package zones that appear on this portal.

- Off Cycle Bill Generator Actions
- Off Cycle Bill Generator Zone
- Financial Transactions On Bill
- Off Cycle Bill Generator Log

Copyright © 2026, Oracle and/or its affiliates.

Off Cycle Bill Generator Actions

This is a standard actions zone.

If the off cycle bill generator is in a state that has valid next states, buttons to transition to each appropriate next state are displayed.

> **NOT_SUPPORTED**
>
> Refer to How To Create An Off Cycle Bill Generator for more information about how to create an off cycle bill on-line / real time.

Off Cycle Bill Generator Zone

The Off Cycle Bill Generator zone contains display-only information about the off cycle bill generator.

Please see the zone's help text for information about this zone's fields.

Financial Transactions On Bill

The Financial Transactions On Bill zone shows financial transactions linked to the off cycle bill generator's bill. Information displayed on this zone is similar to that displayed on the Financial Transactions On A Bill transaction.

Off Cycle Bill Generator Log

An off cycle bill generator log contains an entry for every recorded event during the lifecycle of the off cycle bill generator. There are two general types of log entries:

- **Automatic entries.** The system automatically creates an entry in the log when an off cycle bill generator is created or there is a status change or when a related entity is created. This also includes any implementation-specific log entries. Users cannot modify or delete these log entries.
- **Manual entries.** Users can add manual entries to record significant events at their discretion.

Multi Cancel/Rebill

The Multi-Cancel/Rebill transaction is used to cancel / rebill (and freeze) one or more bill segments linked to a service agreement.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You would typically use this transaction when something has been wrong for an extended period of time. For example, if the customer was assigned the wrong rate when they first started service, you would correct the customer's rate using Service Agreement - Rate Info and then use this transaction to cancel / rebill the customer's historical bill segments.

The following topics describe how to use this transaction.

- Multi Cancel/Rebill - Main
- Multi Cancel/Rebill - Graph

Multi Cancel/Rebill - Main

The main page is used to select the bill segments to be cancel/rebilled and to initiate the cancel/rebill request. Open **Menu**, and then **Financial**, and then **Multi-Cancel/Rebill** to use this function.

Description of Page

This page contains all non-cancelled bill segments associated with a given **SA ID**. The service agreement's **SA Info**, **Account ID** (and name), characteristic **Premise** (and address), **Current Balance** and **Payoff Balance** are displayed at the top of page.

When the page first opens, the grid is populated with ALL non-cancelled bill segments displayed in reverse chronology order. All frozen bill segments are pre-selected for cancel/rebill processing. The following information appears in the grid:

- Selected Switch. If this switch is turned on, the bill segment will be cancel / rebilled when the **Cancel/Rebill/Freeze** button is clicked. You can use the mouse to toggle the value of this switch on a specific bill segment. Alternatively, you can click the **Select All** button or **Clear All** button to toggle the switch for all frozen bill segments. The checkbox is disabled if the bill segment's bill is written off. Note: the total number of selected bill segments is shown above the grid for confirmation purposes.
- **Start Date.** This is the first day of the bill segment.
- **End Date.** This is the last day of the bill segment.
- **Create Date/Time.** This is the day/time the bill segment was created.
- **Amount.** This is the payoff amount of the bill segment.
- **Original Amount.** This column only contains a value if the bill segment is a rebill of another bill segment. In this situation, one of the following values will be displayed:
- If the amount of the new bill segment differs from the amount of the original bill segment, the amount of the original bill segment will be displayed.
- If the amount of the new bill segment equals the amount of the original bill segment, the word No change will be displayed.
- **Bill Segment Information.** This is the standard bill segment summary information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Sorting By Values In The Grid.** Just like all grids in the system, you can click on a column heading to resort the grid by the value of the respective column. This might be useful to sort the bill segments in **Amount** order.

After selecting the bill segments to be cancel/rebilled, click the **Cancel/Rebill/Freeze** button. When this button is clicked, an additional informational page displays. Specify the following information and then click **Calculate** to cancel the selected bill segment and create new bill segments:

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** is used to define the financial period to which the canceled and new bill segments' financial transactions will be booked.
- Turn on **Allow Estimates** to indicate if Oracle Utilities Meter Data Management should estimate consumption if it cannot find a meter read.
- Use the **Billing Option** to define how the system looks for meter reads:
  - Select Use Schedule if the system should select reads based on the SA's SP's measurement cycle schedule.
  - Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
- Select Use Schedule if the system should select reads based on the SA's SP's measurement cycle schedule.
- Select Use Cut off Date if the system should only select reads from the end date of the existing bill segment.
- If you'd prefer to have the system use the consumption details snapshot on the original bill segments, turn on **Use Old Read/Item/SQ**. If this switch is on then neither **Allow Estimates** nor **Billing Option** is relevant.

> **Note**
>
> The **Use Old Read/Item/SQ** switch is not available if the bill's SA references an SA type with a special role flag of Bill Determinants Required.

> **Warning**
>
> If the cause of the rebill is an incorrect rate and the new rate requires service quantities that were not calculated when the bill was originally calculated, do not turn on **Use Old Read/Item/SQ**. Why? Because you want the system to derive new service quantities during the rebill process.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Default note.** The Accounting Date is defaulted to the current date and the **Use Old Read/Item/SQ** switch is turned on.

After specifying the above information, the system cancel / rebills each segment in chronological order (i.e., earlier segments will be cancel / rebilled before later segments). The new amount of each bill segment is shown in the **Original Amount** column. You can navigate to the Graph page to graphically view the financial ramifications of the cancel/rebills.

> **Note**
>
> **Errors.** It's possible that bill segment errors will occur when you cancel / rebill the bill segments. For example, if you change the SA's rate to a rate that requires a unit-of-measure that doesn't exist on the SA's meter, a bill segment error will be generated. If this occurs, you must go to Bill Segment - Main for the pending cancel bill segment and use the **Undo** action to remove the offending bill segment (and restore the original segment to the Frozen state). Alternatively, you can drill down on the Error bill segment and use the **Regenerate** action after correcting the cause of the problem.
>
> **There is no Undo.** Unlike cancel / rebills performed on Bill Segment - Main, there is no Undo action. This is because the multi-cancel/rebill transaction freezes the newly created bill segments after it cancel/rebills; whereas the bill segment transaction lets you examine the new bill segment before you freeze it. If you made a mistake, simply correct the cause of the mistake and then cancel / rebill again. The erroneous financial transactions will be automatically suppressed on the next bill produced for the customer.

Multi Cancel/Rebill - Graph

This page contains a graph that shows the rebilled and original payoff amounts for a service agreement's non-cancelled bill segments. Open **Menu**, and then **Financial**, and then **Multi Cancel/Rebill** and navigate to the **Graph** page to view this information.

Description of Page

This page contains a graph that shows the payoff amounts for a service agreement's non-cancelled bill segments (represented by the blue bars in the graph). If a bill segment is a rebill of another bill segment, the payoff amount of the original bill segment is displayed on a red-colored bar adjacent to the respective blue bar.

> **Note**
>
> **Click a bar.** You can click on a bar to drill down to a bill segment.

Copyright © 2026, Oracle and/or its affiliates.

SA Billing History

Open **Menu**, and then **Financial Query**, and then **SA Billing History** to view a history of all bill segments produced for a service agreement.

Description of Page

The **Account ID** is displayed with the account's name adjacent.

The **SA Information** shows a summary of information about the service agreement. The **SA ID** is the unique identifier of this SA.

One row is displayed for every bill segment ever produced for the selected service agreement.

The following information is displayed for each bill segment:

- The **Start Date** and **End Date** of the bill segment.
- The number of **Days** in the bill segment consumption period.
- The **Status** of the bill segment
- The **Current** and **Payoff Amount**s of the bill segment.
- The **UOM** (Unit of Measure) of the service quantity designated as the "graph UOM" for the SA's SA type.
- The total amount of the service **Billable Service Quantity** that was billed.
- The **Avg Daily Qty** is the **Billable Service Quantity** divided by the numbers of days of service.

If you need to see more detailed information about the bill segment, click the go to button to transfer to the bill segment page.

Bill Exception

If errors are detected during the billing process, it may cause a record to be written to the bill exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BIERR background process. This process generates a To Do entry for every record in the bill exception table.

NOT_SUPPORTED

Refer to How to Correct a Bill That's In Error for instructions describing how to correct a bill.

Bill Segment Exception

Copyright © 2026, Oracle and/or its affiliates.

If errors are detected during the billing process, it may cause a record to be written to the bill segment exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BSERR background process. This process generates a To Do entry for every record in the bill segment exception table.

NOT_SUPPORTED

Refer to How To Correct A Bill Segment That's In Error for instructions describing how to correct a bill segment.

Maintaining Billable Charges

You create a billable charge whenever a customer should be charged for a service that occurs outside the normal course of business. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line.

Note

Billable charges can be uploaded from an external system. Refer to Uploading Billable Charges and Using Billable Charges For Pass Through Billing for more information.

A billable charge must refer to a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each billable charge. This means that if a service agreement has many unbilled billable charges, it will have many bill segments.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.
- This service agreement must reference an SA Type that has a Special Role of Billable Charge. If such a service agreement does not already exist for the account, one must be created before you can levy a billable charge. See "For more information" at the end of this section for links to other sections that will help you with this process.

Billable charge templates exist to minimize the effort required to create a billable charge for a customer. A billable charge template contains the standard bill lines, amounts and distribution codes used to charge for a one-off charge.

A user may override the information on the template when the billable charge is created. For example, you can create a billable charge template to levy charges when a contractor ruptures a gas main. This template would contain the bill lines, amounts and distribution codes associated with a ruptured pipe repair. Then, when a contractor ruptures a line, a user can

Copyright © 2026, Oracle and/or its affiliates.

create a billable charge using the template and override the amount to reflect the actual repair amount (if it differs from the norm).

> **Note**
>
> **Templates aren't required**. A billable charge can be created without a template for a truly unexpected charge. If you don't use a template, you'll have to enter the bill information manually.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Billable Charge Templates for more information.

> **Note**
>
> **Adding taxes and other calculation lines to a billable charge.** Refer to Billable Charge - SQ Details for information about how to have the system calculate tax lines and append them to the billable charge.

- Billable Charge - Main
- Billable Charge - Line Characteristics
- Billable Charge - SQ Details
- Billable Charge - Read Details

Billable Charge - Main

Open **Menu**, and then **Financial**, and then **Billable Charge**, and then **Search** to maintain a billable charge.

Description of Page

The **Billable Charge ID** is system generated. The **Account ID** of the account responsible for the charge is displayed at the top.

Use **SA ID** to define the ID of the billable charge service agreement that will hold the billable charge's debt. The **Account ID** for the service agreement is displayed for information purposes.

Use **Start Date** and **End Date** to define the charge's bill period.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Installment billable charges.** You can create many future dated billable charges to implement unusual installment plans. For example, if you want to bill a customer $250 today, $150 next month, and $1000 three months from now; you could create three billable charges with a date that reflects the earliest bill date. When a bill is produced for the account, the system will create a bill segment for every billable charge whose start date is on or before the current date.

If you want to create the billable charges from an existing template, enter the ID of the **Billable Charge Template** from which the billable charge information defaults. If your SA Type indicates a billable charge template, that value will be defaulted. The billable charge template is stored with the billable charge, to reference what was used to create it.

> **Note**
>
> Actual billable charge lines can differ from what was originally defaulted from the billable charge template if user overrides were done after defaulting. Refer to Maintaining Billable Charges for more details.

Use **Description on Bill** to define the verbiage that should print on the customer's bill above the line item details.

Billable Charge Status defines the state of the billable charges. Possible values are Billable and Cancelled. Click the **Cancel** button to cancel a billable charge.

Bill Segment ID is the ID of the bill segment on which the billable charge's charge details appear. This information is only populated AFTER the billable charge has been swept onto a bill.

The **Total Bill Amount** displays the sum of all billable charge lines except those that are **Memo Only**. The **Total Line Amount** displays the sum of all billable charge lines except those that **Appear In Summary**.

The information in the scroll defines the line item details associated with the billable charge. Click the drill down button to navigate to the line characteristics. The following fields are required for each line:

Sequence Sequence controls the order in which the line items appear on the bill segment.

Description on Bill Specify the verbiage to print on the bill for the line item.

Charge Amount Specify the amount to charge for the line item.

Show on Bill Turn this switch on if the line item should appear on the customer's printed bill. It would be very unusual for this switch to be off.

Appears in Summary Turn this switch on when the amount associated with this line also appears in a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column. Also note, those lines that appear in a summary line are not included in the total amount associated with the billable charge.

Memo Only, No GL Turn this switch on if the billable charge line does not contribute to the total amount due (e.g., this switch would be on for "information only" lines).

Copyright © 2026, Oracle and/or its affiliates.

Distribution Code Specify the distribution code that identifies the GL account associated with this line item. The description of the code appears adjacent.

NOT_SUPPORTED

For more information about billing billable charges, refer to How To Create A One-Time Invoice. For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Billable Charge - Line Characteristics

Characteristics may be associated with billable charge lines. Typically, this is done to categorize billable charge lines for reporting purposes.

Open **Menu**, and then **Financial**, and then **Billable Charge**, and then **Search** and navigate to the **Line Characteristics** page to maintain characteristics that are associated with billable charge.

Description of Page

The **Billable Charge Line** scroll defines the billable charge line to which you wish to assign characteristic values. The following fields display:

Characteristic Type The type of characteristic.

Characteristics Value The value of the characteristic.

Note

You can only choose characteristic types defined as permissible on a billable charge line. Refer to Setting Up Characteristic Types & Their Values for more information.

Billable Charge - SQ Details

Optionally, you may specify service quantities associated with a billable charge. These are most commonly used when you want the system to calculate charges *in addition to the billable charge lines* (e.g., adding tax to the billable charge lines).

Note

The SA must specify a rate. If you want the system to calculate additional charges based on the service quantities defined on this page, the related billable charge service agreement must reference a rate. This rate must have calculation rules that calculate charges based on the service quantities specified on this page. When the service agreement is next billed, the system will sweep on the billable charges and then call the rate and pass it the service quantities defined on this page. The rate will calculate the additional charges and these will be appended to the bill segment.

Copyright © 2026, Oracle and/or its affiliates.

Open **Menu**, and then **Financial**, and then **Billable Charge**, and then **Search** and navigate to the **SQ Details** page to maintain a billable charges service quantities.

Description of Page

The following fields display:

Sequence Specify the sequence number of the SQ.

UOM Select the unit of measure of this SQ. One or more of **UOM**, **TOU**, or **SQ Identifier** must be selected.

TOU Select the time of use period.

SQ Identifier Select the SQ identifier.

Service Quantity Specify the number of units of this service quantity.

Billable Charge - Read Details

Optionally, you may specify meter read details associated with a billable charge. These are most commonly used when you need to print the read details that were used to calculate the bill lines that appear on the main tab.

> **Note**
>
> **Using these reads to derive billable consumption.** The base package bill segment creation algorithm (BSBC-BC) simply sweeps billable charge read details onto the related bill segment as calculation / audit reads. This algorithm does NOT use these reads to derive billable consumption. This is intentional as most implementations that import reads also import billable consumption and therefore deriving billable consumption is unnecessary (and also error prone). If your implementation needs to derive billable consumption from these reads, you can develop your own bill segment creation algorithm to do just this.

Open **Menu**, and then **Financial**, and then **Billable Charge**, and then **Search** and navigate to the **Read Details** page to maintain a billable charges service quantities.

Description of Page

The grid is a snapshot of the meter reads whose consumption was billed using the bill lines shown on the main tab. One row exists for every read that should be swept onto the related bill segment as a calculation / audit read when the system creates a bill segment for this billable charge. The following information is displayed in the grid:

- **Final Unit of Measure** is the unit of measure of the resultant billing quantity.
- **Final Time of Use** is the time-of-use code of the resultant billing quantity.
- **Final Service Quantity Identifier** is the service quantity identifier of the resultant billing quantity.
- **Register Quantity** is the resultant billing quantity.
- **How To Use (SP)** is the SA/SP's how to use code.

Copyright © 2026, Oracle and/or its affiliates.

- **Service Point ID** is the unique identifier of the service point at which the meter was located.
- **Meter Configuration ID** is the unique identifier of the meter's configuration.
- **Sequence** is the sequence number of the register on the meter.
- **Unit of Measure** is the register's unit of measure.
- **Time of Use** is the register's time-of-use code.
- **SQI** is reserved for future use.
- **Start Register Reading** is the start register reading.
- **End Register Reading** is the end register reading.
- **Measured Quantity** is the resulting measured quantity.
- **Register Constant** is the constant on the register that is multiplied by the measured quantity to derive the register's billable consumption.
- **How To Use** is the register's how to use code.
- **Use Percent** is the SA/SP's usage percentage. A value of 100% is defaulted.
- **Measures Peak Qty** indicates if the unit of measure is one that measures peak consumption.
- **Start Read Date/Time** is the date and time of the start meter read. If the register is consumptive, (as opposed to subtractive) nothing will be displayed.
- **End Read Date/Time** is the date and time of the end meter read.
- **Start Reg Read ID** is the unique identifier of the start register read. If the register is consumptive, (as opposed to subtractive) nothing will be displayed.
- **Read Type** is the read type of the start meter read.
- **End Reg Read ID** is the unique identifier of the end register read.
- **Read Type** is the read type of the end meter read.
- **Multiplier** is the composite multiplier that converts the measured quantity to the registered quantity.

> **Note**
>
> The multiplier is calculated by dividing the Register Quantity by the Measured Quantity (if the Measured Quantity is 0, then the result is 0). This is not stored on the database. There is a chance that the value is not precise due to round-off.

Uploading Billable Charges

This section describes how the system uploads billable charges from an external source. This mechanism is used when you need to "pass through" charges that have been calculated by a third party but appear on your bill.

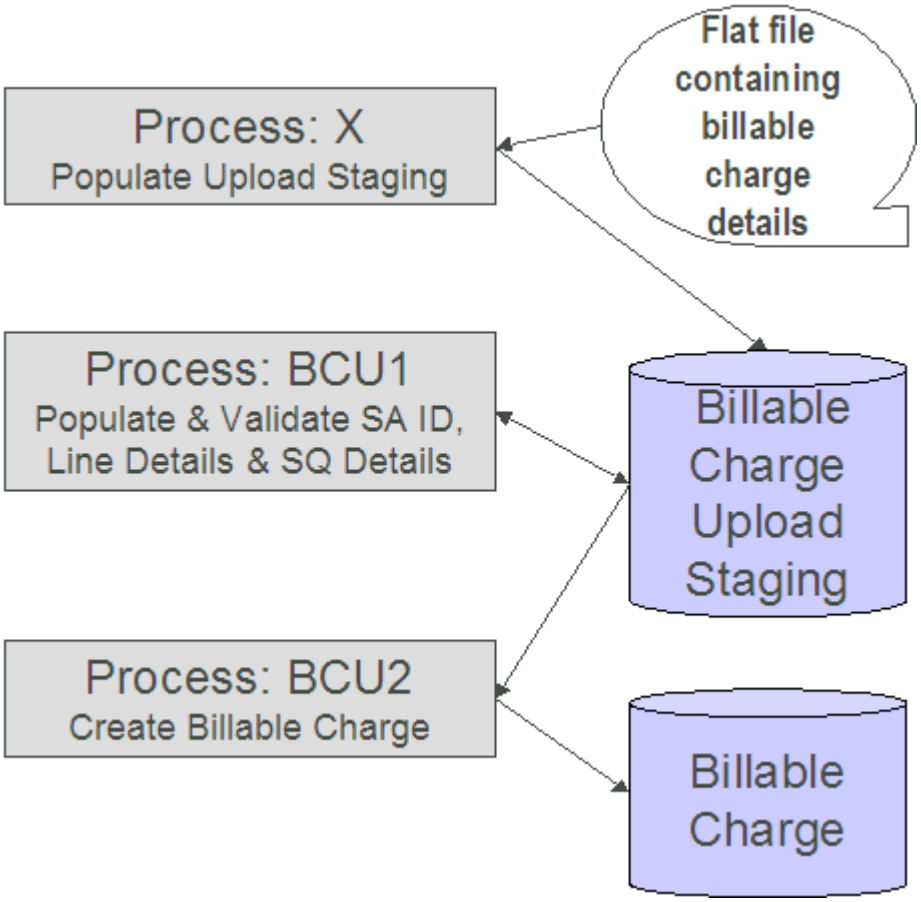
- Billable Charge Upload Background Processes
- BCUP-PRG - Purge Billable Charge Upload Objects

Copyright © 2026, Oracle and/or its affiliates.

- Billable Charge Upload Staging
- Billable Charge Upload Exception

Billable Charge Upload Background Processes

The following diagram illustrates the processes involved in the uploading of billable charges into the system.



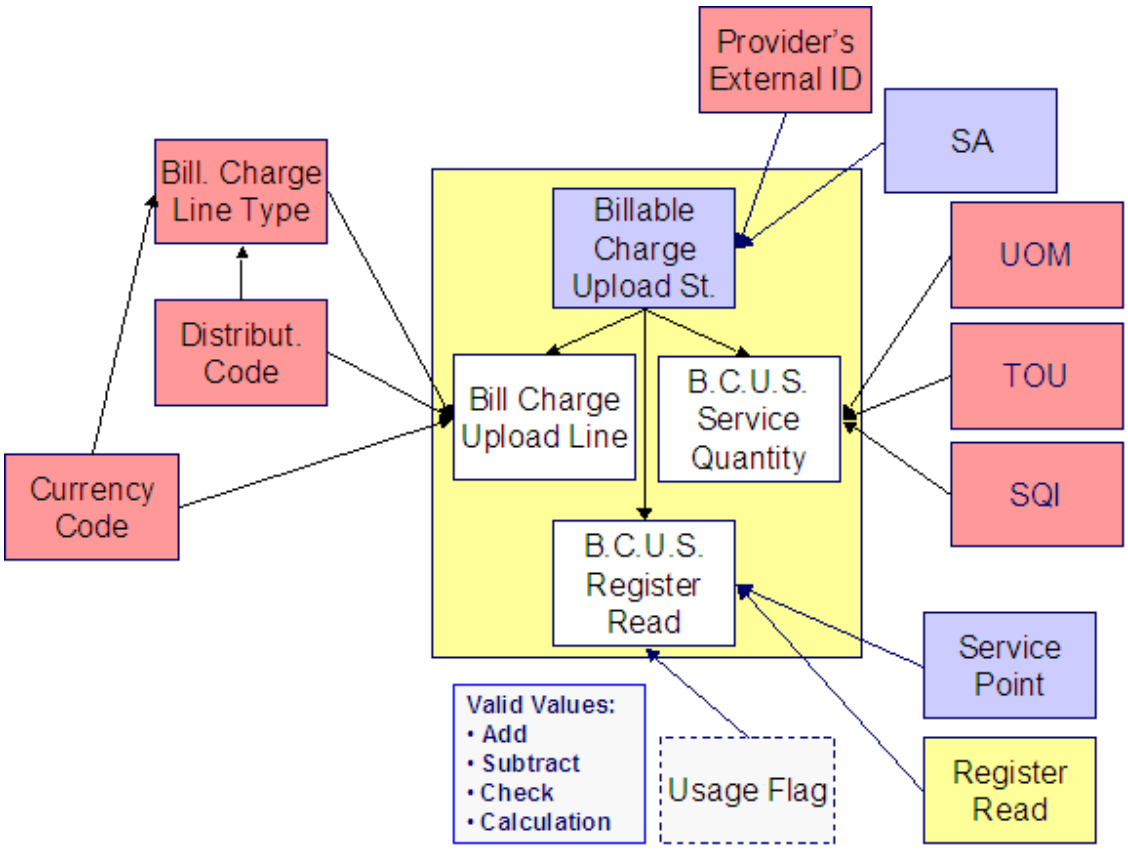
The topics in this section describe how these processes work.

- Process X - Populate BC Upload Staging
- BCU1 - Validate & Populate Billable Charge Upload Staging
- BCU2 - Create Billable Charge

Process X - Populate BC Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the yellow section of the following ERD).

Copyright © 2026, Oracle and/or its affiliates.



The topics in this section describe each of these tables.

- Billable Charge Upload Staging Layout
- Billable Charge Line Upload Staging Layout
- Billable Charge Line Characteristic Upload Staging Layout
- Billable Charge Service Quantity Upload Staging Layout
- Billable Charge Read Details Upload Staging Layout

Billable Charge Upload Staging Layout

You must create an upload staging record for each billable charge. The name of this table is **CI_BCHG_UP**. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments

Copyright © 2026, Oracle and/or its affiliates.

BCHG_UP_ID	30	Y	Char	This is the unique identifier of the record. This value does NOT have to be a random number, but it does need to be unique. If your process that inserts records on this table is capable of calling a routine, call ProgramCIPZKEYG and it will supply a 12 digit random number for you.
BCHG_UP_STAT_FLG	2	Y	Char	This must be set to P for Pending.
NT_XID_CD	30	N	Char	This is the external system id of the sender of the billable charge. See the description of the SA_EXT_REFT_ID field for how this field is used to determine the billable charge's SA_ID.

Copyright © 2026, Oracle and/or its affiliates.

SA_EXT_REF_ID	36	N	Varchar2	This field along with NT_XID_CD are used to uniquely identify the service agreement against which the billable charge will be stored. These two fields are used to identify a SA relationship record; the SA relationship record, in turn, has a sub SA and it is this SA that the billable charge will be linked to. Note well: these fields are only necessary if you do not know the billable charge SA ID. Refer to Maintaining SA Relationships for more information.
BCHG_EXT_REF_ID	50	N	Varchar2	This is the identifier of the billable charge in the sender's system.
CRE_DTTM	26	N	Date/Time	The date and time the upload staging row was inserted (this can be used for audit purposes).
START_DT	10	Y	Date	The start date of the period encompassed by the billable charge.
END_DT	10	Y	Date	The end date of the billable charge period.
DESCR_ON_BILL	254	Y	Varchar2	This is the description that should prefix the charges on the printed bill.

Copyright © 2026, Oracle and/or its affiliates.

SA_ID	10	See note	Char	This must correspond with a SA ID of a billable charge service agreement. You may leave this column blank if NT_XID_CD and SA_EXT_REF_ID are populated.
BILLABLE_CHG_ID	12	N	Char	Leave this column blank. It will be assigned by the system when it creates a billable charge record.

Billable Charge Line Upload Staging Layout

You must create a billable charge line upload record for each line to be uploaded. The name of this table is **CI_BCHG_LINE_UP**. The following table describes each column on this table.

> **Note**
>
> **Defaulting from BCHG_UP_XTYPE.** Please pay special attention to the **Comments** column below as many fields, if left blank, will default from each record's BCHG_UP_XTYPE.

Column Name	Length	Req'd	Data Type	Comments
BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
LINE_SEQ	3	Y	N	This is the unique line number of the billable charge line. This value must be unique within the billable charge.
DESCR_ON_BILL	254	Y	Varchar2	This is the description that will be printed on the bill for the line.
CHARGE_AMT	13.2	N	N	This is the amount associated with the billable charge line.

Copyright © 2026, Oracle and/or its affiliates.

BCHG_LINE_XID	20	N	Varchar2	This is the unique identifier of the billable charge line in the sender's system.
BCHG_UP_XTYPE	30	N	Char	This is the type of billable charge line. This is a foreign key reference to the billable charge line type control table. This control table is used to populate the remaining fields (if they are blank). If this field is blank, the remaining fields must be specified as they cannot be defaulted from the billable charge line type control table. Refer to Billable Charge Line Type for more information.
CURRENCY_CD	3	See note	Char	This is a foreign key reference to the currency code of the billable charge line. This field must be specified if CHARGE_AMT is non-zero.
SHOW_ON_BILL_SW	1	See note	Char	This should be Y if the line should appear on the customer's printed bill. This should be N if the line should not appear on the customer's printed bill. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.

Copyright © 2026, Oracle and/or its affiliates.

APP_IN_SUMM_SW	1	See note	Char	This should be Y if the line appears in a summary line. This should be N if the line does not appear in a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.
MEMO_SW	1	See note	Char	This should be Y if the line does NOT affect the general ledger. This should be N if the line affects the general ledger. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.
DST_ID	10	See note	Char	If the line affects the general ledger, the set ID (the previous field) and the distribution code of the GL account must be specified. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.

Billable Charge Line Characteristic Upload Staging Layout

If you want to upload characteristics on your billable charge lines, you must create a billable charge line characteristic upload record for each characteristic on each line. The name of this table is **CI_B_LN_UP_CHAR**. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments

Copyright © 2026, Oracle and/or its affiliates.

BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
LINE_SEQ	3	Y	N	This is the foreign key to the billable charge line upload staging record
CHAR_TYPE_CD	8	Y	Char	This is the unique identifier of the billable charge line characteristics. This value must be unique within the billable charge line characteristics.
ADHOC_CHAR_VAL	30	N	Varchar2	An ad hoc characteristic value must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.
CHAR_VAL	16	N	Char	A characteristic value must be supplied if the ad hoc switch on characteristic type is N. Otherwise, it is not allowed.
CHAR_VAL_FK1	50	N	Varchar2	A characteristic value foreign key must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.
CHAR_VAL_FK2	50	N	Varchar2	A characteristic value foreign key must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.

Copyright © 2026, Oracle and/or its affiliates.

CHAR_VAL_FK3	50	N	Varchar2	A characteristic value foreign key must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.
CHAR_VAL_FK4	50	N	Varchar2	A characteristic value foreign key must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.
CHAR_VAL_FK5	50	N	Varchar2	A characteristic value foreign key must be supplied if the ad hoc switch on characteristic type is Y. Otherwise, it is not allowed.

Billable Charge Service Quantity Upload Staging Layout

You must create a billable charge service quantity upload record for each service quantity to be uploaded. The name of this table is **CI_BCHG_UP_SQ**. The following table describes each column on this table. Refer to "pass through" charges for more information about how this type of information is used.

Column Name	Length	Req'd	Data Type	Comments
BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
SEQ_NUM	3	Y	N	Sequence of the service quantity within the Billable Charge Upload

Copyright © 2026, Oracle and/or its affiliates.

UOM_CD	30	N	Varchar	The unit of measure of the Billable Charge Upload SQ row. If specified, it must reference a valid value on CI_UOM. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
TOU_CD	30	N	Char	The time of use of the Billable Charge Upload SQ row. If specified, it must reference a valid value on CI_TOU. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
SQI_CD	30	N	Varchar	The service quantity identifier of the Billable Charge Upload SQ row. If specified, it must reference a valid value on CI_SQI. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
SVC_QTY	12.6	N	N	Quantity of the service used for the UOM / TOU / SQI combination

Billable Charge Read Details Upload Staging Layout

You must create a billable charge read details upload record for each read detail to be uploaded. The name of this table is **CI_BCHG_UP_READ**. The following table describes each column on this table. Refer to "pass through" charges for more information about how this type of information is used.

Column Name	Length	Req'd	Data Type	Comments

Copyright © 2026, Oracle and/or its affiliates.

BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
SP_ID	10	Y	Char	Note, if USAGE_FLG is calculation, you can set this field to 0 if you do not know the SP ID. The only thing that's important is that the combination of BCHG_UP_ID, SP_ID and SEQNO are unique.
SEQNO	5	Y	Number	This should be a value greater than zero. See SP_ID comment for more information. This is typically the sequence number of the register on the meter.
END_READ_DTTM	26	Y	DateTime	This is the date and time of the end read
END_REG_READ_ID	12	N	Char	This field must reference an existing register read ID if **USAGE_FLG** is anything other than a calculation.
END_REG_READING	9.6	Y	Number	This is the reading of the end read.
FINAL_REG_QTY	12.6	Y	Number	This is the final amount of the reading (this is the amount of billable consumption the read detail contributed to the entire billed consumption).

Copyright © 2026, Oracle and/or its affiliates.

FINAL_SQI	8	N	Char	The service quantity identifier of the FINAL_REG_QTY. If specified, it must reference a valid value on CI_SQI. Note, at least one of the following fields should be specified: FINAL_SQI, FINAL_UOM_CD.
FINAL_TOU_CD	8	N	Char	The time-of-use code of the FINAL_REG_QTY. If specified, it must reference a valid value on CI_TOU.
FINAL_UOM_CD	4	N	Char	The unit of measure of the FINAL_REG_QTY. If specified, it must reference a valid value on CI_UOM. Note, at least one of the following fields should be specified: FINAL_SQI, FINAL_UOM_CD.
HOW_TO_USE_FLG	2	Y	Char	See the field description for valid values (**CI_BCHG_UP_READ**).
MSR_PEAK_QTY_SW	1	Y	Char	Y or N
MSR_QTY	12.6	Y	Number	This is the difference between **END_REG_READING and START_REG_READING for subtractive registers.**

Copyright © 2026, Oracle and/or its affiliates.

REG_CONST	6.6	Y	Number	This is the register's constant (the value used to adjust the measured quantity due to something associated with the physical meter). Set this to a value of 1 if it is not known.
SQI_CD	30	N	Varchar	The service quantity identifier of **END_REG_READING and START_REG_READING.** If specified, it must reference a valid value on CI_SQI. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
START_READ_DTTM	26	Y	DateTime	This is the date and time of the start reading.
START_REG_READ_ID	12	Y	Char	This field must reference an existing register read ID if **USAGE_FLG** is anything other than a calculation.
START_REG_READING	9.6	Y	Number	This is the reading of the start read.
TOU_CD	30	N	Char	The time-of-use code of **END_REG_READING and START_REG_READING.** If specified, it must reference a valid value on CI_TOU.
UOM_CD	30	Y	Char	The unit of measure of **END_REG_READING and START_REG_READING**.

Copyright © 2026, Oracle and/or its affiliates.

USAGE_FLG	2	Y	Char	See the field description for valid values (**CI_BCHG_UP_READ**).
USE_PCT	3	Y	Number	This is the percentage of the consumption that was billed (this will typically be 100).

BCU1 - Validate & Populate Billable Charge Upload Staging

This process has several functions:

- It populates the SA ID on all BC upload staging records where SA ID is left blank. This means that you don't have to know the SA ID in order to interface a billable charge into the system. The program populates SA ID with the service agreement related to the SA relationship with the most recent effective date for a given combination of service provider's external system ID and customer's ID in the sender's system. The system can only do this if the following points are true:
- You use the system's SA relationship functionality to define that you provide bill ready billing services for a service provider. On the customer's SA relationship record you must define the customer's ID in the service provider's system. Refer to Maintaining SA Relationships for more information.
- You identify the service provider's external system ID on the billable charge upload staging record.
- You identify the customer's ID in the sender's system.

> **Note**
>
> You can still interface billable charges if you don't use the system's SA relationship functionality. How? You simply need to derive the SA ID associated with the uploaded billable charge.

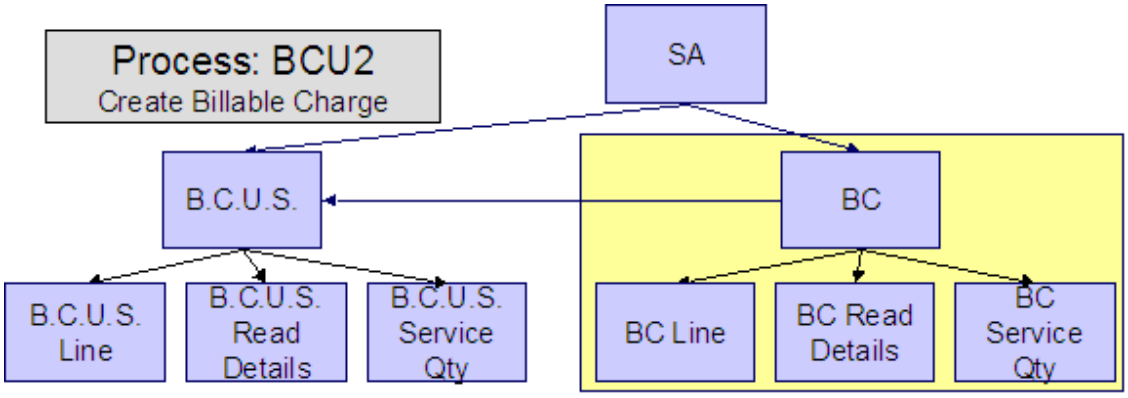
- It populates various fields (e.g., the GL distribution code, memo only switch) on the billable charge upload line records from the billable charge line type specified on each respective record.

Any validation / population errors detected during this process are written to the BC Upload Exception table. You can fix errors using BC Upload Staging page (don't forget to change the record's status back to Pending).

BCU2 - Create Billable Charge

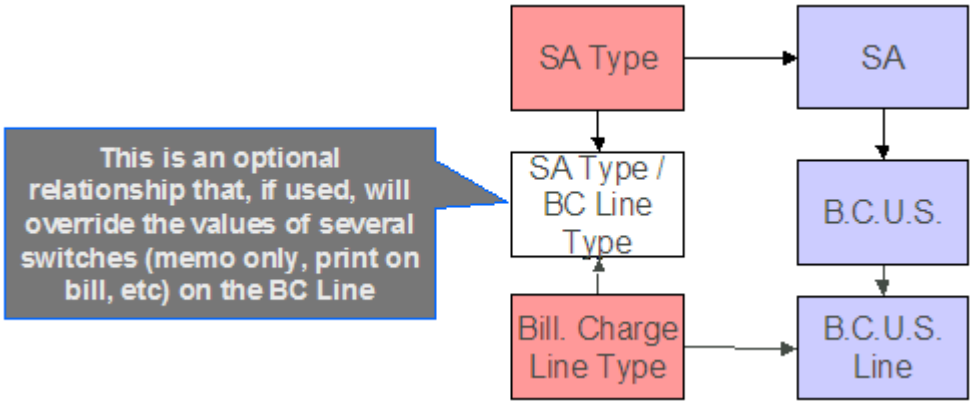
This process creates a billable charge and billable charge lines (shown in the yellow section of the following ERD) for all BC upload records in the Pending state.

Copyright © 2026, Oracle and/or its affiliates.



If the billable charge's SA is Closed, it will Reactivate it so that the billable charges will be swept onto the next bill produced for the account.

> **Note**
>
> This process will override the values of the various switches referenced on billable charge upload staging line if the respective service agreement's SA type has an override value for the line's billable charge line type. Refer to SA Type - Billable Charge Overrides for more information.



Any errors detected during this process are written to the Billable Charge Upload Exception table. You can fix errors using BC Upload Staging page (don't forget to change the record's status back to Pending).

BCUP-PRG - Purge Billable Charge Upload Objects

Completed billable charge upload staging objects should be periodically purged from the system by executing the BCUP-PRG background process. This background process allows you to purge all Completed billable charge upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of Completed billable charge upload objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge Completed billable charge upload objects as they exist only to satisfy auditing and reporting needs.

Copyright © 2026, Oracle and/or its affiliates.

Billable Charge Upload Staging

The Billable charge Upload Staging page has three purposes:

- You can view historical billable charge upload records.
- You can correct billable charge upload records that are in error.
- You can add new billable charge upload records. These will be uploaded by the billable charge upload process (although it's probably easier to just add the billable charges using the Maintaining Billable Charges page).

The topics in the section describe how to maintain this information.

- Billable Charge Upload - Main
- Billable Charge Upload - Lines
- Billable Charge Upload - SQ Details
- Billable Charge Upload - Read Details

Billable Charge Upload - Main

Open this page using **Menu**, and then **Financial**, and then **Billable Charge Upload Staging**, and then **Search.**

Description of Page

NOT_SUPPORTED

Refer to Billable Charge Upload Staging for more information about these fields.

Note

Correcting Errors. If a Billable Charge Upload record's status is Error , you should correct the problem and then change its status back to Pending. The next time the billable charge upload process runs, it will revalidate the record and, if valid, it will create a billable charge record.

Billable Charge Upload - Lines

Open this page using **Menu**, and then **Financial**, and then **Billable Charge Upload Staging**, and then **Search** and navigate to the **Lines** page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Billable Charge Line Upload Staging for more information about these fields.

Billable Charge Upload - SQ Details

Open this page using **Menu**, and then **Financial**, and then **Billable Charge Upload Staging**, and then **Search** and navigate to the **SQ Details** page.

Description of Page

NOT_SUPPORTED

Refer to Billable Charge Service Quantity Upload Staging for more information about these fields.

Billable Charge Upload - Read Details

Optionally, you may specify meter read details associated with a billable charge. These are most commonly used when you need to print the read details that were used to calculate the bill lines that appear on the main tab.

Note

Using these reads to derive billable consumption. The base package bill segment creation algorithm (BSBS-BC) simply sweeps billable charge read details onto the related bill segment's read detail. This algorithm does NOT use these reads to derive billable consumption. This is intentional as most implementations that import reads also import billable consumption and therefore deriving billable consumption is unnecessary (and also error prone). If your implementation needs to derive billable consumption from these reads, you can develop your own bill segment creation algorithm to do just this.

Open **Menu**, and then **Financial**, and then **Billable Charge Upload Staging**, and then **Search** and navigate to the **Read Details** page to view this information.

Description of Page

NOT_SUPPORTED

Refer to Billable Charge Read Details Upload Staging for more information about these fields.

Copyright © 2026, Oracle and/or its affiliates.

Billable Charge Upload Exception

If errors are detected during the billable charge upload process, a record is written to the billable charge upload exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BCUPL background process. This process generates a To Do entry for every record in the billable charge upload exception table.

You can fix this error using the Billable Charge Upload Staging page and change the status of the record from Error to Pending. When the billable charge upload process next runs, it attempts to upload this record again.

Budget Review

This portal shows account budgets that require manual review, as identified by the Budget Monitor process.

You can access this portal from **Menu**, and then **Financial Query**, and then **Budget Review.**

Refer to the embedded help text for more information.

Payments

In this section, we describe how to manage your customer's payments.

- The Big Picture of Payments
- Maintaining Payment Events
- Payment Event Quick Add
- Payment Quick Add
- Maintaining Payments
- Payment Portal
- How To
- Financial Transactions On A Payment
- Payment History
- Payment / Tender Search
- Maintaining Deposit Controls
- Maintaining Tender Controls
- Interfacing Payments From External Sources

Copyright © 2026, Oracle and/or its affiliates.

The Big Picture of Payments

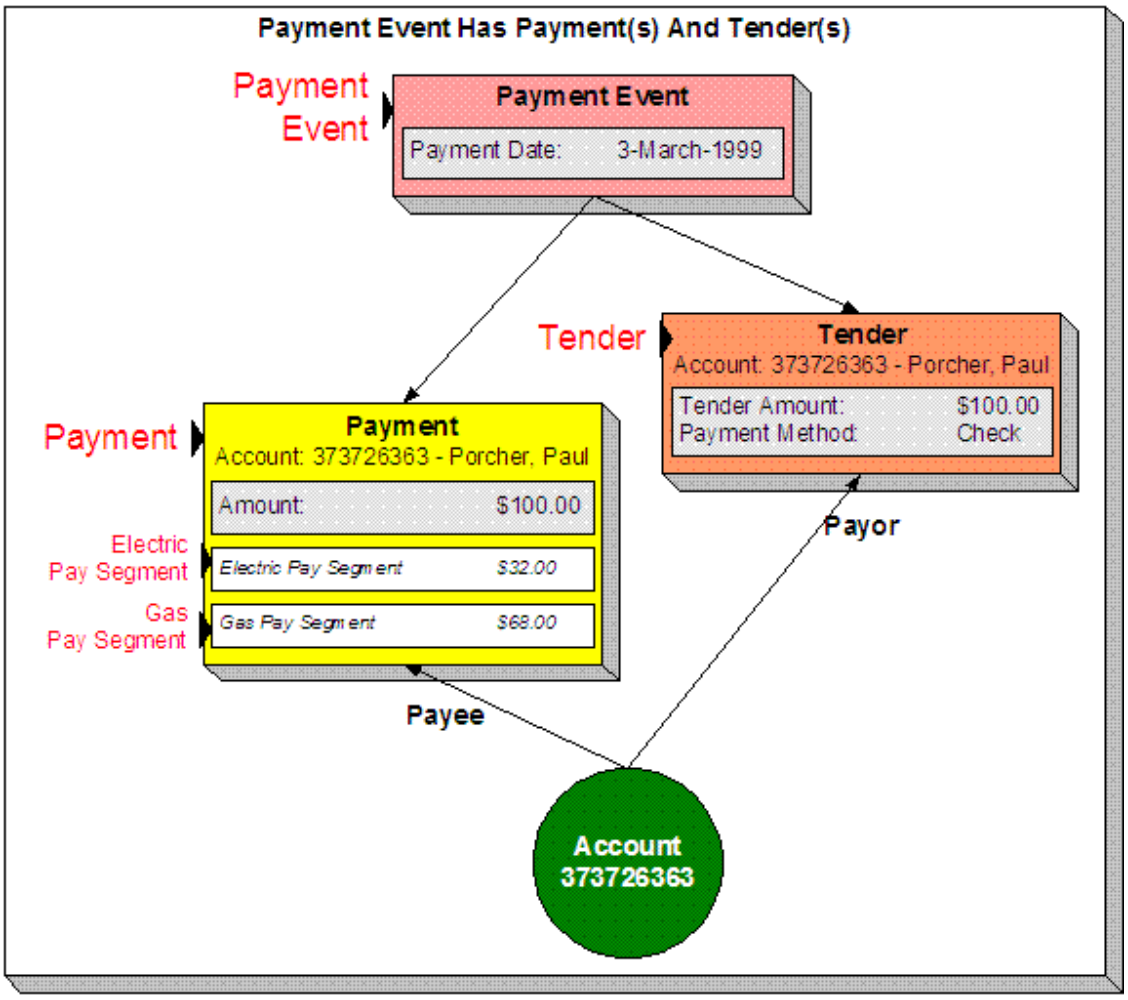
A payment reduces how much an account owes. The topics in this section provide background information about a variety of payment topics.

- A Payment Event Has Payments And Tenders
- Multiple Tenders Used To Pay For Multiple Accounts
- An Overview Of The Payment Event Creation & Allocation Process
- Distributing A Payment Event
- Distributing A Payment Amongst An Account's Service Agreements
- Overpayment
- Canceling A Tender Versus Canceling A Payment
- NSF Cancellations
- Transferring A Payment
- Unbalanced Payment Events
- How And When Payments Appear On Bills
- Tender Management and Workstation Cashiering
- Payment Financial Transaction Considerations
- A Payment May Affect More Than Just Customer Balances
- Automatic Payments
- Third Party Payment Processing
- Issuing A Payment Advice Instead Of Creating An Automatic Payment

A Payment Event Has Payments And Tenders

The explanation in The Financial Big Picture provides an accurate, but incomplete view of payments. The missing pieces concern *payment events* and *tenders*. The following diagram illustrates the difference between a payment event, its payment(s) and its tender(s).

Copyright © 2026, Oracle and/or its affiliates.



The following concepts are illustrated above:

A payment event defines the event A payment event is required whenever any form of payment is received. The payment event defines the payment date (and that's all).

A payment event has tender(s) A tender exists for every form of tender remitted as part of the payment event. A payment event must have at least one tender otherwise nothing was remitted. A payment event may have many tenders when multiple payment methods are associated with an event (e.g., paying with cash, a check, and a credit card).

A payment is allocated to account(s) The total amount of tenders under a payment event is distributed to one or more accounts.

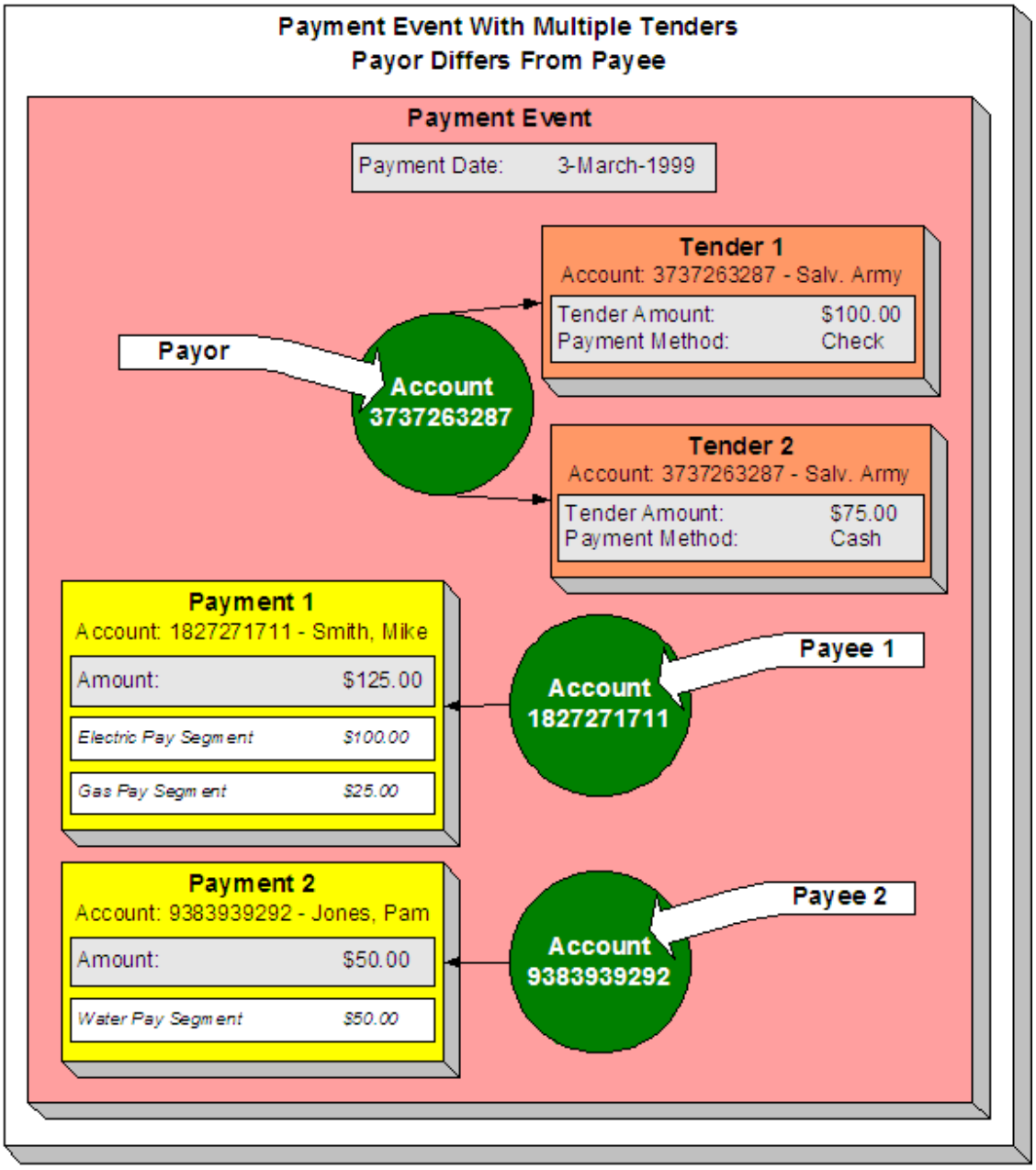
A payment is distributed to SAs The system allocates an account's payment amount amongst its service agreements. The system creates a payment segment for each service agreement that receives a portion of the payment.

Payor and payee are frequently the same The account remitting the tender (the payor) is frequently the same as the account to which the funds are allocated (the payee). The next illustration provides an example when this is not the case.

Multiple Tenders Used To Pay For Multiple Accounts

Copyright © 2026, Oracle and/or its affiliates.

The following diagram illustrates a payment event with multiple tenders where the payor of the tender is not the same as the account(s) receiving the payment.



A payment event may have many tenders A single payment event may have many tenders. While the above example shows both tenders being paid by the same account, each tender may reference a different account.

Many accounts may be paid under 1 event The total amount of tenders under a payment event are distributed to one or more accounts.

Payor may differ from payee The account(s) remitting the tender may differ from the account(s) whose debt is relieved.

Copyright © 2026, Oracle and/or its affiliates.

An Overview Of The Payment Event Creation & Allocation Process

When a payment event occurs, the system stores a tender for each form of remittance (e.g., cash, check, charge). It then allocates the sum of the tenders to one or more accounts.

By default, the system allocates the sum of the tenders to the account that remits the tenders. You may override this default and specify any number of accounts and their respective payment allocation amount. This is useful, for example, when a social service agency pays for many accounts. If applicable, you may also configure the system to use your own payment event distribution rule(s).

NOT_SUPPORTED

Refer to Distributing A Payment Event for more information.

The system distributes a payment amongst an account's service agreements based on the age of each service agreement's debt AND distribution priority. The system creates a payment segment for each service agreement that receives part of the payment.

NOT_SUPPORTED

Refer to Distributing A Payment Amongst An Account's Service Agreements for more information.

You may manually redistribute the payment amount amongst the account's service agreements before you commit the distribution. When the distribution is acceptable, you freeze the payment. Freezing a payment causes the system to create a financial transaction for each related payment segment. It is the financial transaction(s) that causes the service agreements' payoff and current balances to be reduced. The financial transaction also contains the journal details that debit "cash" and credit some other GL account.

And that's it. The remaining topics in this section provide more information about the creation and allocation of payment events.

Note

Batch and real-time payment event creation / allocation. There is only one payment event creation / allocation routine and therefore anything the batch payment process does for whole batches of payments, you can do to a payment on-line.

Distributing A Payment Event

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> This section deals with the concept distributing a payment amount into payment(s). It does not discuss the distribution of a single payment into segments. **For more information** about payment distribution, refer to Distributing A Payment Amongst An Account's Service Agreements.

The base-package, by default, creates a single payment for a payment event. Some business practices require potentially many payments to be created when payment events are added.

A few examples of when multiple payments may be necessary are:

- A payment amount needs to be distributed towards different distribution types:
  - $50 in interest
  - $60 in collection charges
  - $70 in taxes
- Each of the above distributions is realized as a separate payment identified by its own match type (i.e., there'll be one match type called Interest, another called Collection Charges, etc.).

In this example, the debt of a single service agreement may be relieved by each of these payments.

- In a similar way, you may want to create a separate payment for an overpayment to differentiate it from regular payments (using yet another match type).
- In the case of a social service agency that pays for many accounts, a single payment event may be distributed amongst multiple accounts.

The method by which a payment amount is distributed to create payment(s) is contained in Create Payment algorithms plugged in on a distribution rule.

There is yet another aspect to having control over how payment events are created. The default method of creating payment events assumes knowledge of account IDs (of the payor and the payee) when making a payment. In cases where payments are made by and towards business entities other than accounts, knowledge of their corresponding account IDs may not be available at payment time. Consider the following examples:

- A payment is made to pay a person's bill, providing the person's identification at payment time.
- A payment is made towards a specific service agreement using a "contract number" defined as SA characteristic.

The method by which the tender account is determined by means of an alternate identifier is contained in Determine Tender Account algorithm plugged in on a distribution rule.

> **NOT_SUPPORTED**
>
> Refer to Making Payments Using Distribution Rules for information on how to configure your system to use this distribution method.

Copyright © 2026, Oracle and/or its affiliates.

Distributing A Payment Amongst An Account's Service Agreements

> **Warning**
>
> This section deals with the concept of distributing a payment amongst an account's service agreements. It does not discuss how the sum of a payment event's tenders is balanced out by payment allocations. **For more information** about payment event balance, refer to Unbalanced Payment Events.

A payment must be distributed to one or more service agreements for its financial impact to be realized. When a payment satisfies an account's entire debt, you don't have to worry about how the system distributes the payment. The concept of payment distribution is only relevant when a partial or excess payment is distributed.

The first important point to understand is that the method of distributing a payment amongst an account's service agreements is contained in an algorithm that's plugged in on to Customer Class. This means that you can have different distribution algorithms for different customer classes.

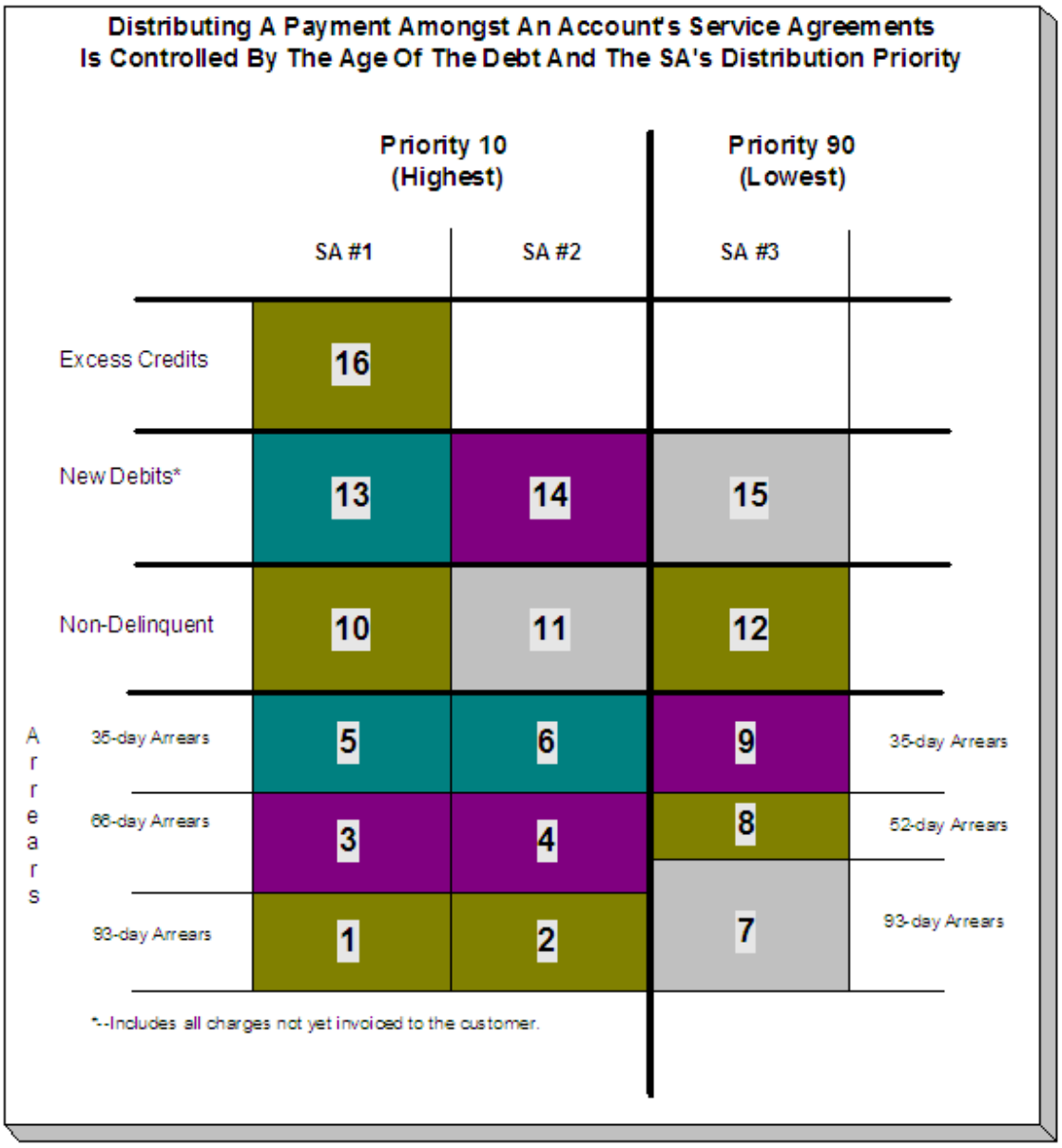
> **Note**
>
> **Manual overrides.** Most of the time, you'll let the payment distribution algorithm distribute the payment amongst an account's service agreements. However, you may manually distribute a payment when a customer directs a payment to specific service agreement(s).

The following explanation describes one of the base package payment distribution algorithms (refer to Payment Distribution - Pay Priority and Debt Age (PYDIST-PPRTY) for information about this algorithm). This algorithm distributes a payment based on:

- The age of each service agreement's debt.
- The payment distribution priority of each service agreement's SA type.

The following diagram helps illustrate how the distribution algorithm works.

Copyright © 2026, Oracle and/or its affiliates.



The above example shows three columns, one for each service agreement linked to a hypothetical account. Notice that two of the service agreements have the same distribution priority, the third has a lower priority. The numbers in the cells indicate the order in which the system distributes a partial payment.

> **Note**
>
> **Debt terminology.** Before we can discuss the distribution algorithm, you must understand the terminology we use to categorize debt. **Delinquent** debt is associated with financial transactions that appear on overdue bills. **Non-delinquent** debt is associated with financial transactions that appear on current bills. **New debits** debt is associated with financial transactions that do not yet appear on a completed bill (e.g., a late payment charge that hasn't been billed yet).

The following points describe the algorithm used to distribute the partial payment:

Copyright © 2026, Oracle and/or its affiliates.

- The system pays off delinquent debt of the highest priority service agreements first. In the above example, where multiple service agreements have the same distribution priority, the system does NOT payoff one service agreement before it starts on the next (which one would it pick?). Rather, it distributes the payment amongst the service agreements based on the age of the respective debt on each service agreement. In the above example, this is represented by steps 1 through 6 (notice how the distribution jumps between SA1 and SA2).
- After all delinquent debt has been relieved from the highest priority service agreement(s), the system pays off the next priority until all delinquent debt is relieved. In the above example, this is represented by steps 7 through 9.
- The system next pays off non-delinquent debt using each service agreement's respective distribution priority. Note well, the payment distribution algorithm doesn't associate an age with non-delinquent debt and therefore the distribution is based purely on the service agreements' respective distribution priority. In the above example, this is represented by steps 10 through 12.
- After all non-delinquent debt is relieved, the system next pays off "new debit" debt based on the service agreements' respective distribution priority. In the above example, this is represented by steps 13 through 15.
- Refer to Overpayment for a description of what happens if money still exists after the above distribution is complete.

> **Note**
>
> **Payment segments and financial transactions.** A payment segment exists for each service agreement that receives a portion of a payment. Linked to each payment segment is a financial transaction. It is the financial transaction that causes the service agreement's debt to be relieved and the general ledger to be impacted.

> **NOT_SUPPORTED**
>
> Refer to Payment Exception for more information about how the system handles errors detected during the payment distribution process.

> **Note**
>
> **Overriding the distribution algorithm for uploaded payments.** The standard distribution algorithm is used for payments that are Interfaced From An External System unless you specify a MATCH_VALUE and MATCH_FLG on the Payment Staging row associated with the uploaded payment. These fields are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_FLG indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID).
>
> **Open item customers.** For an open-item customer, you MUST override the standard distribution algorithm because the payment is distributed as per the open items that it is relieving. Refer to Payments And Match Events for more information.

Copyright © 2026, Oracle and/or its affiliates.

Overpayment

Overpayment refers to the situation where money is left over after a payment has been distributed to all eligible service agreements, and all debt is relieved. Refer to Overpayment Segmentation for a description on how to configure the system to handle your overpayment requirements.

Canceling A Tender Versus Canceling A Payment

A payment event has tender(s) and payment(s). You can cancel a tender when it's not valid, e.g., when a check bounces. You can cancel a payment when the account should not have received the payment (e.g., a misdistribution or a canceled tender).

When you cancel a tender, the system automatically cancels ALL frozen payments. We do this because if the tender is canceled, there are no funds to distribute to accounts (unless there are other non-canceled tenders under the event). However, when you cancel a payment, the system does NOT cancel the tender(s) because we assume that, if the tenders were incorrect, you would have canceled them rather than the payment.

NSF Cancellations

When a tender is canceled, a cancellation reason must be supplied. If the cancellation reason indicates a NSF (non sufficient funds) charge should be levied, the system invokes the NSF charge algorithm specified on the tender's account's customer class. Algorithms of this type will typically create an adjustment or billable charge to levy the NSF charge. Refer to NSFC-DFLT for an example of such an algorithm.

> **Note**
>
> **The payor gets the NSF adjustments.** It's possible for the payor of a tender to differ from the payee (e.g., when a social service agency pays for other customers' debt). When you cancel a tender you must specify a cancellation reason. If the cancellation reason indicates an NSF charge should be levied, the sample algorithm provided with the system levies the NSF charge on the PAYOR's account. Because adjustments must be linked to a service agreement, the algorithm picks the payor's service agreement with the highest payment distribution priority and levies the adjustment specified in the service agreement's SA type's NSF adjustment type.

Besides calling this algorithm, a NSF cancellation may affect the tendering account's credit rating and cash-only score. The cancellation reason indicates the extent to which the account's ratings are affected.

And finally, whenever a tender is canceled due to non-sufficient funds, the account is scheduled for review by the Account Debt Monitor.

Copyright © 2026, Oracle and/or its affiliates.

Transferring A Payment

If the account on an event's tender and payment are wrong, you can use the Transfer button on the Payment Event page to transfer the payment to another account.

If the account on the payment is wrong - but the tender is correct, you can use the Transfer button on the Payment page to transfer the payment only to another account.

Unbalanced Payment Events

The system, by default, distributes the sum of a payment event's tenders to the account that remits the tender with a single payment. After distribution the sum of the tenders equals the sum of the payments when the event is first created. We refer to such an event as being balanced.

However, it is possible for an event's tender amount to not equal the sum of the payment allocations (i.e., the event becomes unbalanced). How? Well, there are several ways this can happen:

- While the system DEFAULTS the payment amount to be the tender amount, you can override the payment amount and therefore make a previously balanced event unbalanced.
- While the system DEFAULTS the payment account to be the tender account, you can add additional accounts / amounts and therefore make a previously balanced event unbalanced.
- When you cancel a tender (e.g., because a check bounces), the system cancels ALL payments linked to the tenders payment event. If the payment event has multiple tenders, this will cause the event to become unbalanced. To correct this situation, you must add payment allocations to equal the amount of uncanceled tenders.
- If you cancel a payment and forget to add another payment for the same amount, the event becomes unbalanced. To correct this situation, you must add another payment (or cancel the tender).
- You may delete a tender from an event while its tender control is open. If you delete a tender and don't do anything about the related payments, the event becomes unbalanced.
- You may add a tender to an event at any time. If you don't allocate the tender amount to an account, the event becomes unbalanced.

NOT_SUPPORTED

Refer to Payment Event Exception for more information about how the system reminds you about unbalanced payment events.

How And When Payments Appear On Bills

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Financial Transactions Created Between Bills for more information.

Tender Management and Workstation Cashiering

When you add a tender, you must identify its **Tender Source**. For example,

- A specific cash drawer ID is the source of tenders remitted to a cashier.
- The notional lockbox ID is the source of tenders interfaced from a lockbox.
- The remittance processor is the source of tenders interfaced from a remittance processor.

Note

Default note. When a user adds a tender control the system attempts to default a tender source based on the currency of the deposit control and the tender source(s) defined on the user's record. If your organization accepts alternate currency payments, then a tender source must exist for each currency code accepted at a location.

NOT_SUPPORTED

For more information, refer to Setting Up Tender Sources.

A tender source's tenders must be balanced against an expected total before they can be deposited at a bank. This periodic balancing requires all tenders to exist in respect of a **Tender Control**. Over time, a tender source may have many tender controls (one per balancing event).

An example of a cashier's cash drawer will help clarify the tender control concept:

- When a cashier starts in the morning, they start with a fresh cash drawer (i.e., one without tenders). Whenever a drawer starts afresh, a new Tender Control must be created because you balance the contents of a drawer.
- A cash drawer typically contains funds to make change. These funds are the tender control's **Opening Balance**.

Note

Default note. A tender control's starting balance defaults from its tender source.

- During the day, customers remit tenders to the cashier. Every tender put into the drawer is associated with the drawer's tender control created at the start of the day.
- A tender control's balance increases during the day as tenders are recorded. A cashier can view the balance at any time.

Copyright © 2026, Oracle and/or its affiliates.

- The cashier can turn in funds to the head cashier during the day. Each turn in event can be recorded in the system. Note well, if the amount of funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind them to turn in funds.
- At some point, the contents of the drawer must be balanced against the total tenders linked to the tender control. When balancing starts, no additional tenders may be put into the tender control. If the cashier receives additional tenders after balancing starts, a new tender control must be created (and the above process starts afresh).
- During the balancing process, some modifications may be made to the tenders associated with the tender control, but no additional tenders may be added. When the tender control is balanced, neither it nor its tenders may be modified.

> **NOT_SUPPORTED**
>
> For more information, refer to Managing Your Cash Drawers.

While the above explanation is true, it isn't complete. In addition to the requirement that a tender must reference a tender control, the tender control must refer to a **Deposit Control**. Deposit controls give you administrative control over all of the tender controls whose contents will be deposited en masse. The following concepts will help explain the power of deposit controls:

- As explained above, when a cash drawer is started afresh, a new tender control must be created. The tender control "holds" all new tenders received by the cashier.
- Similarly, when a tender control is created, it must reference a deposit control. During the day, a deposit control's tender controls are constantly changing. You can view the total impact of a deposit control's tender controls at any time.
- At some point, you will want to deposit the tenders received during the day. To do this, you must indicate how much will be deposited at the bank. This deposit amount must equal the sum of the tender controls linked to the cash drawer. When this deposit balancing starts, no additional tender controls may be associated with the deposit control.
- When the deposit total equals the sum of the tender controls, the deposit control becomes balanced and no changes may be made to it, its tender controls, or its tender controls' tenders.

> **NOT_SUPPORTED**
>
> For more information, refer to Managing Your Cash Drawers.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Background processes use the same concepts.** Tenders that are interfaced from external sources (e.g., lockboxes and remittance processors) make use of the concepts describe above. For example, tenders interfaced from a remittance processor are linked to a tender control and this tender control is linked to a deposit control. The main difference is that the background processes require no human intervention; the system automatically creates tender and deposit controls and sets their states to balanced when the interface concludes successfully.
>
> **The ACH activation process also creates tender and deposit controls.** Refer to Activating Automatic Payments for more information.

The topics in this section elaborate on the tender management concepts described above.

- Managing Your Cash Drawers
- Turn Ins
- Balancing By Tender Type
- Cash Back
- Managing Payments Interfaced From External Sources

Managing Your Cash Drawers

> **Warning**
>
> This section assumes you are familiar with the concepts described in The Lifecycle Of A Deposit Control and The Lifecycle Of A Tender Control.

There are many ways to handle the daily management of tenders received via cash drawers. It really depends on how your organization works. To help you understand the potential of the system, we'll continue the example started above.

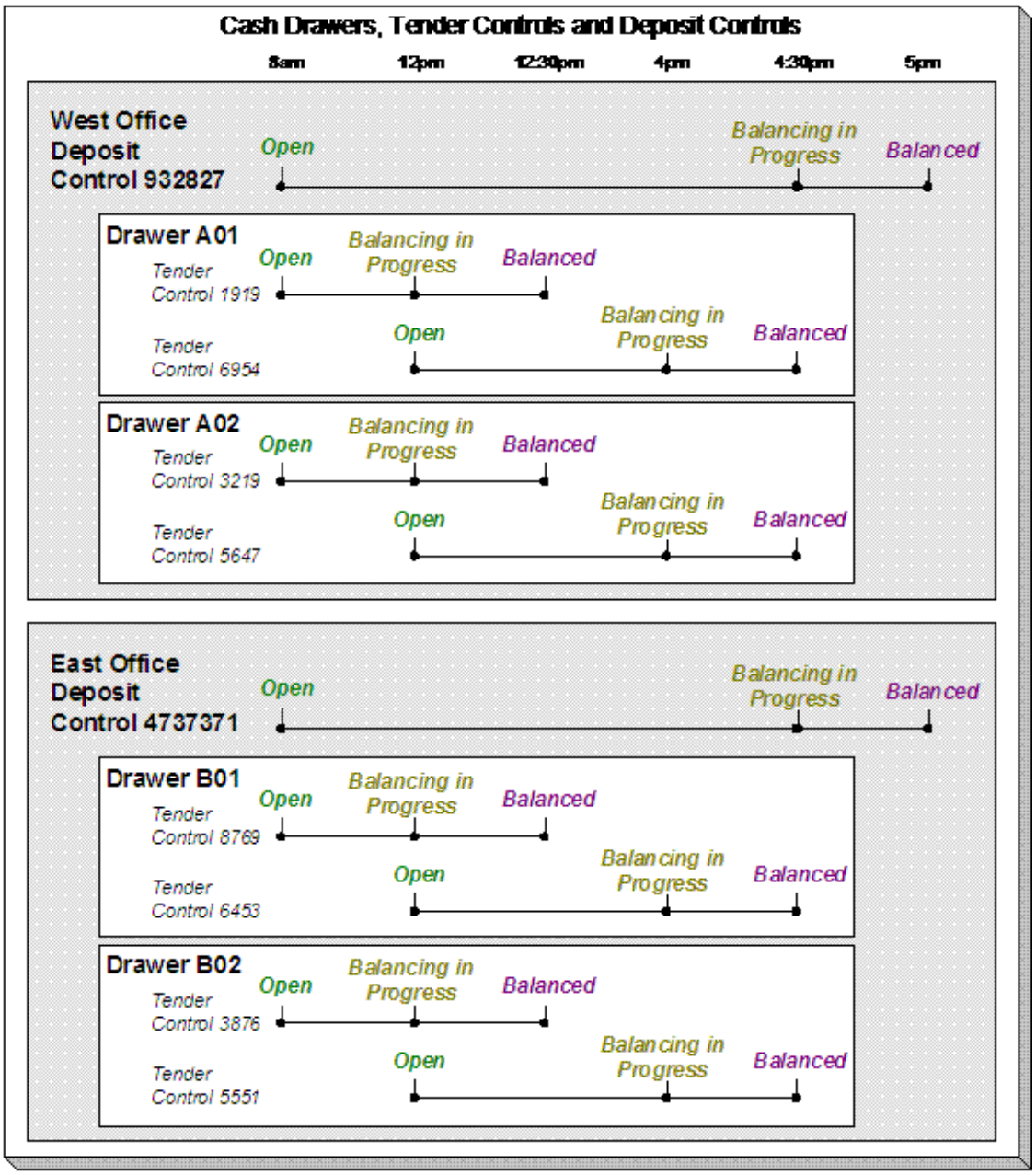
Assume that the cash drawers in your western office are balanced and deposited independently from those in your eastern office. We'll assume that both offices follows the same daily routine:

- Load fresh drawers first thing in the morning. Each drawer contains a starting balance of $150.00. Note: the drawer's tender control's starting balance defaults from its tender source.
- At 10 am, the cashier turns in funds to the chief cashier and continues to receive additional tenders.
- At 12 noon, each drawer is pulled and balanced by a supervisor.
- By 12:30 pm, the tender controls are balanced.
- At 4 pm, the cashiering stations are closed. Each drawer is pulled and balanced by a supervisor.

Copyright © 2026, Oracle and/or its affiliates.

- By 4:30 pm, the tender controls are balanced.
- At 5 pm, the deposit control is balanced and funds are ready to be deposited at the bank.

Given this, the following diagram illustrates the deposit controls and tender controls used by each office on a given day.



The following concepts are illustrated above:

- An Open deposit control must exist before you can create a tender control. And an Open tender control must exist before you can create a tender. From a business process standpoint, this means:
  - A supervisor would create a deposit control at the start of the day (8 am in the above illustration).
  - Each cashier would create a tender control when they start a drawer and reference the deposit control created by the supervisor.

Copyright © 2026, Oracle and/or its affiliates.

- During the day, the cashier can turn-in moneys to the chief cashier. These turn-in events are recorded in the system as they play a part in the ultimate balancing of the drawer. Refer to Turn Ins for more information.
- At some point, the contents of a drawer can be pulled and balanced. If additional tenders can be received in a drawer, a new tender control must be created for the drawer. Refer to Balancing By Tender Type for more information.
- At the end of the day, the supervisor checks to make certain that all tender controls linked to the deposit control are Balanced. After this has been done, the supervisor indicates the deposit amount on the deposit control and changes it to Balanced. Notice that in the above illustration each deposit control references four tender controls.

> **Note**
>
> **Alternate currency payments.** If your organization accepts payments in alternate currencies, that is, a currency other than the account's currency; a supervisor would create a deposit control for each such currency. Likewise, the cashier would create a tender control for each such currency.

> **NOT_SUPPORTED**
>
> Refer to Alternate Currency Payments for more information.

- Typically, a cash drawer has one tender control Open at any point in time (meaning that the tenders being received are being linked to a specific tender control). However, this is not a hard rule. If you want, you may have multiple tender controls Open at any point for a specific cash drawer (for example, if multiple cashiers can work the same drawer during the day but take their drawer with them).
- Typically, a specific cashier puts tenders into a specific tender control. However, this is not a hard rule. On a tender control, you can define if it's limited to a specific operator OR if any operator can link tenders to it.
- When you're ready to balance a drawer, you change the tender control to Balancing in Progress. This prevents new tenders from being added to the tender control. If the cashier can continue to receive tenders, they must create another tender control. In the above example, all drawers are balanced at 12 noon by a supervisor while the cashier continues to take payments.
- When the tender control is balanced, you change its state to Balanced. This prevents any changes to the tender control or its tenders.
- All tender controls exist in respect of a deposit control (in fact, the deposit control must be created before the tender control). This way, a supervisor can check the state of the related drawers throughout the day. Notice that the state transition of a deposit control is identical to that of the tender control (refer to The Lifecycle Of A Deposit Control and The Lifecycle Of A Tender Control). There is only a temporal difference. Notice that the deposit control stays open throughout the day while any number of tender controls are being opened and balanced.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multiple deposits in a day.** While the above example illustrates a single deposit per office per day, it is quite possible to have multiple deposit controls on any given day.
>
> **Turns ins.** The above example did not illustrate the fact that a cashier can turn-in moneys during the day without having to balance the drawer. Refer to Turn Ins for more information.

Turn Ins

A cashier may optionally turn-in funds received into a cash drawer to a head cashier. The turn-in process requires two steps:

- Each time a cashier turns-in funds, they add a turn-in event on the Tender Control - Turn Ins page. Note that a separate turn-in event is required for each type of tender that's turned in. This is because the balancing of a tender control is performed for each tender type and therefore the system must know how much of each tender type has been removed from the drawer.

> **Note**
>
> **Turn in warning.** If the amount of cash-like funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind them to turn in funds.

- The head cashier (the person responsible for the deposit control associated with the various tender controls) approves the turn in using the Deposit Control - Turn Ins page.

> **Note**
>
> **All turn-ins must be approved for balancing to complete.** A tender control cannot be balanced until the head cashier has approved all turn-in events.

Balancing By Tender Type

It should be noted that when it's time to balance a tender control, the cashier must enter the amount of each tender type that is in the drawer in order to balance it. For example, assume the following takes place:

- A drawer is opened with a starting balance of $150.50 (tender type is Cash).
- During the day, the cashier receives $5,000 in Cash, and $1,000 in Checks.
- During the day, the cashier turns in $750 or the Checks and $4,000 of Cash.

At the end of the day, the following operator would have to enter the balances shown in the Ending Balance column.

Copyright © 2026, Oracle and/or its affiliates.

Tender Type	Starting Balance	Tenders Received	Turn Ins	Ending Balance
Cash	$150.50	$5,000	$4,000	$1150.50
Check	-	$1,000	$750	$250

> **Note**
>
> **Time saver.** The system assists in the balancing effort by amalgamating the amount of tenders by tender type when the tender control's status is changed from Open to Balancing In Progress. The cashier just needs to enter the Ending Balance. The system can then compare the cashier's Ending Balance against the Expected Ending Balance. When these values are equal for all tender types, the tender control's status can become Balanced.

Cash Back

When a payment is added, the user defines the following:

- The amount of debt to be relieved (i.e., the payment amount)
- The amount remitted (i.e., the tender amount)
- The form of the remittance (i.e., the tender type)

The payment amount typically equals the tender amount unless cash will be returned to the customer. For example, if a customer remits $100, but only wants to pay off $25 of debt, the tender amount will be $100 and the payment amount will be $25. The system will only allow a user to remit more than the payment amount if the tender type indicates "cash back" is allowed. For example, you may not allow cash to be returned if a check is remitted, but you may allow it to be returned if cash is remitted.

If cash back is allowed for the tender type, the system displays the amount of cash to be returned on the Payment Event. In addition, because the system enforces balancing the cash drawer by tender type, the system adjusts the payment event's tendered information as follows:

- When there is cash back, the payment event will have two tenders - one will be for the amount and type entered by the user, the other will be a negative amount with a tender type of cash (note, this tender type is retrieved from the Starting Balance Tender Type on the installation record). For example, if a customer remits $100 in traveler's checks, but only wants to pay off $25 of debt, there will be two tenders: one for the $100 travelers check and the other for the -$25 of cash.

> **Note**
>
> **Alternate currency payments.** If your organization accepts payments in alternate currencies, that is, a currency other than the account's currency; the cash back tender is always created in the account's currency.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multiple tenders and payment cancellation.** If multiple tenders were created because of "cash back" processing, both tenders must be cancelled if the payment event needs to be cancelled.

When modifying an unfrozen payment on the Payment Event, if the payment becomes unbalanced, a button is displayed allowing the user to **Recalculate Cash Back**. If the user clicks this button, the system reassesses the cash back tender as follows:

- If there is now cash back, a new tender is created for the credit amount
- If the cash back amount has changed, the tender for the cash back is adjusted to the new amount
- If there is no longer cash back due, the tender for the cash back is removed.

Managing Payments Interfaced From External Sources

Just like a payment recorded on-line via a cash drawer, a payment interfaced from an external source (e.g., lock box or remittance processor) must reference a tender control and the tender controls, in turn, must reference a deposit control. The only real differences between these two types of payments are highlighted below:

- Whilst an operator must create and balance the tender and deposit controls for real-time payments, the system creates the tender and deposit controls associated with interfaced payments.
- It's impossible for an invalid account to be referenced on a payment recorded real-time. However, it is quite possible for an interfaced payment to reference an unknown account. If an invalid account is referenced on an interfaced payment (using no distribution rules), the system links it to the suspense service agreement referenced on the tender source control table.

> **NOT_SUPPORTED**
>
> Refer to Interfacing Payments From External Sources for more information.

Payment Financial Transaction Considerations

A payment segment exists for each service agreement that receives a portion of a payment. Linked to every frozen payment segment is a financial transaction. This financial transaction affects a service agreement's payoff balance and/or current balance. It also contains the journal details that debt cash and credit some other account.

The topics in this section provide information about the financial impact of a payment segment.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Financial Big Picture for more information about a payment's place in the financial big picture.

- Payment - Current Balance versus Payoff Balance
- The Source Of GL Accounts On A Payment Financial Transaction

Payment - Current Balance versus Payoff Balance

Warning

If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

A payment segment financial transaction almost always affects payoff balance and current balance by the same amount (think of it like this - when a customer pays, the amount they think they owe goes down by the amount they really owe). The only exception is a payment segment for a charitable contribution. These payment segments only affect current balance because the customer was never billed for the contribution in the first place.

NOT_SUPPORTED

Refer to Setting Up Payment Segment Types for more information about how payment segment type affects how a payment segment is produced and how its financial transaction is generated.

The Source Of GL Accounts On A Payment Financial Transaction

A payment segment's financial transaction also contains the double-sided accounting entry that defines how the payment segment affects the general ledger.

NOT_SUPPORTED

Refer to The Source Of GL Accounts On Financial Transactions for a description of where the system extracts the distribution codes used to construct the GL accounts.

A Payment May Affect More Than Just Customer Balances

Copyright © 2026, Oracle and/or its affiliates.

The topics in this section provide information about obscure things that may happen when a payment is distributed and frozen.

- Open Item Accounting and Match Events
- FT Freeze Repercussions

Open Item Accounting and Match Events

NOT_SUPPORTED

Refer to Payments and Match Events for more information about how payments can create match events for open-item accounts.

FT Freeze Repercussions

NOT_SUPPORTED

Refer to Obscure Things That Can Happen for more information about things that can happen when an FT is frozen (and FT's get frozen when a payment is frozen).

Automatic Payments

This section discusses how to set up and manage customers who pay their bills automatically (via direct debit or credit card debits)

- How To Set Up A Customer To Pay Automatically
- What Are Automatic Payments?
- How And When Are Automatic Payments Created?
- Automatic Payment Dates
- How To Implement Maximum Withdrawal Limits
- How Are Automatic Payments Cancelled?
- Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created
- Pay Plans and Automatic Payments
- Non-Billed Budgets and Automatic Payment
- Downloading Automatic Payments and Interfacing Them To The GL
- ACH Record Layouts
- SEPA Direct Debit Record Layouts

Copyright © 2026, Oracle and/or its affiliates.

- Third Party Auto Pay Record Layouts

How To Set Up A Customer To Pay Automatically

If a customer wants to pay automatically, transfer to Account - Auto Pay and define the source of the funds and the customer's account or credit card number.

What Are Automatic Payments?

An automatic payment is just like any other payment (refer to A Payment Event Has Payments And Tenders for more information about payments in general). However, automatic payments have one special trait - they cause funds to be transferred into your company's bank account. Refer to Downloading Automatic Payments and Interfacing Them To The GL for how this transference happens.

How And When Are Automatic Payments Created?

Automatic payments can be created in several ways:

- The system creates automatic payments for bills linked to accounts with an active auto pay option (note, the system uses the account's autopay option that's effective on the bill's due date). When the system does this is dependent on the value of the **Autopay Creation Option** on the installation record:
  - If you've set this option to Create At Bill Completion, the automatic payment is created when the account's bill is completed. The payment is distributed and frozen at the same time. This means that the account's balance is almost always zero (or in credit if they have a deposit). You will see the automatic payments in the payment list immediately after the bill is completed (rather than waiting until the automatic payment's extraction date).
  - If you've set this option to Create On Extract Date, the automatic payment is NOT created when the bill is completed. Rather,
- At bill completion time, the bill is stamped with the automatic payment's extract date and amount. The date is the automatic payment source's extraction date (refer to Automatic Payment Dates for more information on how this date is calculated).
- The automatic payment background process (APAYCRET) creates the automatic payment on the extract date stamped on the bill. If the supplied Auto Pay Creation algorithm (APAY-CREATE) is used and is configured to override the automatic payment amount based on the account’s current balance, the algorithm can either adjust the auto pay amount (if the balance is less than the original auto pay amount) or skip automatic payment creation (if the balance is zero or a credit). Refer to the algorithm type description for more details.
- The automatic payment is NOT distributed and frozen when the automatic payment is initially created. A separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to Automatic Payment Dates for more information on how this date is calculated). This means that the customer's balance increases when the bill is completed and is only reduced when the automatic payment is marked for interface to the general ledger.

Copyright © 2026, Oracle and/or its affiliates.

Note that it is possible for automatic payments to be distributed and frozen after being extracted and interfaced to a financial institution. Please refer to Downloading Automatic Payments and Interfacing Them To The GL and The Nightly Processes.

An algorithm plugged in on the Installation Record calculates the payment amount whether the automatic payment is created at bill completion time or on the extract date. Please refer to APAM-DFLT for more information about how the algorithm that is supplied with the base package calculates this amount.

> **Note**
>
> **With balance forward accounting, automatic payments are not just for new charges.** The base package algorithm includes prior balances when it creates a customer's first automatic payment. For example, if a customer has an existing balance of $100 and then signs up for automatic payment, their next bill will cause an automatic payment of $100 plus any new charges to be created (assuming the $100 remains unpaid at the time the next bill is completed). Refer to Open Item Versus Balance Forward Accounting for information about balance forward accounting.

- If a customer with a pay plan indicates that the payment method is via "autopay", a background process (PPAPAY) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the installation record is set to Create On Extract Date, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to Automatic Payment Dates for more information on how this date is calculated). Refer to The Big Picture Of Pay Plans for more information about pay plans.
- If a customer with an account that is set up for automatic payment has a Non-Billed Budget that is not excluded from automatic payment, a background process (NBBAPAY) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the installation record is set to Create On Extract Date, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to Automatic Payment Dates for more information on how this date is calculated). Refer to **What is a Non-Billed Budget** for more information about Non-Billed Budgets.
- A user can create an automatic payment by simply adding a payment tender with a tender type that indicates it is for automatic payment purposes. This would be a rather unusual thing to do, but you might do this if you want to immediately debit a customer's bank account after a large adjustment is added to the system (e.g., if they suddenly owe you a lot of money and you don't want to wait until the next bill to collect it). Automatic payments created by this method must be distributed and frozen before they can be extracted.

> **Note**
>
> **An algorithm is used to create automatic payments.** The logic used to create automatic payments is plugged in on the Installation Record. This algorithm is **not** called when a user manually creates an automatic payment (by adding a payment tender with a tender type that indicates that it is for automatic payment purposes) and when the automatic payment is for prenotification purposes.

Copyright © 2026, Oracle and/or its affiliates.

When an automatic payment is first created, it gets marked with a distribution date. The distribution date is the date on which the automatic payment's FT's GL details can be interfaced to the general ledger (via the standard GL interface). The distribution date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that is responsible for calculating the GL Distribution (Posting) date. On the GL distribution date, the automatic payment will be interfaced to the general ledger.

When a checking or savings account is used for the first-time on a self-service automatic payment enrollment, self-service one time payment, self-service wallet management or on a manual payment event (created using the Add One Time Payment user interface), a payment event, payment tender and auto pay staging records are created for prenotification purposes. Refer to Web Debit Account Validation for more details.

Automatic Payment Dates

As described in the previous section, an algorithm (that's plugged in on Auto Pay Route Types) controls the date on which the automatic payment is interfaced to your general ledger. We refer to this date as the GL distribution date.

This algorithm also populates the following dates:

- The payment date that is stored on the payment.
- The date on which the automatic payment is interfaced to the financial institution.

The algorithm that is supplied with the base package provides many parameters that allow you to dictate how these dates are calculated. Please refer to APAY-DTCALC for the details.

How To Implement Maximum Withdrawal Limits

In some locales, customers can define a "maximum withdrawal amount" to limit the amount of money that is automatically debited from their bank account. For example, a low-income customer may want to prevent direct debits of more than $50 from being applied to their checking account.

You define a customer's maximum withdrawal amount when you setup their automatic payment information on Account - Auto Pay.

The following points describe how the system implements maximum withdrawal limits:

- When a bill is completed for a customer who pays automatically the system calls the automatic payment creation algorithm that's plugged in on the Installation Record. The base-package autopay creation algorithm (APAY-CREATE) checks if the amount of the automatic payment exceeds the customer's maximum withdrawal amount. If so, the autopay creation algorithm calls the automatic payment over limit algorithm that's plugged in on the account's customer class. Algorithms of this type have the ability to reduce the amount of the autopay or to prevent the autopay from being created. Refer to APOL-RA for an example plug-in.

Copyright © 2026, Oracle and/or its affiliates.

- When a user manually creates an automatic payment (by adding a tender with a tender type that indicates that it is for automatic payment purposes), the system issues a warning message when the tender amount exceeds the account's maximum withdrawal amount.

> **Note**
>
> **Pay plans and Non-Billed Budgets.** Automatic payments that are created as a result of pay plans and Non-Billed Budgets are not subject to maximum withdrawal limits. This is because both of these options required customer approval and therefore the customer should be able to plan accordingly. Refer to How And When Are Automatic Payments Created for more information.

How Are Automatic Payments Cancelled?

There are two ways to cancel an automatic payment:

- The system will cancel an automatic payment behind-the-scenes if the related bill (if any) is reopened BEFORE the automatic payment is interfaced to the financial institution. When you recomplete the bill, the system will create a new automatic payment that reflects the new amount due (and the canceled automatic payment will net out the original automatic payment).
- An operator can cancel an automatic payment (refer to How To Cancel A Tender) at any time. You would do this if the automatic payment was rejected by the financial institution.

> **NOT_SUPPORTED**
>
> Refer to Credit / Correction Notes and Auto Pay for information about how a credit note may cancel an automatic payment.

Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created

The system creates a match event when a bill is completed for open-item customers that pay automatically (i.e., direct debit customers). The match event groups together the bill's new charges against the automatic payment's payment segments.

If the bill is subsequently re-opened, the match event will be cancelled when the automatic payment is cancelled.

> **NOT_SUPPORTED**
>
> Refer to Open Item Accounting for more information.

Copyright © 2026, Oracle and/or its affiliates.

Pay Plans and Automatic Payments

If a customer wants to pay their pay plan scheduled payments automatically, the account must be set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically). In addition, the payment method defined on the pay plan must indicate automatic payment is being used.

When this is done, a background process (PPAPAY) creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

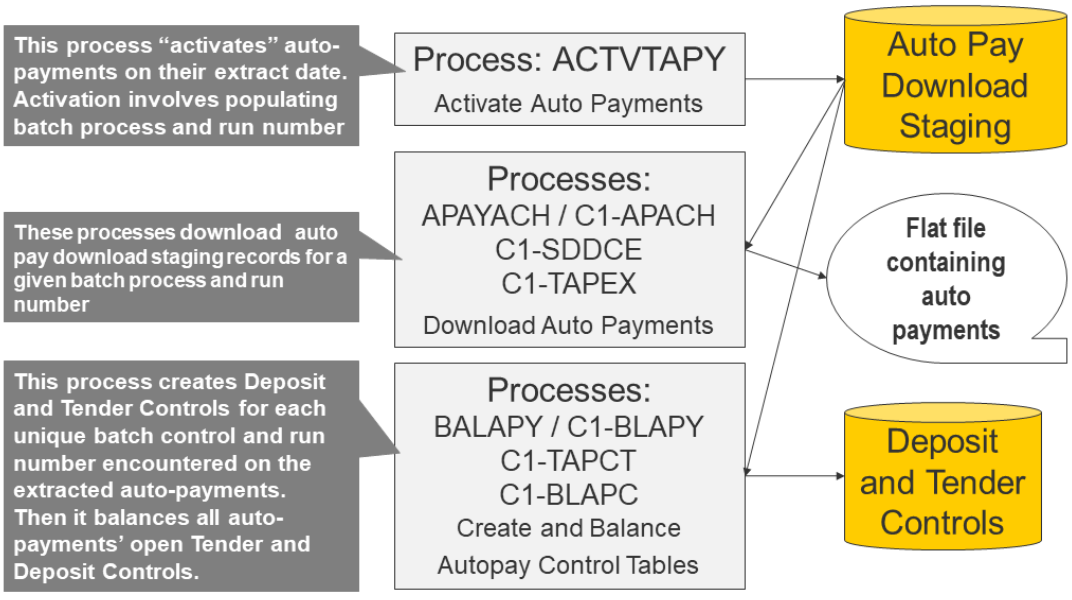
Non-Billed Budgets and Automatic Payment

If a customer wants to pay their Non-Billed Budget scheduled payments automatically, the account must be set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically). In addition, the Non-Billed Budget must indicate that automatic payment is being used.

When this is done, a background process (NBBAPAY) creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

Downloading Automatic Payments and Interfacing Them To The GL

The following diagram illustrates the background processes that interface automatic payment out of the system:



These processes are described in the following topics.

- ACTVTAPY - Activating Automatic Payments

Copyright © 2026, Oracle and/or its affiliates.

- APAYACH - Download Automatic Payments To The ACH (Automated Clearing House)
- C1-APACH - Download Automatic Payments To The ACH (Automated Clearing House)
- C1-SDDCE - SEPA Direct Debit Payment Extract
- C1-TAPEX - Third Party Auto Pay Extract
- BALAPY - Creating Automatic Payment Tender Controls
- C1-BLAPY - Create Autopay Tender Controls
- C1-BLAPC - Balance Autopay Control Tables

ACTVTAPY - Activating Automatic Payments

When an automatic payment is first created, it gets marked with an extract date. The extract date is the date the automatic payment will be downloaded to the respective financial institution. The extract date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that calculates this date.

On the extract date, the automatic payment is "activated". The automatic payment is activated is marked for download the next time its download process runs. An automatic payment's download process is defined on its auto-pay source's route type.

APAYACH - Download Automatic Payments To The ACH (Automated Clearing House)

This process reads all auto pay download staging records marked with a given batch control ID and run number, and creates the flat file that's passed to the ACH. Refer to ACH Record Layouts for the details of the record layouts.

The following outlines the differences between APAYACH and C1-APACH:

- APAYACH uses the Business Process Date as the draft date. C1-APACH populates the Effective Entry Date field on the Company Batch Header Record with a date (the actual draft date) as to when the payment should be withdrawn from the customer's account.
- APAYACH produces a file that has one File Header Record, a Company Batch Header Record per payment source (if multiple payment sources exist), and multiple detail lines. C1-APACH produces a file with multiple occurrences of the Company Batch Header Record differing on the Effective Entry Date field and the payment source (if multiple payment sources exist).

> **Note**
>
> **This process may be rerun.** You can reproduce the flat file at any time. Simply request this job and specify the run number associated with the historic run.

If you require a different flat file format, you must create additional versions of this program. Refer to Setting Up Automatic Payment Extracts for instructions describing how to add another process.

Copyright © 2026, Oracle and/or its affiliates.

C1-APACH - Download Automatic Payments To The ACH (Automated Clearing House)

This process is an alternative to using APAYACH. Similar to APAYACH, this process reads all auto pay download staging records marked with a given batch control ID and run number, and creates the flat file that's passed to the ACH. Refer to ACH Record Layouts for the details of the record layouts.

The following outlines the differences between APAYACH and C1-APACH:

- C1-APACH populates the Effective Entry Date field on the Company Batch Header Record with a date (the actual draft date) as to when the payment should be withdrawn from the customer's account. APAYACH uses the Business Process Date as the draft date.
- C1-APACH produces a file with multiple occurrences of the Company Batch Header Record differing on the Effective Entry Date field and payment source (if multiples payment sources exist). APAYACH produces a file that has one File Header Record, a Company Batch Header Record per payment source (if multiple payment sources exist), and multiple detail lines.

> **Note**
>
> **This process may be rerun.** You can reproduce the flat file at any time. Request this job and specify the run number associated with the historic run.

If you require a different flat file format, you must create additional versions of this program. Refer to Setting Up Automatic Payment Extracts for instructions describing how to add another process.

C1-SDDCE - SEPA Direct Debit Payment Extract

This process reads all auto pay download staging records marked with a given batch control ID and run number to create an XML extract file that's passed to the financial institution. Refer to SEPA Direct Debit Record Layouts for the details of the record layouts.

> **Note**
>
> **This process may be rerun.** You can reproduce the XML extract at any time. Simply request this job and specify the run number associated with the historic run.

C1-TAPEX - Third Party Auto Pay Extract

This process reads all auto pay download staging records marked with a given batch control ID and run number to create an extract file that is sent to the third party payment processing system.

This is a plugin-driven batch process that supports any of three file layouts (comma-separated values (CSV), fixed position and XML) specified in an input batch parameter.

The C1–TAPEX batch control is delivered with two algorithms plugged in:

Copyright © 2026, Oracle and/or its affiliates.

- A Select Records algorithm is responsible for selecting the auto pay staging records to download.
- A Process Record algorithm is responsible for formatting and writing each extract record into the output flat file.

Your implementation can plug in your own Select Records and Process Record algorithms if your business rules for this extract differ.

This batch process is meant to be run after the Activate auto-pay (ACTVTAPY) runs and before Third Party Auto Pay Tender Controls (C1–TAPCT) runs.

Refer to Third Party Auto Pay Record Layouts for the details of the record layouts.

> **Note**
>
> **This process may be rerun.** You can reproduce the XML extract at any time. Simply request this job and specify the run number associated with the historic run.

BALAPY - Creating Automatic Payment Tender Controls

This process creates a new tender control (with an associated deposit control) for each unique batch control ID and run number encountered in the extracted automatic payments (where its payment tender is not yet linked to a tender control). The payment tender of each of these automatic payments is then linked to the corresponding tender control. This process also balances the open tender control records afterwards.

> **Note**
>
> A separate batch process (C1-BLAPC) must be scheduled after BALAPY to balance open tender and deposit control records.

C1-BLAPY - Create Autopay Tender Controls

This process creates a new tender control (with an associated deposit control) for each unique batch control ID and run number encountered in the extracted automatic payments (where its payment tender is not yet linked to a tender control).

This is a plugin driven version of BALAPY. The difference is that it excludes payments that were processed by a third party payment processing system. These are handled by other tender controls processes - C1–TAPCT (for third party auto pays) and C1–TOPCT (for third party one time payments)

The C1–BLAPY batch control is delivered with two algorithms plugged in:

- A Select Records algorithm is responsible for determining the auto pay staging records to process.
- A Process Record algorithm is responsible for linking the auto pay record’s payment tender to the tender control.

Copyright © 2026, Oracle and/or its affiliates.

Your implementation can plug in your own Select Records and Process Record algorithms if your business rules for this extract differ.

> **Note**
>
> A separate batch process (C1-BLAPC) must be scheduled after C1–BLAPY to balance open tender and deposit control records.

C1-BLAPC - Balance Autopay Control Tables

This batch process is responsible for balancing tender and deposit control tables in the autopay extract process. This batch process is executed after BALAPY completes.

ACH Record Layouts

The topics in this section describe the layout of the records created by APAYACH - Download Automatic Payments To The ACH (automated clearing house) and C1-APACH - Download Automatic Payments To The ACH (automated clearing house).

> **Note**
>
> APAYACH and C1-APACH share most of the same fields. The exceptions are documented in the following tables.

- File Header Record
- Company Batch Header Record
- Entry Detail Record
- Company Batch Control Record
- File Control Record

File Header Record

The ACH extract flat file must have one record of this type and it must be the first logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"1"
PRIORITY-CD	A2	"01"
RESERVED-01	A1	Spaces
IMMEDIATE-DESTINATION	A9	The auto pay route type's tender source's bank account's DFI ID. **Note.** Tender source may be overridden on CIS Division

Copyright © 2026, Oracle and/or its affiliates.

COMPANY-ID	A10	Auto Pay Route Type / Company Id **Note.** May be overridden on CIS Division
FILE-CRE-DT	A6	YYMMDD. Current date
FILE-CRE-TM	A4	HHMM. Current time
FILE-ID-MODIFIER	A1	"A"
RECORD-SIZE-CONST	N3	"094" The "FILLER" at the end of all but the Entry Detail Record serves to bring the total length of each record up to this constant.
BLOCKING-FACTOR	N2	"10"
FORMAT-CD	A1	"1"
DEST-FIN-INST-NAME	A23	The auto pay route type's tender source's bank description. **Note.** Tender source may be overridden on CIS Division
COMPANY-NAME (ORIG-FIN-INST-NAME)	A23	Auto Pay Route Type / Company Name **Note.** May be overridden on CIS Division
REFERENCE-CD	A8	Spaces

Company Batch Header Record

The ACH extract flat file can contain multiple company batch headers - one header per payment source as distinguished by STD-ENTRY-CLASS. This record must be the second logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"5"
SERVICE-CLASS-CD	A3	"200"
COMPANY-NAME	A16	Auto Pay Route Type / Company Name **Note.** May be overridden on CIS Division
COMPANY-DISCRETIONARY	A20	Spaces
COMPANY-ID	A10	Auto Pay Route Type / Company Id **Note.** May be overridden on CIS Division

Copyright © 2026, Oracle and/or its affiliates.

STD-ENTRY-CLASS	A3	"PPD", unless the payment is a one-time payment. For one-time payments, the STD-ENTRY-CLASS can be populated using a One Time Payment Source characteristic on the associated tender. Base logic for self-service one-time payment processing adds a One Time Payment Source characteristic with a value of 'WEB' to the tender. Your implementation may define other characteristic values for any of your additional payment sources/triggers. Refer to the help for the One Time Payment Sources section of the Self-Service Integration Master Configuration for more details.
CO-ENTRY-DESCR	A10	For standard Auto Pay Route Type: Company Entity Description **Note.** May be overridden on CIS Division
COMPANY-DESCR-DT	A6	Business process date
EFF-ENTRY-DT	A6	For APAYACH, the EFF-ENTRY-DT is the business process date. For C1-APACH, the EFF-ENTRY-DT is calculated from the ACH staging / Scheduled Extract Date + Number of Days batch parameter if specified greater than 0. A company batch is created for each value.
RESERVED-01	A3	Spaces
ORIGINATOR-STAT-CD	A1	"1"
ORIGIN-DFI-ID	A8	The auto pay route type's tender source's bank account's DFI ID. **Note.** Tender source may be overridden on CIS Division
BATCH-NBR	N7	The batch number of this batch within the file.

Entry Detail Record

The ACH extract flat file must have one record of this type for every direct debit record.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"6"

Copyright © 2026, Oracle and/or its affiliates.

TRANSACTION-CD	A2	Pay Tender's Tender Type / External Type Flag
TRANSIT-RTG-NBR	A9	Auto pay source / External Source Id
DFI-ACCT-NBR	A17	Auto pay staging / External Account Id
AMOUNT	N8.2	Pay Tender / Tender Amount
INDIVIDUAL-ID	A15	Pay Tender / Payor Account Id
INDIVIDUAL-NAME	A22	Auto pay staging / Entity Name
DISCRETIONARY-DATA	A2	Spaces
ADDENDA-REC-IND	A1	"0"
DFI-ID	A8	The auto pay route type's tender source's bank account's DFI ID. **Note.** Tender source may be overridden on CIS Division
ENTRY-DETAIL-CNT	N7	Ascending counter starting at 1 and incremented by 1 for each entry detail record

> **Note**
>
> **External Account ID.** The EXT_ACCT_ID field supports up to 50 characters. If the value entered in the field is longer than that supported by the record layout (17 characters), the value will be truncated.

Company Batch Control Record

The ACH extract flat file must have one record of this type and it must follow the Entry Detail records.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"8"
SERVICE-CLASS-CD	A3	"200"
ENTRY-ADDENDA-CNT	N6	The total number of entry detail records in this batch.
ENTRY-HASH	N10	The product of the first 8 digits of the external source id of the autopay source, multiplied by the number of entry detail records in the batch.
TOTAL-DR-DOLLAR-AMT	N10.2	Total tender amounts of the entry detail records.
TOTAL-CR-DOLLAR-AMT	N10.2	Zero.

Copyright © 2026, Oracle and/or its affiliates.

COMPANY-ID	A10	Auto Pay Route Type / Company Id **Note.** May be overridden on CIS Division
RESERVED-01	A19	Spaces
RESERVED-02	A6	Spaces
ORIGIN-DFI-ID	A8	The auto pay route type's tender source's bank account's DFI ID. **Note.** Tender source may be overridden on CIS Division
BATCH-NBR	N7	The batch number of this batch within the file.

File Control Record

The ACH extract flat file must have one record of this type and it must be the last logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"9"
BATCH-CNT	N6	The number of batches in this file.
BLOCK-CNT	N6	Calculation of the total number of records in the file / 10.0 + 0.9
ENTRY-ADDENDA-CNT	N8	The total number of entry detail records in this file.
ENTRY-HASH	N10	The sum of the entry hash values on all the batch control records in this file.
TOTAL-DR-DOLLAR-AMT	N10.2	Sum of the total debit entry dollar amounts of all the batches in this file.
TOTAL-CR-DOLLAR-AMT	N10.2	Sum of the total credit entry dollar amounts of all the batches in this file.
RESERVED-01	A39	Spaces

SEPA Direct Debit Record Layouts

- Group Header
- Payment Information
- Direct Debit Transaction Information

Copyright © 2026, Oracle and/or its affiliates.

Group Header

A SEPA XML file must include a group header. The XML path for the group header is **CstmrDrctDbtInitn/GrpHdr** and the elements under the group header are as follows:

Description	Element Name	Source/Value
Message Identification	MsgId	Retrieved from the direct debit task type
Creation Date/Time	CreDtTm	Batch business date in the format YYYY-MM-DDThh:mm:ss
Number of Transactions	NbOfTxs	Calculated at extract time
Initiating Party Name	InitgPty/Nm	Retrieved from the direct debit task type

Payment Information

Payment Information contains elements related to the credit side of the transaction, such as creditor and payment type. The XML path for payment information is **CstmrDrctDbtInitn/PmtInf** and the elements are as follows:

Description	Element Name	Source/Value
Payment Information Identification	PmtInfId	Constructed at extract time by concatenating the following: Batch Code, Batch Run Number, Batch Business Date and a Sequence Number assigned to each Payment
Payment Method	PmtTpInf/PmtMtd	Defaulted to DD (Direct Debit)
Service Level Code	SvcLvl/Cd	Retrieved from the direct debit task type; defaulted to 'SEPA'
Local Instrument Code	PmtTpInf/LclInstrm/Cd	Retrieved from the direct debit task type; defaulted to 'CORE'
Sequence Type	PmtTpInf/SeqTp	Direct debit mandate task's Payment Type
Requested Collection Date	ReqdColltnDt	Payment event's pay date
Creditor Name	Cdtr/Nm	Direct debit mandate task's Creditor Name
Creditor Account Identification IBAN	CdtrAcct/Id/IBAN	CI_BANK_ACCOUNT.ACCOUNT_NBR
Creditor Agent Financial Institution Identification BIC	CdtrAgt/FinInstnId/BIC	CI_BANK_ACCOUNT.DFI_ID_NUM
Creditor Account Number	CdtrSchmeId/Id/PrvtId/Othr/Id	Direct debit task type's Creditor Identification
Creditor Party	CdtrSchmeId/Id/PrvtId/Othr/SchmeNm/Prtry	Defaulted to 'SEPA'

Copyright © 2026, Oracle and/or its affiliates.

Direct Debit Transaction Information

Direct Debit Transaction Information contains elements related to the debit side of the transaction, such as debtor and remittance information rules. The XML path for direct debit transaction information is **CstmrDrctDbtInitn/PmtInf/DrctDbtTxInf** and the elements are as follows:

Description	Element Name	Source/Value
Payment End to End Identification	PmtId/EndToEndId	CI_PAY_TNDR.PAY_TENDER_ID
Payment Instructed Amount	InstdAmt	CI_PAY_TNDR.TENDER_AMT
Mandate Related Information – Mandate Identification	DrctDbtTx/MndtRltdInf/MndtId	Direct debit mandate task's External Mandate Reference
Mandate Related Information – Date of Signature	MndtRltdInf/DtOfSgntr	Direct debit mandate task's Effective Date
Mandate Related Information – Electronic Signature	MndtRltdInf/ElctrncSgntr	Direct debit mandate task's Signature
Debtor Agent Financial Institution Identification BIC	DbtrAgt/FinInstnId/BIC	CI_APAY_SRC.EXT_SOURCE_ID
Debtor Name	Dbtr/Nm	CI_APAY_CLR_STG.ENTITY_NAME
Debtor Account Identification IBAN	DbtrAcct/Id/IBAN	CI_APAY_CLR_STG.EXT_ACCT_ID

Third Party Auto Pay Record Layouts

The Third Party Auto Pay Extract file is a comma-separated (CSV) format file that contains one or more automatic payment detail records and a footer record.

Automatic Payment Detail Record

There must be at least one record of this type in the file.

Field Name	Format	Source/Value/Description
Account ID	A10	Account Auto Pay Option / Account ID
Auto Pay Amount	N13.2	Pay Tender linked to the Auto Pay Clearing Staging / Tender Amount
Due Date	A10	Due Date of the Bill ID that is linked to the Auto Pay Clearing Staging
Maximum Withdrawal Amount	N13.2	Account Auto Pay Option of the related Account ID / Maximum Withdrawal amount field
Auto Pay Staging ID	A12	Auto Pay Clearing Staging / Auto Pay Staging ID

Summary/Footer Record

There must be one record of this type at the end of the file.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Total Number of Records	N13.2	Total number of auto pay detail records in the file
Total Amount	N13.2	Sum of auto pay amounts in the file

- Payment Information
- Footer Information

Payment Information

The Third Party Auto Pay Extract file must have one record of this type for every direct debit record. The **C1-TAPEX-PR** Process Record algorithm uses the **C1-ThirdPartyAutoPayExtractRec** data area, which contains the following:

Description	Element Name	Source/Value
Payor Account ID	payorAccountId	Payor account ID on the auto pay tender
Auto Pay Amount	autopayAmount	Tender amount on the auto pay tender
Due Date	billDueDate	Due date of the Bill ID linked to the auto pay
Maximum Withdrawal Amount	autopayMaxWithdrawalAmount	From the account's auto pay option
Auto Pay Clearing ID	autopayClearingId	From the auto pay staging record

Footer Information

The Third Party Auto Pay Extract file must have one and only one record of this type. The **C1-TAPEX-PR** Process Record algorithm uses the **C1-ThirdPartyBatchRunVariables** data area, which contains the following:

Description	Element Name	Source/Value
Record Count	recordCount	Total number of payments in the file
Total Amount	totalAmount	Total amount of payments in the file

Third Party Payment Processing

A third party payment processing system is a service provider that can handle all aspects of electronic payments for a utility. When this service provider is integrated with *Oracle Utilities Customer To Meter*, every payment made through channels such as web self-service or integrated voice response (IVR) is accepted and processed within the service provider's fully hosted payment acceptance environment. Payment data (e.g., credit card number, bank account number, etc.) is stored and processed outside of the utility's network. The base product stores minimal information in Account Auto Pay and Person Payment Options records that is required for auto pay processing and one-time payment processing, respectively.

Copyright © 2026, Oracle and/or its affiliates.

The following sections describe the functions that are supported when this product is integrated with a third party payment processing system.

- One Time Payments
- Account Verification
- Automatic Payment Setup
- Interfacing Automatic Payments To A Third Party
- Highlighting Scheduled Payments
- Automatic Payment Confirmations And Cancellations
- Third Party Payment Tender Controls Processing

One Time Payments

A one time payment is a one-off, non-recurring payment option where the third party payment processing is authorized to collect a payment from a utility customer's credit card or bank account.

One time payments can be immediate or scheduled:

- Immediate one-time payments are processed on the same date that the payment is created/captured in the third party payment processing system.
- Scheduled one-time payments are processed at a future date that the customer designates.

The base product provides an inbound web service **C1-ThirdPartyOneTimePayNotification** that receives one time payment information from the third party payment processor, and, based on the required action, creates/updates the payment in the system. This web service supports the following functions:

- Creating an immediate or scheduled one time payment.
- Updating a scheduled one time payment
- Processing a scheduled one time payment
- Canceling a processed one time payment

The following sections describe these actions.

- Creating One Time Payments
- Updating Scheduled One Time Payments
- Processing Scheduled One Time Payments
- Canceling One Time Payments

Creating One Time Payments

When the third party payment processing system processes a new one time payment, it will call the **C1-ThirdPartyOneTimePayNotification** inbound web service with an Add action. The payment details include the payment date/time, payment amount and basic details about the payment method (*i.e.*, bank account or credit card).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Because bank account numbers and credit card numbers are captured outside of the system, the full account numbers are not sent. In most cases, only a partial value is sent, e.g. last four digits.

The inbound web service processes the payment as follows:

- If the Payment Date is the current date, it is processed as an immediate payment.
  - The system creates a Third Party One Time Payment Task, which in turn creates a frozen payment.
  - The payment date/time sent in by the third party payment processor is stored as a characteristic on the payment tender using the Processing Date/Time Characteristic Type in the Third Party One Time Payment Processing Integration master configuration. This is to capture the actual date/time when the payment was processed. This information is used in tender control processing.

    > **Note**
    >
    > The payment date/time is validated for XSD format. Any subsequent manual updates to this date/time will need to be in the correct XSD format. An adhoc characteristic validation algorithm type C1–VALDATE exists for this purpose.

  - A reference to the Third Party One Time Payment Task's Service Task ID is stored as a characteristic on the payment tender using the Processing Reference Characteristic Type in the Third Party One Time Payment Processing Integration master configuration. This helps identify that the payment event/tender was created from third party payment processing.
  - The Payment Event ID is sent back to the third party payment processing system so that it has a way to identify the payment should there be a need to cancel the payment in the future.
- If the Payment Date is a future date, it is processed as a scheduled payment.
  - The system creates a pending Third Party One Time Payment Task. No further action is taken until the third party payment processing system sends another notification that the scheduled payment has been processed.
  - The Service Task ID is sent back to the third party payment processing system so that it has a way to identify the service task on subsequent requests to either update or process the pending one time payment.

> **Note**
>
> To prevent the creation of duplicate payments when the web service times out and the third party payment processing retries the web service call:
>
> - The system checks if a frozen payment was already created from a previous web service call.
> - The account is locked during payment event creation.

Copyright © 2026, Oracle and/or its affiliates.

For more information about the lifecycle of the Third Party One Time Payment Notification service task, refer to the **C1-ThirdPartyOneTimePayTask** business object in the system.

For more information about master configuration settings, refer to the **Third Party Payment Processing Integration** master configuration in the system.

Updating Scheduled One Time Payments

If the third party payment processing system receives a request from the customer to update an existing one time payment, it will call the **C1-ThirdPartyOneTimePayNotification** inbound web service with an **Update** action, passing in the pending one time payment's Service Task ID and the updated payment details.

> **Note**
>
> Because bank account numbers and credit card numbers are captured outside of the system, the full account numbers are not sent. In most cases, only a partial value is sent, e.g. last four digits..

The inbound web service updates the existing Third Party One Time Payment Task with the new payment details sent in. The Third Party One Time Payment Task remains in the Pending state.

For more information about the lifecycle of the Third Party One Time Payment Notification service task, refer to the **C1-ThirdPartyOneTimePayTask** business object in the system.

Processing Scheduled One Time Payments

When it is time to process a scheduled one time payment, the third party payment processing will process the payment as it does an immediate one time payment. However, in this case, it will call the **C1-ThirdPartyOneTimePayNotification** inbound web service with a **Process** action, passing in the pending one time payment's Service Task ID and the payment details.

> **Note**
>
> Because bank account numbers and credit card numbers are captured outside of the system, the full account numbers are not sent. In most cases, only a partial value is sent, e.g. last four digits.

The inbound web service processes the one time payment as follows:

- The system transitions the existing Third Party One Time Payment Task so that it proceeds to create a frozen payment event.
- The payment date/time sent in by the third party payment processor is stored as a characteristic on the payment tender using the Processing Date/Time Characteristic Type in the Third Party One Time Payment Processing Integration master configuration. This is to capture the actual date/time when the payment was processed. This information is used in tender control processing.
- A reference to the Third Party One Time Payment Task's Service Task ID is stored as a characteristic on the payment tender using the Processing Reference Characteristic Type in the Third Party One Time Payment Processing Integration master configuration. This

Copyright © 2026, Oracle and/or its affiliates.

helps identify that the payment event/tender was created from third party payment processing.

- The Payment Event ID is sent back to the third party payment processing system so that it has a way to identify the payment should there be a need to cancel the payment in the future.

> **Note**
>
> To prevent the creation of duplicate payments when the web service times out and the third party payment processing retries the web service call:
>
> - The system checks if a frozen payment was already created from a previous web service call.
> - The account is locked during payment event creation.

For more information about the lifecycle of the Third Party One Time Payment Notification service task, refer to the **C1-ThirdPartyOneTimePayTask** business object in the system.

For more information about master configuration settings, refer to the **Third Party Payment Processing Integration** master configuration in the system.

Canceling One Time Payments

Scheduled or processed one time payments could get canceled for specific reasons. For instance, a processed payment could get canceled if a customer disputed a credit card charge or if the bank account has insufficient funds. When there is a need to cancel an existing scheduled or processed one time payment, the third party payment processor will cancel the one time payment in their system and call the **C1-ThirdPartyOneTimePayNotification** inbound web service with a **Cancel** action, passing in either a pending one time payment's Service Task ID (if canceling a scheduled payment) or a processed one time payment's Payment Event ID (if canceling a processed payment).

The inbound web service processes the one time payment cancellation as follows:

- For a scheduled on time payment, the existing Third Party One Time Payment Task is simply transitioned to the Discarded state.
- For a processed payment, the cancellation is processed as follows:
  - Since the original Third Party One Time Payment Task is already in the Completed state, a new Third Party One Time Payment Task is created to process the cancellation. Logic on the Process Payment state will cancel the existing Payment Tender using the Payment Event ID that is passed in from the web service.
  - The Tender Cancel Reason is taken from the **Third Party Payment Processing Integration** master configuration.
  - A reference to the cancellation Service Task ID is stored as a characteristic on the payment tender using the Cancellation Reference Characteristic Type that is defined in master configuration.
  - If master configuration is also set up with a Payment Auto-Cancel To Do Type, a To Do Entry is created.

For more information about the lifecycle of the Third Party One Time Payment Notification service task, refer to the **C1-ThirdPartyOneTimePayTask** business object in the system.

Copyright © 2026, Oracle and/or its affiliates.

For more information about master configuration settings, refer to the **Third Party Payment Processing Integration** master configuration in the system.

Account Verification

A third party payment processing system can perform account verification prior to processing one time payments and automatic payment enrollments. This is to verify that the account exists, as well as to get specific information about the account such as the account balance, due date, past due amount, and existing auto pay information.

The base product provides an inbound web service **C1-ThirdPartyVerifyAccount** that receives an Account ID from the third party payment processor and creates a Third Party Account Verification Task that is responsible for verifying the account. Base-supplied logic on this service task checks that the account exists and returns the account's balance, due date, past due amount (if any) and existing auto pay information (if any). Refer to the **C1-ThirdPartyAcctVerifTask** business object for more details.

Automatic Payment Setup

When the system is integrated with a third party payment processing system, customers who opt for automatic bill payment enrol through the third party payment processing system, where payment information that includes credit card or bank account number are captured and stored.

The third party payment processing system sends a notification when it processes new or updated account auto pay information.

The base product provides an inbound web service **C1-ThirdPartyAutoPaySetup** that receives auto pay setup information and creates a Third Party Auto Pay Setup Task, which is responsible for creating or updating the account's auto pay information. Base-supplied logic on this service task adds an account auto pay option, updates an existing account auto pay option's details or cancels an existing account auto pay option by setting its end date to the designated date. Refer to the **C1-ThirdPartyAutoPayTask** business object for more details.

Interfacing Automatic Payments To A Third Party

For customers who are enrolled to automatically pay bills through a third party payment processing system, existing auto pay processing that occurs during bill completion applies. Refer to How And When Are Automatic Payments Created? for more details.

The difference with automatic payment processing via third party payment processing system is that the auto pay staging records are not sent directly to the financial institution. Since the routing is different, a third party payment processing system is associated with a distinct Auto Pay Source and Auto Pay Route Type. Refer to Define Auto Pay Sources and Auto Pay Route Types for more details on this configuration.

Third party automatic payment transactions are extracted using the Third Party Auto Pay Extract (C1–TAPEX) batch process. Refer to Downloading Automatic Payments and Interfacing Them To The GL for details on how this batch process fits into the auto pay batch process flows.

Copyright © 2026, Oracle and/or its affiliates.

Highlighting Scheduled Payments

The Scheduled Payments Log zone in Control Central - Account Information displays the following:

- Pending third party one time payments
- Pending self-service one time payments
- Pending bill-related automatic payment records
- One time payments in error that have not been resolved, i.e. not succeeded by a Pending or Completed one time payment

Refer to the zone help for more information.

Automatic Payment Confirmations And Cancellations

A third party payment processing system can notify the system about whether or not the auto pay transactions sent to it for processing were processed successfully.

The base product provides two batch processes, one each for auto pay confirmation and auto pay cancellation. Refer to the following sections for more details.

- C1-TAPDF - Third Party Auto Pay Confirmations
- C1-TAPCN - Third Party Auto Pay Cancellations

C1-TAPDF - Third Party Auto Pay Confirmations

This batch process receives a comma-delimited (CSV) file containing information about auto pay transactions that were successfully processed at the third party payment processing system. Each record contains the following:

- Account ID
- Auto Pay Clearing ID (i.e. PK of the auto pay staging record)
- Payment Date/Time

A Process Record algorithm **C1-TAPDF-PR** is plugged in on this batch control. For each record in the file, it distributes and freezes the payment associated with the auto pay staging record. It also updates the payment event date if it differs from the payment date sent with the confirmation. The payment date/time on the confirmation is stored as a characteristic on the auto pay tender.

C1-TAPCN - Third Party Auto Pay Cancellations

This batch process receives a comma-delimited (CSV) file containing information about auto pay transactions that were not processed successfully at the third party payment processing system. Each record contains the following:

- Account ID
- Auto Pay Clearing ID (i.e. PK of the auto pay staging record)

A Process Record algorithm **C1-TAPCN-PR** is plugged in on this batch control. For each record in the file, it cancels the tender associated with the auto pay staging record, using the

Copyright © 2026, Oracle and/or its affiliates.

Tender Cancel Reason that is defined in the Third Party Payment Processing Integration master configuration.

Third Party Payment Tender Controls Processing

A third party payment processing system handles two types of payments: one time payments and recurring automatic bill payments.

The third party payment processing system sets the daily cutoff time for payment processing. For instance, a third party payment processor could set the default daily processing cutoff time to 8pm Eastern Time. But each implementation can override this processing cutoff time. The third party's processing cutoff time is defined in the Third Party Payment Processing Integration master configuration.

For recurring bill auto pays, the system will create the payment event, payment (non-frozen), tender and auto pay staging record. The Tender Control ID is not assigned until a batch process runs and creates the tender and deposit controls for the extracted auto pay transactions.

For one time payments, the tender and deposit control processing will be handled in the same way as the recurring auto pays. The one time payment's tender will not be linked to a tender control at one time payment tender creation time. Instead, the tender control will be created in batch.

The base product provides two batch processes for Third Party Auto Pay Tender Controls (C1–TAPCT) and Third Party One Time Payment Tender Controls (C1–TOPCT). For a third party one time payment, a pre-processing batch process C1–TOPPP is also provided. It takes the place of the auto pay extract process in a typical auto pay batch flow. This is because third party one time payment auto pays are used purely for tender controls processing and are not sent out by the system.

The following sections describe these batch processes.

- C1-TAPCT - Third Party Auto Pay Tender Controls
- C1-TOPPP - Third Party One Time Payment Controls Pre-Processing
- C1-TOPCT - Third Party One Time Payment Tender Controls

C1-TAPCT - Third Party Auto Pay Tender Controls

This process creates a tender control (with an associated deposit control) for each batch control and run number encountered for third party automatic payments that are not yet linked to a tender control.

This is a plugin-driven batch process. The C1–TAPCT batch control is delivered with two algorithms plugged in:

- A Select Records algorithm is responsible for selecting the auto pay staging records to process.
- A Process Record algorithm is responsible for linking each auto pay staging record's payment tender to the corresponding tender control.

Your implementation can plug in your own Select Records and Process Record algorithms if your business rules for this extract differ.

This batch process is meant to be run after the Third Party Auto Pay Extract (C1–TAPEX) runs and before Balance Autopay Control Tables (C1–BLAPC) runs.

Copyright © 2026, Oracle and/or its affiliates.

C1-TOPPP - Third Party One Time Payment Controls Pre-Processing

This process reads all auto pay download staging records marked with a given batch control ID and run number, to verify that the associated one time payment was processed within the current processing cutoff time. Any one time payments that were processed after the cutoff will be updated with the next batch run number.

This is a plugin-driven batch process.

The C1–TOPPP batch control is delivered with two algorithms plugged in:

- A Select Records algorithm is responsible for selecting the auto pay staging records to retrieve, based on the input batch code and batch run number..
- A Process Record algorithm is responsible for verifying that each auto pay staging record selected was processed within the current cutoff time.

Your implementation can plug in your own Select Records and Process Record algorithms if your business rules for this extract differ.

This batch process is meant to be run after the Activate auto-pay (ACTVTAPY) runs and before Third Party One Time Payment Tender Controls (C1–TOPCT) runs. This batch control should be specified on your configured auto pay route type for third party one time payments.

C1-TOPCT - Third Party One Time Payment Tender Controls

This process creates a tender control (with an associated deposit control) for each batch control and run number encountered for third party one time payments that are not yet linked to a tender control.

This is a plugin-driven batch process. The C1–TOPCT batch control is delivered with two algorithms plugged in:

- A Select Records algorithm is responsible for selecting the auto pay staging records to process.
- A Process Record algorithm is responsible for linking each auto pay staging record's payment tender to the corresponding tender control.

Your implementation can plug in your own Select Records and Process Record algorithms if your business rules differ.

This batch process is meant to be run after the Third Party Auto Pay Extract (C1–TOPPP) runs and before Balance Autopay Control Tables (C1–BLAPC) runs.

Issuing A Payment Advice Instead Of Creating An Automatic Payment

If the system is configured to send the customer a payment advice (instead of initiating an electronic funds transfer) when a bill is completed, the automatic payment records - i.e. payment event, payment, tender and auto pay clearinghouse staging - are not created. Refer to Payment Advices for more information.

- How To Set Up A Customer To Receive Payment Advices
- Payment Advice Option Is For Bill-Related Automatic Payments Only

Copyright © 2026, Oracle and/or its affiliates.

How To Set Up A Customer To Receive Payment Advices

Use Account - Auto Pay to capture the customer's bank details and indicate an auto pay method of Payment Advice.

Payment Advice Option Is For Bill-Related Automatic Payments Only

Payment advices can be printed for auto pays that result from completed bills.

An auto pay for a pay plan scheduled payment will not be created if the account's effective auto pay option is set to Payment Advice. The Pay Plan Auto Pay (PPAPAY) batch process will log an error in this case.

An auto pay for a Non-Billed Budget scheduled payment will not be created if the account's effective auto pay option is set to Payment Advice. The Non-Billed Budget Pay Plan Auto Pay (NBBAPAY) batch process will log an error in this case.

Manually created automatic payments (i.e. auto pays created via payment event UI) are always processed as direct debit.

Maintaining Payment Events

A payment event is used to record when moneys are remitted and how the moneys are allocated amongst accounts. The topics in this section describe how to maintain payment events.

> **Note**
>
> **The system creates most payment events behind-the-scenes.** Most payment events are created by the system when it uploads payments and when it creates automatic payments. You should only have to access the payment event transaction if you need to correct a payment event or add a payment event real-time. For information about how the system creates payment events, refer to The Big Picture of Payments.

- Payment Lifecycles
- Payment Event - Add Dialog
- Add One-Time Payment
- Payment Event - Main Information
- Payment Event - Tenders
- Payment Event - Characteristics
- Exception
- Payment Event Action Codes

Copyright © 2026, Oracle and/or its affiliates.

Payment Lifecycles

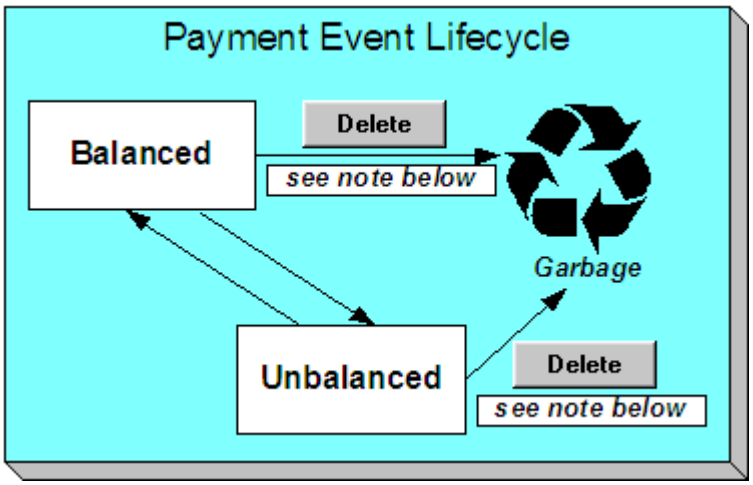
The topics in this section describe the lifecycle of the various payment objects.

- Payment Event Lifecycle
- Tender Lifecycle
- Payment Lifecycle

Payment Event Lifecycle

The following diagram shows the possible lifecycle of a payment event.

> **Warning**
>
> This diagram only makes sense in the context of the page used to maintain payment events. Refer to Payment Event - Main Information for the details.



The system, by default, distributes the sum of a payment event's tenders to the account that remits the tenders. After distribution the sum of the tenders equals the sum of the payments (remember, the term *payment* is used to refer to an allocation of some/all of a payment event's tenders to an account's debt) when the event is first created. We refer to such an event as being Balanced.

It is possible via any of the methods described in Unbalanced Payment Events to make a balanced payment event Unbalanced.

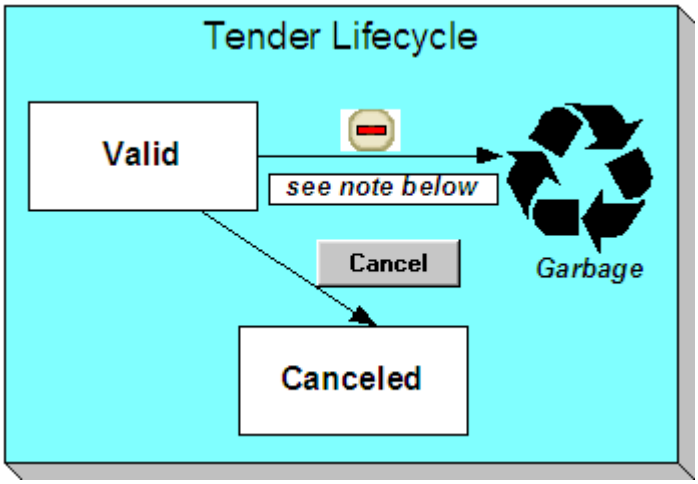
Click **Delete** to physically remove a balanced or unbalanced payment event from the database. You may not delete a payment event if: a) there are frozen or canceled payments linked to the event, or b) if there are canceled tenders linked to the event, or c) if a tender linked to the event is part of a balanced tender control. When the payment event is deleted, the system also deletes its tenders, payments, and payment segments.

Copyright © 2026, Oracle and/or its affiliates.

Tender Lifecycle

The following diagram shows the possible lifecycle of a payment event.

> **Warning**
>
> This diagram only makes sense in the context of the page used to maintain tenders. Refer to Payment Event - Tenders for the details.



A tender is initially saved in the Valid state. It is possible via any of the methods described in Unbalanced Payment Events to make a balanced payment event Unbalanced.

If a tender is invalid, click **Cancel** to cancel the tender AND ALL PAYMENTS LINKED TO THE EVENT.

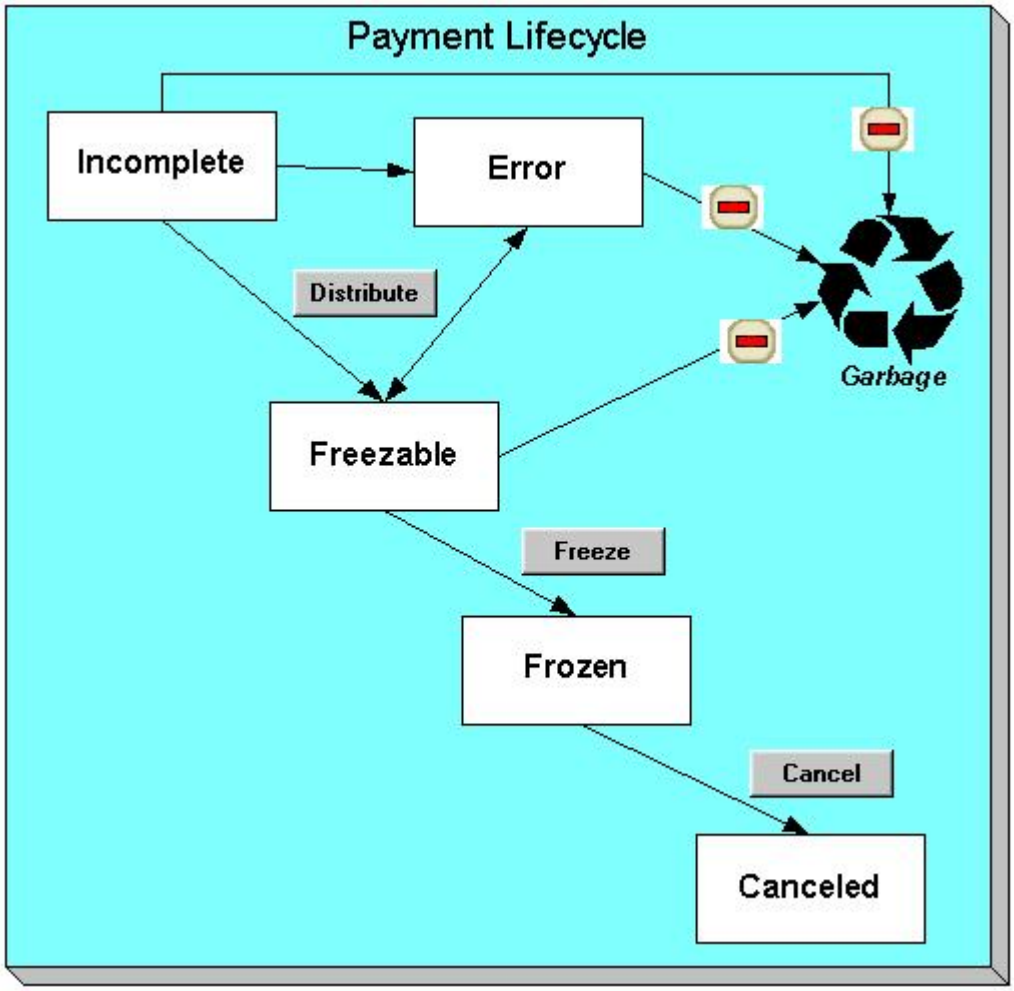
Click the delete button to physically remove a tender from the database. You may not delete a tender if it is part of a balanced tender control.

Payment Lifecycle

The following diagram shows the possible lifecycle of a payment.

> **Warning**
>
> This diagram only makes sense in the context of the page used to maintain a payment. Refer to Payment - Main for the details.

Copyright © 2026, Oracle and/or its affiliates.



Payments are initially created in the Incomplete state. Payments in this state don't have payment segments or financial transactions; they are simply a stub awaiting distribution.

Click **Distribute** to distribute a payment amongst an account's service agreements.

- If the system cannot distribute the payment (for whatever reason), the payment is moved to the Error state. You may delete such a payment.
- If the system successfully distributes a payment, the payment becomes Freezable.

Click the - button to physically remove an Incomplete, Error or Freezable payment from the database.

Click **Freeze** to freeze a payment and its financial transaction. Freezing the payment causes the following to occur:

- The system executes any payment freeze algorithms linked to the account's customer class and to the service agreement's SA type.
- The payment's state becomes Frozen and the payment may now appear on a customer's bill.

You may not change a payment once it is frozen. However, you may reverse the payment's financial effect by clicking **Cancel**. Clicking this button will cause the following to occur:

- A new financial transaction is generated and linked to the payment. This financial transaction reverses the financial effects of the original payment.

Copyright © 2026, Oracle and/or its affiliates.

- The system executes any payment cancellation algorithms linked to the account's customer class.
- If the payment has a related adjustment, this adjustment is also cancelled. Refer to Loan Overpayments for an example of payments with adjustments.
- The payment becomes Canceled.

Payment Event - Add Dialog

The Payment Event transaction features an unusual dialog that simplifies the addition of new payment events. This page appears if you open the **Menu**, and then **Financial**, and then **Payment Event**, and then **Add** from either the Account context menu or from Menu (it also appears if you click the clear button when on the Payment Event page).

> **Note**
>
> If you have opted to always use the payment event distribution rules method as your default method, the Payment Event Quick Add (Single Payment Event) page appears instead.
>
> There is an Add One-Time Payment dialog that behaves similarly to Payment Event - Add, but adds a number of capabilities related to automatic payment methods. Refer to Add One-Time Payment for more details. To use the Add One-Time Payment popup instead of the Payment Event - Add popup, refer to One-Time Payment Options for details on enablement.

Description of Page

The **Payor Account ID** is the account that remitted the payment. We assume this account is both the tendering account and the account whose debt is being relieved by the payment. If this assumption is not correct, choose a **Distribute Action** of Do Not Distribute and then change the tendering or paying account when the Payment Event page appears (after you click **OK**).

> **Note**
>
> **Default note.** If you have navigated to this page from account context menu, the Payor Account ID and Payment Amount are defaulted to this account.

The **Payment Amount** is the amount of the customer's debt to be relieved by the payment. Note, this amount is defaulted using an algorithm plugged in on the Installation Record. Please refer to the Automatic Payment Amount Calculation algorithm, (C1-APAM-DFLT) for more information on the calculation.

The payment tenders grid allows you to enter multiple tender types and amounts. Click + to add a new tender. For each tender specify the following fields:

- The **Amount Tendered** is the amount of moneys remitted for the tender type.
- **Tender Type** describes what was remitted (e.g., cash, check, ...). **Tender Type** defaults from the Quick Add Tender Type that is defined on the installation record.

Copyright © 2026, Oracle and/or its affiliates.

- If a check was tendered, use **Check Number** to specify the identity of the check.

> **Note**
>
> **Cash back causes an additional tender to be created.** If cash should be returned to the customer (because the customer overpaid and the tender type's cash back allowed switch is true and the tender type is not "like cash"), a negative tender for the cash back amount is created for the payment event. Refer to Cash Back for a description of how the system can recommend Cash Back amount if the customer tendered more than they are paying.

Match Type and **Match Value** are used if either of the following conditions is true:

- This **Payor Account** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill ID" and a match value of the bill ID being paid.

> **Note**
>
> **Shortcut.** If you enter a **Match Type** of "bill ID" and leave the **Match Value** blank, the system assumes the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

The **Payment Date** defaults to the current date.

If the **Payor Account ID**'s customer class is designated as non-CIS (i.e., the person making the payment isn't a customer), the following information appears in the above window:

- **Non CIS Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.

Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button:

- Choose Distribute and Freeze if OK if this is a simple payment that should require no manual intervention. By "simple payment" we mean:
  - The account is both the tendering account and the account whose debt is being relieved by the payment
  - The payment date is the current date
  - The payment should be distributed amongst the account's service agreements using standard distribution logic
- If this option is selected, the system distributes the **Payment Amount** amongst the account's service agreements. If the distribution is successful, the system automatically freezes the payment. If the distribution is not successful, the payment will be in the Error or

Copyright © 2026, Oracle and/or its affiliates.

Incomplete state. When the Payment Event page appears, you can view the error and then correct it. After the cause of the error is corrected, you must distribute and freeze the payment manually (this can be done on several pages including Payment Event - Main and Payment - Main).

- Choose Manual Distribution if you need to manually distribute the payment amongst the account's service agreements. If this option is selected, the system creates a payment event, a tender and a payment and then transfers you to the Payment - Manual Distribution page where you can define the amount to be allocated to each of the account's service agreements. After you've distributed the payment, don't forget to freeze it.
- Choose Do Not Distribute if you want to process the payment event manually (e.g., if you need to define multiple accounts whose debt is relieved by the payment). If this option is selected, the Payment Event - Main page opens with the information you entered defaulted accordingly. You can make any changes you want and then distribute and freeze the payment. Refer to How To Add A New Payment Event for more information.

> **Note**
>
> **Cash-only warning.** If the account has exceeded your cash-only threshold and the **Tender Type** isn't marked as "like cash", a warning appears advising of such. Note, a customer's cash-only points are maintained on Account - Credit Rating.

Add One-Time Payment

The Add One-Time Payment user interface allows for quick add of one-time payment events using any of the following:

- Stored payment methods from self-service one-time payments and wallet maintenance
- A currently effective account auto pay option for recurring bill automatic payments
- New checking or savings payment methods - subject to web debit account validation, if the bank account is being used for the first time
- Cash
- Non-cash, non-auto pay tender types - e.g., check

> **Note**
>
> The Add One-Time Payment user interface is an alternative for the Payment Event - Add Dialog. Refer to One Time Payment Options for information on how to enable it.

When enabled, the Add One-Time Payment dialog can be launched from any of the following:

- **Take A Payment** action from the Customer sidebar zone.
- Main Menu: **Menu**, and then **Financial**, and then **Payment Event**, and then **Add**
- Account context menu: **Payment Event**, and then **Add**

In addition, the Add One-Time Payment script can also be called from other places (zones, maps, etc) in the system where your implementation allows the capture of a one-time payment.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If you have opted to always use the payment event distribution rules method as your default method, the Payment Event Quick Add (Single Payment Event) page appears instead.

The **Payment Date** defaults to the current date.

The **Payor Account ID** is the account that is tendering the payment. This is defaulted from the Account in context, if any.

> **Note**
>
> **Cash-only warning.** If the account has exceeded your cash-only threshold a message appears advising of such. A customer's cash-only points are maintained on the Account - Credit Rating page.

The **Payment Amount** is the amount of the customer's debt to be relieved by the payment. Note, this amount is defaulted using an algorithm plugged in on the Installation Record. Please refer to the Automatic Payment Amount Calculation algorithm, (C1-APAM-DFLT) for more information on the calculation.

The **Tenders - Existing Payment Options** grid is shown if the Payor Account has existing self-service payment methods and/or a currently effective recurring bill auto pay enrollment. Select a **Payment Option** and specify the **Tender Amount**.

The **Tenders - New Auto Pay Options and Other Tender Types** grid is used for new automatic payment methods, cash tender type(s) and other non-cash/non-auto pay tender types (e.g. check). Select a **Tender Type** and specify the applicable information. A new checking/savings payment method can be saved, to make it available for future one-time payments, including self-service payments. Specify the **Tender Amount**.

> **Note**
>
> **Cash back causes an additional tender to be created.** If cash should be returned to the customer (because the customer overpaid and the tender type's cash back allowed switch is true and the tender type is not "like cash"), a negative tender for the cash back amount is created for the payment event. Refer to Cash Back for a description of how the system can recommend Cash Back amount if the customer tendered more than they are paying.
>
> **Credit card tender types are not supported.** To ensure compliance with credit card processing rules, it is highly recommended that credit card payments be processed through a third-party payment processing system. If your implementation needs to support credit card tender types in this dialog, you can create a custom BPA script to override the base-owned BPA script. Refer to <xref? One Time Payment Options> for details.

Match Type and **Match Value** are shown used if either of the following conditions is true:

Copyright © 2026, Oracle and/or its affiliates.

- The **Payor Account** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill ID" and a match value of the bill ID being paid.

> **Note**
>
> **Shortcut.** If you enter a **Match Type** of "bill ID" and leave the **Match Value** blank, the system assumes the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

If the **Payor Account ID**'s customer class is designated as non-CIS (i.e., the person making the payment isn't a customer), the following information appears in the above window:

- **Non CIS Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.

Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button:

- Choose **Distribute and Freeze if OK** if this is a simple payment that should require no manual intervention. "Simple payment" means:
  - The account is both the tendering account and the account whose debt is being relieved by the payment
  - The payment date is the current date
  - The payment should be distributed amongst the account's service agreements using standard distribution logic

  If this option is selected, the system distributes the **Payment Amount** amongst the account's service agreements. If the distribution is successful, the system automatically freezes the payment. If the distribution is not successful or is on hold due to web debit account validation, the payment will be in the Error or Incomplete state, respectively. When the Payment Event page appears, you can view the error and then correct it. After the cause of the error is corrected, you must distribute and freeze the payment manually (this can be done on several pages including Payment Event - Main and Payment - Main). If any new auto pay methods are checking/savings bank accounts that have never been used on prior recurring auto pays and/or prior self-service one-time payments, the bank accounts will be subject to web debit account validation if the system is configured to do so. The One-Time Payment service task is created to manage bank account validation. The service task is linked to the payment event, which remains in Incomplete status until the validation is complete. The service task will either complete or delete the payment event, depending on the validation results. Refer to Web Debit Account Validation for more details.
- Choose **Manual Distribution** if you need to manually distribute the payment amongst the account's service agreements or if you want to process the payment event manually (e.g. if you need to define multiple payee accounts). If this option is selected, the system creates a payment event, a tender and a payment and then transfers you to the Payment - Manual

Copyright © 2026, Oracle and/or its affiliates.

Distribution page where you can define the amount to be allocated to each of the account's service agreements. After you've distributed the payment, don't forget to freeze it. If allocating the payment to multiple payee accounts, navigate to the Payment Event - Main page and make any changes before distributing and freezing the payment.

Payment Event - Main Information

The Main page contains core payment event information. Open this page using **Menu**, and then **Financial**, and then **Payment Event**, and then **Search**.

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To for a description of how to perform common payment event maintenance functions.

Description of Page

Note

Cash-only customers. When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer and the tender's tender type is not cash. An account is considered cash-only if their current cash-only points exceed the cash-only threshold maintained on the Installation record or account's CIS Division. A customer's cash-only points are maintained on Account - Credit Rating.

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, Multiple Distribution Details Exist is displayed instead. If document numbering has been enabled, the assigned document number will appear. If the payment was made in an alternate currency, the message Alternate Currency Used is displayed. **Pay Event Info** is only displayed after the payment event has been added to the database.

Refer to Alternate Currency Payments and Document Numbers for more information. **Payment Event ID** is the system-assigned unique identifier of the payment event.

The area under **Pay Event Info** provides warnings about the payment event. Possible warnings are Unfrozen Payments and The Payment Event is Unbalanced. A warning is also issued to the cashier to remind them to turn in funds - see Turn Ins.

If multiple distribution details are linked to the payment event, the distribution rule, value, and amount for each distribution detail displays.

Payment Date is the business date associated with the payment event.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> If you change the payment date and this event's tender(s) are automatic payments, the extract date (i.e., the date the automatic payment is sent to the financial institution) will be changed to equal the payment date. Refer to Automatic Payments **for more information**.

Payment(s) contains the total number and value of payments linked to the event.

Tender(s) contains the total number and value of tenders linked to the event.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to Cash Back for details).

If the payment event becomes unbalanced , a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to Cash Back for details.

While most payment events contain a single payment, the system allows many payments to exist under a payment event (when the payment event's tender(s) are distributed amongst multiple accounts). If a payment event has a large number of payments, you can use the **Account Filter** to limit the payments that appear in the **Payments** grid. The following options are available:

- Account. Use this option if you only want to see the payment linked to an **Account ID**.
- All. Use this option to view all payments linked to the payment event.
- Person Name. Use this option to restrict payments linked to accounts whose main customer has a primary name that matches **Person Name**.

The **Payments** grid contains the payment(s) linked to the payment event. The following points describe the attributes in this scroll; refer to How To for instructions describing how to perform common maintenance activities.

- **Account ID** references the payment's account. The name of the account's main customer is displayed adjacent.
- **Payment Amount** is the amount of the account's debt relieved by the payment. The adjacent context menu allows you to drill down to the details of the payment (this is where you can see the payment's payment segments and where you can override the distribution of the payment).
- **Payment Status** is the payment's status. If the payment's status is Error, the error message is displayed adjacent. Refer to Payment Lifecycle for the potential values and how to handle a payment when it exists in a given state.
- **Match Type** and **Match Value** should only be used if either of the following conditions is true:
  - This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

Copyright © 2026, Oracle and/or its affiliates.

  - The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

- The remaining columns are only used if the payment is linked to an **Account ID** that belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information. If such an account exists, the following fields must be defined.
  - **Non CIS Name** is the name of the person remitting the payment.
  - **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
  - **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.
- Note, you can also define a comment for a non-CIS payment. To define a comment, use the context menu adjacent to **Payment Amount** to drill to Payment - Main.
- **Payment ID** is the unique, system-assigned identifier of the payment. This value only appears after the payment has been added to the database.

Refer to Payment Actions and Payment Event Actions for information about the action buttons on this page. Refer to How To for a description of typical business processes that use these buttons.

Payment Event - Tenders

The Tenders page contains a scroll showing one row for every tender associated with the payment event. Open this page using **Menu**, and then **Financial**, and then **Payment Event**, and then **Search** and navigate to the **Tenders** page.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To for instructions describing how to perform common maintenance functions.

Description of Page

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, Multiple Distribution Details Exist is displayed instead. If document numbering has been enabled, the assigned document number will appear. If the payment was made in an alternate currency, the message Alternate Currency Used is displayed. **Pay Event Info** is only displayed after the payment event has been added to the database.

Payment Event ID is the system-assigned unique identifier of the payment event.

Payment Date is the business date associated with the payment event.

Payment(s) contains the total number and value of payments linked to the event.

Tender(s) contains the total number and value of tenders linked to the event.

Copyright © 2026, Oracle and/or its affiliates.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to Cash Back for details).

If the payment event becomes unbalanced, a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to Cash Back for details.

The **Tenders** scroll controls the display of the tenders linked to the payment event. The following simply describes the fields in this scroll; refer to How To for instructions describing how to perform common tender maintenance activities.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning Cash-Only appears.

> **Warning**
>
> If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to Automatic Payments **for more information**.

- **Tender Amount** is the amount of the tender. The amount is disabled if the payment was made in an alternate currency.
- If a check was used, the **Check Number** contains the identity of the check.
- **Pay Tender ID** is the system-assigned unique identifier of the tender. This value appears after the tender has been added to the database.
- **Tender Type** describes what was remitted (e.g., cash, check). If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.

> **Note**
>
> **Authorizing Tenders.** The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

> **NOT_SUPPORTED**
>
> **For information** about tender types requiring authorization, refer to Credit Card Payments.

- If the payment tender was remitted in an alternate currency, i.e. a currency other than the payor account's currency, **Alternate Currency** and **Exchange Rate** will display the formatted alternate amount ((includes the alternate amount and currency) and the exchange rate that was in effect on the payment date.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For information about accepting payments in alternate currencies, refer to Alternate Currency Payments.

- **Tender Status** is the tender's status. Refer to Tender Lifecycle for the potential values and how to handle a tender when it exists in a given state.
- If the **Tender Type** is associated with an automatic payment, the system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution. You can override automatic payment information as long as the automatic payment has not been sent to the financial institution or cancelled. The **Auto Pay Section** contains the following fields:
  - **Auto Pay Source** is the financial institution / credit card company that receives the automatic payment request.
  - **Scheduled Extract Date** is the date the automatic payment request was sent / is scheduled to be sent to the financial institution. This information is display-only.
  - **External Account ID** is the customer's account number at the financial institution.

    Note

    The system supports encryption for External Account ID. If your implementation has configured the system to encrypt External Account ID, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information.

  - **Expires On** is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment).
  - **Name** is the customer's name in the financial institution's system.
  - **Bill ID** is the ID of the bill whose completion caused the creation of the automatic payment. This information only appears if the automatic payment was created as a result of a bill. This information is display-only.
- If the payment was uploaded, the following fields MAY contain information (if they were populated in the interface file):
  - **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.

    Note

    The system supports encryption for MICR ID. If your implementation has configured the system to encrypt MICR ID, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information.

  - **Customer ID** is the customer's account ID that appeared on the interfaced payment.

Copyright © 2026, Oracle and/or its affiliates.

  - **Ext. Reference ID** is the unique identifier of the payment upload interface record.
  - **Name** is the customer's name on the interfaced payment record.
- In the **Tender Control** area, the **Tender Control ID** is the identity of the remittance processor batch or cash drawer bundle in which the tender was remitted. Refer to Tender Management and Workstation Cashiering for more information about tender controls.
- Displayed adjacent to **Tender Control** is the date, tender source, and status of the tender control.

Every new tender must reference an open tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one open, user-specific Tender Control, the system will default one, at random.
- If the user has no open, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source indicates it's used for online cashiering. If multiple exist, one will be selected at random.
- Turn off **Included in Tender Ctl Balance** if the tender should not be included in the tender amount in the tender control's tender balance. You would turn this switch off if you canceled a tender because it was mistakenly applied to the wrong account (or the amount was wrong). This switch is only enabled if:
  - The tender control isn't balanced.
  - The tender is canceled.
- **Deposit Control** contains a concatenation of the tender control's deposit control's creation date, tender source type, and status. Refer to Tender Management and Workstation Cashiering for more information about deposit controls.
- The **Tender Action** area contains a button that you use to cancel a tender. Refer to Tender Actions for information about this button. Refer to How To Cancel A Tender for more information.

The **Characteristics** collection on the Tenders page contains information that describes miscellaneous information about the tender. The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Payment Event - Characteristics

To update payment event characteristics, open **Menu**, and then **Financial**, and then **Payment Event**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the payment event.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Copyright © 2026, Oracle and/or its affiliates.

Characteristic Value Indicate the value of the characteristic.

Exception

The topics in this section describe exceptions that are detected when the system allocates a payment.

- Payment Event Exception
- Payment Exception
- Resolving Exceptions Automatically

Payment Event Exception

It is possible for a payment event's tenders to not equal its payments. Such events are classified as unbalanced. For each unbalanced payment event, a record is written to the Payment Event Exception table (CI_PAY_EVT_EXCP). A To Do background process creates To Do entries for records in this table.

Unbalanced payment events cause a record to be written to the payment event exception table with a message indicating the nature of the error.

To view the messages associated with the exception records, schedule the TD-UNBAL background process. This process generates a To Do entry for every record in the payment event exception table.

NOT_SUPPORTED

Refer to Unbalanced Payment Events for instructions describing how to correct an unbalanced payment event.

Payment Exception

When the system is unable to distribute a payment, a record to be written to the payment exception table with a message indicating the nature of the error. A common example of a payment exception is **no service agreement to hold a credit.** This can occur when an account overpays their debt and the account doesn't have a single service agreement that is allowed to hold a credit, a payment error is generated.

The system saves payments that are in error just as it saves payments that are error-free. This is done because payments are nothing more than a snapshot of the data that was used to distribute the payment. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively correct it.

Every payment in error is written to the Payment Exception table (CI_PAY_EXCP). A To Do background process creates To Do entries for records in this table.

To view the messages associated with the exception records, schedule the TD-PYERR background process. This process generates a To Do entry for every record in the payment exception table.

Copyright © 2026, Oracle and/or its affiliates.

After correcting the cause of the error, drill into the Payment and attempt to redistribute it.

Resolving Exceptions Automatically

Some payment errors occur because master data was not fully set up prior to receiving a payment for the account. For these cases, the system will periodically check to see whether the master data problem has been resolved by attempting to distribute and freeze the payment in error.

A background process, Resolve Payments in Error - PY-RPE, exists for this purpose. This background process works as follows:

- It looks for payments in error where the error was caused by the lack of active service agreements linked to the payment's account. Note, this may occur when a customer is verbally asked to pay a connection fee or deposit prior to starting service, and for some reason, the appropriate service agreement has not been set up yet, or is still pending.
- For each such payment, it attempts to re-distribute the payment. If service agreements have been created in the meantime, this payment will distribute and freeze successfully.

Payment Event Action Codes

The topics that follow describe each of the actions that appear on the payment event pages.

- Payment Actions
- Payment Event Actions
- Tender Actions

Payment Actions

The topics that follow describe payment-oriented actions. Refer to How To for a description of several business processes that use these buttons.

- Distribute (Payments)
- Redistribute (Payments)
- Print
- Freeze (Payments)

Distribute (Payments)

Clicking the **Distribute** button distributes all Incomplete, Error, or Freezable payments linked to the event. Refer to Distribute (A Payment) for information of how a single payment is distributed.

This button is enabled if at least one payment is Incomplete, Error, or Freezable.

Copyright © 2026, Oracle and/or its affiliates.

Redistribute (Payments)

Clicking the **Redistribute** button redistributes all Incomplete, Error, or Freezable payments linked to the event. Refer to Redistribute (A Payment) for information of how a single payment is redistributed.

This button is enabled if at least one payment is Incomplete, Error, or Freezable.

Print

The Print button is enabled under the following conditions:

- The payment event must have at least one payment that is Frozen and no payments that are Incomplete, Error or Freezable.
- A processing script is configured in the Point Of Sale (POS) Printer Integration master configuration.
  - For the Payment Event page, a **POS Printing - Payment Event** BPA script should be configured.
  - For the Payment Event Quick Add page, a **POS Printing - Payment Event Quick Add** BPA script should be configured.
  - For the Payment Quick Add page, a **POS Printing - Payment Quick Add** BPA script should be configured.

Refer to How To Print Receipts And Endorsements for more information.

Freeze (Payments)

Clicking the **Freeze** button causes all Freezable payments linked to the event to become frozen. Refer to Freeze (A Payment) for information of how a single payment is frozen.

This button is enabled if at least one payment is Freezable.

> **Note**
>
> **If problems are detected after freezing.** A payment may not be changed after it is frozen. All subsequent changes must occur by canceling the frozen payment and creating a new one. You can cancel and redistribute a payment on the next tab. Refer to Canceling A Tender Versus Canceling a Payment for more information.

Payment Event Actions

The topics that follow describe payment event-oriented actions. Refer to How To for a description of typical business processes that use these buttons.

- Transfer

Copyright © 2026, Oracle and/or its affiliates.

- Delete A Payment Event

Transfer

If the payment event was created using distribution rule(s), any transfer of its payments should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment event is associated with at least one distribution detail or not.

Transfer (No Distribution Details Exist)

The **Transfer** button is enabled if:

- There is a single non-cancelled tender linked to the event AND
- There is a single frozen payment linked to the event AND
- The account on the tender is the same as the account on the payment.

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in the **Payment Event Transfer** dialog to transfer a payment:

- **Account ID** is the account to which the payment event should be transferred.
- **Cancel Reason** defines why you are performing the transfer.
- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match Type and Match Value are used for an account that belongs to an open item customer class or to restrict the distribution of the payment to a specific service agreement. Refer to Payment Event - Main for more information.
- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information.

Clicking **OK** causes the following to take place:

- The account on the tender is changed to reflect the transfer to account.
- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- The new payment is distributed.
- If so designated, the new payment is frozen.

Transfer (Distribution Details Exist)

The **Transfer** button is enabled if:

- There is a single payor AND
- All existing payees before the transfer are the same as the payor AND

Copyright © 2026, Oracle and/or its affiliates.

- There is at least a single non-cancelled tender linked to the event

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in this dialog to transfer payment(s):

- **Distribution Rule** is the account to which the payment event should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment(s) associated with the original account are canceled.
- The new set of distribution details are processes, creating new payment(s) for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- If all new payments are for the same payee account (and the payee is different than the current payor) the account on the tender is changed to reflect the transfer to account.
- Original distribution details (only those with a non-zero amount) are deleted and replaced by the new set of details on the payment event.

> **Note**
>
> **Distribution rule entries can have a zero amount.** Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.
>
> **Determine Tender Account.** This dialog does not call the Determine Tender Account algorithm defined on distribution rule as this action does not rebuild the payment tender collection.

> **NOT_SUPPORTED**
>
> Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.

Delete A Payment Event

The **Delete** button deletes a payment event and its tenders and payments.

This button is enabled if there are no frozen or canceled payments and no canceled tenders.

Tender Actions

The topics that follow describe tender-oriented actions. Refer to How To for a description of several business processes that use these buttons.

Copyright © 2026, Oracle and/or its affiliates.

- Cancel A Tender

Cancel A Tender

Clicking **Cancel** causes the following to take place:

- All payments associated with the tender's payment event are canceled.
- The tender becomes canceled.
- If the cancellation reason indicates the cancellation is due to non-sufficient funds, other actions may occur. Refer to NSF Cancellations for more information.

The **Cancel** button is enabled if the tender is not canceled

When clicked, the **Tender Cancel** window opens.

You must specify the following parameters to cancel a tender:

- Select a **Cancel Reason** to describe why the tender is being canceled.
- The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank** and **Bank Account Key** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

> **Note**
>
> **When non-canceled tenders are still linked to the event.** If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.
>
> **Cancellation after cash back.** If a customer tenders a check for $100, but only owes $80, the system will recommend returning $20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not "like cash", a second tender is created for -$20 (the first tender is for $100). If the check subsequently bounces, both tenders must be cancelled.

Payment Event Quick Add

The Payment Event Quick Add page is used to quickly add, distribute and freeze one or more payment events using distribution rules. Open this page using **Menu**, and then **Financial**, and then **Payment Event Quick Add**, and then **Search.**

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an open tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one open, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.

Copyright © 2026, Oracle and/or its affiliates.

- If the user has no open, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source has a tender source type of online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Use **Number of Payment Events** to indicate whether you intend to add a single payment event or multiple payment events.

Choose the multiple payment events option if you intend to add "simple" payment events. By "simple" we mean:

- A single account is both the payor account and the account whose debt is being relieved by the payment.
- The payment tender is not an auto-pay tender.

For all other cases choose the single payment event dialogue as it is designed to handle more complex payment event configurations such as when:

- The payor account is different than the payee account.
- Multiple payors cover payments for one or more payees.
- The payment tender is an auto-pay tender.

Once you have made your selection the page is adjusted to support the corresponding dialog.

> **Note**
>
> **Cash-Only customers.** When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer. An account is considered cash-only if their current cash-only points exceed the cash-only threshold maintained on the Installation record or the account's CIS Division. A customer's cash-only points are maintained on Account - Credit Rating.

- Multiple Payment Events Dialog
- Single Payment Event Dialog

Multiple Payment Events Dialog

This section describes the default dialog setup when navigating from **Menu**, and then **Financial**, and then **Payment Event Quick Add**

Description of Page

Using the multiple payment events option, each row in the payment detail grid typically represents a unique distribution detail. However, more than one row in the grid can belong to a single payment event. All distribution details for identical tender account will be grouped under a single pay event.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a default distribution rule, it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.

Copyright © 2026, Oracle and/or its affiliates.

- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the installation record.

> **Note**
>
> **Authorizing Tenders.** The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

> **NOT_SUPPORTED**
>
> **For information** about tender types requiring authorization, refer to Credit Card Payments.

> **Note**
>
> **Automatic Payments.** The multiple payment events dialog does not support automatic payments. Switch to the single payment event dialog to enter automatic payments.

- Use **Check Number** if a check is remitted.
- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.

> **Note**
>
> The system supports encryption for MICR ID. If your implementation has configured the system to encrypt MICR ID, the data will be displayed with a masked value; such as, *******.Refer to Encryting Sensitive Data for more information.

- You may use **Customer ID** to record additional customer information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.
- **MICR ID** is the MICR ID associated with the tender.
- **Payor Account ID** is the tender account as determined by the Determine Tender Account algorithm defined on the distribution rule. This information is only populated after the distribution detail has been processed.

After specifying the various payment distribution details in the grid, click **Create**. The system attempts to create payment event(s) as follows:

- As mentioned earlier, all distribution details for identical tender accounts are grouped under a single pay event. The first step in identifying common tenders is to determine the tender account. To do that the system calls the Determine Tender Account algorithm defined on the distribution rule.

Copyright © 2026, Oracle and/or its affiliates.

- A payment event is created for each unique tender account.
- All rows having all attributes of the tender identical except for the amount are grouped together where each group represents a single tender.
- A single payment tender is created for each unique tender and linked to the payment event.
- The total payment amount for each distinct group of distribution details (for the payment event) having the same rule and value is summarized. For each distinct group, the system calls the Create Payment algorithm defined on the distribution rule providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for this payment event.
- Add the distinct payment event distribution detail and its total amount beneath the payment event.
- If your implementation is configured for printing at a cashier station, the Print dialog appears. Refer to How To Print Receipts and Endorsements for more details.

The **Payment Event ID** column appears in the grid after the distribution details have been processed. It shows the payment event ID created for the distribution detail record and a short description as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

After you've added a group of payment events, you should press the clear button (or Alt-C), to ready the page for the next group of payment event details.

Single Payment Event Dialog

If you have opted to always use the payment event distribution rules method as your default method, this section describes the default dialog setup when you navigate to the **Payment Event** page in **Add** mode.

Adding a single payment event at a time, allows for more complex payment event information to be captured.

Enter one row in the **Tenders** grid for every unique tender associated with the payment event.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning Cash-Only appears.

> **Warning**
>
> If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to Automatic Payments **for more information**.

- **Tender Amount** is the amount of the tender.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the installation record.
- Use **Check Number** if a check is remitted.

Copyright © 2026, Oracle and/or its affiliates.

- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.

> **Note**
>
> The system supports encryption for MICR ID. If your implementation has configured the system to encrypt MICR ID, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information.

- You may use **Customer ID** to record additional customer information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.
- If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.
- If the **Tender Type** is associated with an automatic payment, the system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.
  - **Auto Pay Source** is the financial institution / credit card company that receives the automatic payment request.
  - **External Account ID** is the customer's account number at the financial institution.
  - **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
  - **Name** is the customer's name in the financial institution's system.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a default distribution rule, it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.
- Use **Payment Amount** to define the amount of the payment.

After specifying the various payment distribution details in the grid, click **Create**.

> **Note**
>
> **Go To Payment Event.** If you wish to be transferred to the payment event page once processing is complete, remember to check the Go To Payment Event check box before you click **Create**. This field does not appear when this page is used as the default Payment Event - Add Dialog and you are automatically transferred to the payment event page once the payment event is created.

After distribution details have been processed, a short description of the payment event that has been created is displayed to the right of the **Go To Payment Event** field. The description

Copyright © 2026, Oracle and/or its affiliates.

provides information as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

The system attempts to create the payment event as follows:

- Validate that the sum of the tenders is not less than the sum of the payments.
- Create the payment event with a separate payment tender for each row in the Tenders grid.
- Summarize the total payment amount for each distinct group of distribution details having the same rule and value. For each distinct group, call the Create Payment algorithm defined on the distribution rule providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for the payment event.
- Add the distinct payment event distribution detail summary beneath the payment event.
- If your implementation is configured for printing at a cashier station, the Print dialog appears. Refer to How To Print Receipts and Endorsements for more details.

> **Note**
>
> **Determine Tender Account.** Having an explicit collection of tenders eliminate the need to determine the tender account. Therefore this dialog does not call the Determine Tender Account algorithm.

Payment Quick Add

The Payment Quick Add page is used to quickly add, distribute and freeze payment events for up to fifteen accounts. Open this page using **Menu**, and then **Financial**, and then **Payment Quick Add.**

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an open tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one open, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.
- If the user has no open, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source has a tender source type of online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Enter one row in the grid for every payment event to be added. The following information is entered for each payment event:

- Use **Account ID** to define the customer who tendered the payment. It's important to note that this transaction assumes that the tendering account is the same as the account whose balance is relieved by the payment. Refer to How To Allocate The Tender Amount To Multiple Accounts if you need to distribute the payment to an account (or accounts) other than the tendering account.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Cash-Only customers.** You cannot use this transaction to add a payment for a "cash-only" account if the tender type is not "like cash". An account is considered cash-only if its cash-only points exceed the cash-only threshold maintained on the Installation record. A customer's cash-only points are maintained on Account - Credit Rating. You must use the Payment Event transaction to add such a payment (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).

- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the installation record.

> **Note**
>
> **Authorizing Tenders.** The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

> **NOT_SUPPORTED**
>
> **For information** about tender types requiring authorization, refer to Credit Card Payments.

- Use **Check Number** if a check is remitted.
- **Match Type** and **Match Value** are only used if either of the following conditions is true:
  - This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

> **Note**
>
> **Shortcut.** If you enter a **Match Type** of "bill id" and leave the **Match Value** blank, we assume the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

After specifying the various accounts and amounts in the grid, click the **Distribute and Freeze** button. When you click this button, the system attempts to create a payment event for each row in the grid. Four potential outcomes are possible for each row:

Copyright © 2026, Oracle and/or its affiliates.

- If the data you entered for a payment event isn't complete (e.g., you don't specify a valid account or amount):
  - No payment event is created.
  - A **Message** describing the problem is displayed.
  - All fields on the row remain modifiable.
- You should correct each such line and then press the **Distribute and Freeze** button.
- If the data you entered is complete, but the system issues a warning (e.g., the account is considered "cash-only" and a tender type other than cash is remitted, or if the account already has a payment for this amount on this date):
  - No payment event is created.
  - A **Message** containing the warning is displayed.
  - All fields on the row remain modifiable.
- You must clear the line and add a payment using the Payment Event transaction (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).
- If the data you entered is complete, but the system is not successful in distributing the payment:
  - The system creates a payment event, a payment and a tender.
  - A **Message** describing the problem is displayed.
  - The payment's **Status** is displayed
  - All fields on the row are disabled.
- You must press the adjacent go to button to drill to the payment event where the error can be corrected.
- If the data you entered is complete and the system is successful in distributing the payment:
  - The system creates a payment event, a payment, and a tender.
  - The system distributes the payment amongst the account's service agreements and then freezes the payment.
  - The payment's **Status** is displayed.
  - All fields on the row are disabled.
  - If your implementation is configured for printing at a cashier station, the Print dialog appears. Refer to How To Print Receipts and Endorsements for more details.
- You can press the adjacent go to button to view a payment event.

> **Note**
>
> **Separate Commits.** This page is unusual in that each payment event is committed to the database independently. For example, if you enter seven payment events and one is invalid, six payment events will be added to the database when you press **Distribute and Freeze**. When the page is redisplayed, the rows containing the valid payments are protected and an indication of their validity is displayed. The row containing the invalid payment remains unprotected. You can correct the erroneous payment and then press the **Distribute and Freeze** button again.

Copyright © 2026, Oracle and/or its affiliates.

After you've added a group of payments, you should press the clear button (or Alt-C), to ready the page for the next group of payment events.

Maintaining Payments

Payments and payment segments are automatically created by the system when payment events are created. You should only need to access this transaction as follows:

- To view a payment's payment segments (i.e., to view how a payment was distributed amongst its account's service agreements).
- To override the distribution of a payment amongst its account's service agreements. Refer to Distributing A Payment Amongst An Account's Service Agreements for information describing how payments are typically distributed.

The topics in this section describe how to maintain a payment.

- Payment - Main
- Payment - Pay Segments
- Payment - Manual Distribution
- Payment - Characteristics
- Payment Action Codes

Payment - Main

This page contains basic information about a payment. Open this page using **Menu**, and then **Financial**, and then **Payment**, and then **Search.**

> **Warning**
>
> Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's service agreements.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To for instructions on how to perform common maintenance functions.

Description of Page

Payment Info contains a concatenation of the payment date, amount, status, and the name of the main customer on the account.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main

Copyright © 2026, Oracle and/or its affiliates.

customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are Unfrozen Payments and The Payment Event is unbalanced.

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Account ID is the account whose debt is reduced by the payment. The name of the account's main customer is displayed adjacent. This field is gray after the payment is frozen.

Payment Amount is the amount of the payment. This field is gray after the payment is frozen. If the account has exceeded your cash-only threshold, a warning (in red) appears adjacent to **Payment Amount**. Note, a customer's cash-only points are maintained on Account - Credit Rating.

Payment Status is the payment's status. Refer to Payment Lifecycle for the potential values and how to handle a payment when it exists in a given state. If the payment is in Error, the error message is displayed adjacent. If the payment is canceled , the **Cancel Reason** is displayed adjacent.

Match Type and **Match Value** will only be used if either of the following conditions is true:

- This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.
- The customer wants to restrict the distribution of the payment to a specific service agreement. In this situation, specify a **Match Type** of "service agreement ID" and a **Match Value** of the respective service agreement ID.

Match Type and **Match Value** are gray after the payment is frozen.

If the payment is linked to an **Account ID** that belongs to a customer class that is used for non-CIS payments, the following fields appear (note, these fields are gray after the payment is frozen):

- **Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Comments** are used to describe anything unusual about the non-CIS payment.

The contents of the **Payment Segments** section depends on the number of payment segments linked to the payment. If more than 25 payment segments exist, a message appears that allows you to navigate to Payment - Payment Segments where you can view the payment segments (this page has special functionality that allows you to defined which payment segments should be displayed). If 25 or fewer payment segments exist, the grid contains information about each payment segment:

- **Premise** is the address of the service agreement's main premise.
- **SA Information** contains basic information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.

Copyright © 2026, Oracle and/or its affiliates.

- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Refer to Payment Action Codes for information about the action buttons on this page.

Payment - Pay Segments

You can use this page to view all or selected payment segments linked to a payment.

Open **Menu**, and then **Financial**, and then **Payment**, and then **Search** and navigate to the **Payment Segments** page to view this information.

> **Warning**
>
> Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments.

> **Note**
>
> If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of payment segments that should be retrieved. See the Description of page below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are Unfrozen Payments and The Payment Event is unbalanced.

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of payment segments that should be retrieved.

Use the **SA Filter** to define the types of service agreements whose payment segments appear in the grid. The following options are available:

- Address. Use this option to restrict payment segments to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you do not wish to restrict payment segments based on service agreement attributes.
- Geographic Type. Use this option to restrict payment segments to those whose service agreements are linked to service points associated with a given **Geographic Type** and **Geographic Value**.
- SA Type. Use this option to restrict payment segments to those whose service agreements are linked to a given **CIS Division** and **SA Type**.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the payment segments that match your search criteria. The following information appears in the grid:

- The **Premise** column contains the characteristic premise associated with the payment segment's service agreement. Refer to Maintaining Premises for more information about this premise.
- Click the go to button to transfer to view the financial transaction related to the payment segment. This is only available after the payment has been frozen.
- **SA Information** contains summary information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the distribution.
- **Adjustment ID** is displayed if there is an adjustment associated with the payment. Refer to Loan Overpayments for an example of when an adjustment is associated with a payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Copyright © 2026, Oracle and/or its affiliates.

Payment - Manual Distribution

You can use this page to view service agreements linked to the payment's account and to manually distribute a payment amongst specific service agreements.

Open this page using **Menu**, and then **Financial**, and then **Payment**, and then **Search** and navigate to the **Manual Distribution** page.

> **Warning**
>
> Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's service agreements.

> **Note**
>
> If the payment's account has more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved. See the Description of page below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

> **Note**
>
> If the payment's account has more than 25 service agreements, the search criteria are intentionally left blank in order to avoid retrieving all service agreements (with the resultant slow response times). You must therefore use the **SA Filter** to define the type of service agreements that should be retrieved.

Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:

- Address. Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you want to view all service agreements.
- Geographic Type. Use this option to restrict service agreements to those linked to service points associated with a given **Geographic Type** and **Geographic Value**.
- SA Type. Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.

Copyright © 2026, Oracle and/or its affiliates.

Don't forget to click the search button after changing the **SA Filter**.

The grid that follows contains the service agreements that match your search criteria. The following information appears in the grid:

- **Premise** contains the characteristic premise associated with the service agreement.
- **SA Information** contains summary information about the service agreement.
- **Distributed Amt** is the amount of the service agreement's debt relieved by the payment. This field is gray after the payment is frozen.
- **Billed Amt** is the amount that was billed for the service agreement on the bill preceding the payment date.
- **Delinquent Amt** is the amount of debt associated with financial transactions that appear on overdue bills.
- **Current Balance** is the amount currently owing for this service agreement.

> **Note**
>
> If the account being paid is an **open item** account, then an alternate grid is displayed. The alternate grid is designed to break down payable balances by individual bill and service agreement. Both unpaid bill balances and new charges are displayed. After distribution, a separate match event will be created for each bill paid or partially paid.
>
> that an implementation can override the type of grid displayed for open item accounts by setting up a Financial Transaction Options Feature Configuration with an Always Use Manual Payment Distribution By SA option set to Y. With this in place, the system will always display the manual distribution by service agreement grid.

The open item manual distribution grid contains the following columns:

- **Bill Date**
  - Shows bill date if the row contains an account's unpaid bill.
  - Shows '**New Charges** ' if the row contains an account's unbilled new charges.
  - Shows '**Payment Segment** ' for the following conditions:
- Row is a pay segment of a cancelled payment.
- Row is a pay segment of an overpayment to the highest priority SA or an excess credit SA.
- Row is a pay segment of a balance forward payment distribution.
  - **Premise Information** contains the characteristic premise associated with the service agreement.
  - **SA Information** contains summary information about the service agreement.
  - **Distributed Amount**
    - You can enter an amount to distribute if the payment is non-frozen and non-cancelled, otherwise, this field is protected.
  - **Billed Amount**
    - The amount that was billed for the service agreement on the row's bill. If the row does not represent a bill amount then this will be zero.
  - **Unpaid Amount**

Copyright © 2026, Oracle and/or its affiliates.

        * This is determined by the "determine open-item bill amount" algorithm as defined on the installation option.
        * This column will contain zero if:
    - The row is linked to an open non-disputed match event with other bills' FTs on it. In this case, it is not possible to determine the bill amount.
    - The row is linked to an open disputed match event.

For either grid, the following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Payment - Characteristics

To update payment characteristics, open **Menu**, and then **Financial**, and then **Payment**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the payment. The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Payment Action Codes

The topics in this section describe each of the actions that appear on the payment pages.

- Distribute (A Payment)
- Redistribute (A Payment)
- Confirm (A Payment)
- Freeze (A Payment)
- Cancel (A Payment)
- Transfer (A Payment)
- Delete (A Payment)

Copyright © 2026, Oracle and/or its affiliates.

Distribute (A Payment)

Clicking the **Distribute** button causes a payment to be distributed amongst the account's service agreements. Refer to Distributing A Payment Amongst An Account's Service Agreements for a description of how this is achieved.

This button is enabled if a payment is Incomplete, Error, or Freezable. Refer to Payment Lifecycle for more information about these status values.

If the payment is distributed appropriately, click the Freeze button to freeze the payment and its payment segments.

Redistribute (A Payment)

Clicking the **Redistribute** button causes a payment to be redistributed amongst the account's service agreements. Refer to Distributing A Payment Amongst An Account's Service Agreements for a description of how this is achieved.

This button is enabled if a payment is Incomplete, Error, or Freezable. Refer to Payment Lifecycle for more information about these status values.

If the payment is distributed appropriately, click the Freeze button to freeze the payment and its payment segments.

> **Note**
>
> This button is only available on the Payment - Main and Payment - Manual Distribution pages.

Confirm (A Payment)

Clicking the **Confirm** button confirms that manually entered amounts associated to certain SAs will relieve their balances. This is the same process that is triggered by clicking **Save.**

This button is enabled if a value is manually entered in a Distributed Amount field.

Freeze (A Payment)

Clicking the **Freeze** button causes a payment and its payment segments to become frozen. Refer to Payment Lifecycle for more information about freezing.

This button is enabled if the payment is Freezable.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If problems are detected after freezing.** A payment may not be changed after it is frozen. All subsequent changes must occur by canceling the frozen payment and creating a new one.

Cancel (A Payment)

Clicking the **Cancel** button causes the payment and its payment segments to be canceled. Canceling a payment (as opposed to canceling a tender) is typically only done if the original distribution was frozen and it's incorrect.

This button is enabled if the payment is Frozen. Refer to Payment Lifecycle for more information about these status values.

Transfer (A Payment)

The **Transfer** button is enabled if the payment is Frozen. Refer to Payment Lifecycle for more information about these status values.

If the payment being transferred was created using payment event distribution rule(s), the payment transfer should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment's payment event is associated with at least one distribution detail or not.

- Transfer Payment (No Distribution Details Exist)
- Transfer Payment (Distribution Details Exist)

Transfer Payment (No Distribution Details Exist)

Payment transfer causes the payment and its payment segments to be canceled and a new payment to be created. Transferring a payment is typically only done if the original distribution was targeted at the wrong account but all tender information for the pay event is correct. Unlike the payment event transfer, transferring a single payment does not affect the pay event's tender information.

You must specify the following parameters in the **Payment Transfer** window to transfer a payment:

- **Account ID** is the account to which the payment should be transferred.
- **Cancel Reason** defines why you are performing the transfer.
- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match Type and Match Value are used for an account that belongs to an open item customer class or to restrict the distribution of the payment to a specific service agreement. Refer to Payment Event - Main for more information.

Copyright © 2026, Oracle and/or its affiliates.

- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information.

Clicking **OK** causes the following to take place:

- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- The payment is distributed amongst the new account's service agreements.
- If so designated, the new payment is frozen.

Transfer Payment (Distribution Details Exist)

Payment transfer causes the payment to be canceled and new payment(s) to be created using new payment event distribution rule(s). Transferring a payment is typically done when a payment event has multiple payments and one or more payments have been assigned to the wrong SA. Each of the incorrectly directed payments can be transferred to the correct SA(s) using the **Transfer via Distribution Rules** dialog.

Total Amount is the total payment amount to transfer from.

You must specify the following parameters in order to transfer a payment:

- **Distribution Rule** is the service agreement to which the payment should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment associated with the original account is canceled.
- The new set of distribution details is processed, creating new payment(s) for the transfer-to service agreement(s). The payment characteristics associated with the original payment are copied to the new payment.
- The new set of distribution details is added to the payment event. The original distribution details are kept.

> **Note**
>
> **Distribution rule entries can have a zero amount.** Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.
>
> **Determine Tender Account.** This dialog does not call the Determine Tender Account algorithm defined on distribution rule as this action does not rebuild the payment tender collection.

Copyright © 2026, Oracle and/or its affiliates.

Delete (A Payment)

The **Delete** button deletes a payment. This button is enabled if the payment is Incomplete, Error, or Freezable.

Payment Portal

The base package is shipped with a configured payment portal that can be used for payment processing. The portal supports the following functionality that is not supported in the base Payment Event - Add Dialog:

- Processing of payments that require authorization (i.e. credit card payments)
- Creating non CIS payments based on payment templates
- Processing payment tenders in alternate currencies

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Portals and Zones in the *Oracle Utilities Application Framework Administration Guide* for a description of portal and zone functionality.

- Credit Card Payments
- Non CIS Payments
- Alternate Currency Payments

Credit Card Payments

The payment portal provides the functionality needed for posting and authorizing customers' credit card charges in real-time. If the credit card authorization is successful, the payment will be added automatically (assuming you selected the appropriate payment freeze action via the portal).

> **NOT_SUPPORTED**
>
> Refer to How To for instructions describing how to add a payment that requires authorization.

Non CIS Payments

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the

Copyright © 2026, Oracle and/or its affiliates.

payment distribution, allowing non-CIS payments to be directly allocated against the appropriate distribution codes.

NOT_SUPPORTED

Refer to How To for instructions describing how to add a non-CIS payment using a payment template.

Alternate Currency Payments

The payment portal allows you to process a payment tender remitted in an alternate currency. The system enables conversion of the tendered amount into the account's currency.

NOT_SUPPORTED

Refer to How To for instructions describing how to process a payment in an alternate currency.

How To

The topics in this section describe how to perform common payment maintenance functions.

- How To Add A New Payment Event
- How To Cash A Check
- How To Allocate The Tender Amount To Multiple Accounts
- How To Print Receipts And Endorsements
- How To Cancel A Tender
- How To Transfer A Payment From One Account To Another
- How To Distribute A Payment To A Specific Service Agreement
- How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under)
- How To Add A Non-CIS Payment Using A Template
- How To Authorize A Credit Card Tender
- How To Reverse A Credit Card Tender
- How To Process An Alternate Currency Payment

How To Add A New Payment Event

There are several ways to add a new event:

Copyright © 2026, Oracle and/or its affiliates.

- You can use Payment Quick Add to quickly add one or more payment events. You'd use this approach to add simple payments where no manual intervention is required. By "simple payment" we mean:
  - The account is both the tendering account and the account whose debt is being relieved by the payment
  - The payment date is the current date
  - The payment should be distributed amongst the account's service agreements using standard distribution logic
- If applicable to your business practice, you can use Payment Event Quick Add to quickly add one or more payment events using distribution rules.
- You can use Payment Event Maintenance to add a payment event. You would use this approach if multiple forms of payment are remitted (e.g., cash and a check) or if there are multiple payors and/or payees linked to the payment event.

> **Note**
>
> By default the system opens the Payment Event - Add Dialog when you navigate to this page in add mode. If you have opted to always use the payment event distribution rules method as your default method, the Payment Event Quick Add (Single Payment Event) page appears instead. Refer to these pages for more information.

How To Cash A Check

If you allow customers to cash checks, you'll have a payment event with two tenders:

- The first tender contains information about the check.
- The second tender contains information about the cash refund (the tender amount is a negative amount equal to the amount of the check).

The interesting aspect of this payment event is that it has no payments because the sum of the tenders is zero.

We start this explanation in the middle of How To Add A New Payment Event.

In the **Payment Event Add** dialog, be sure to select Do Not Distribute before clicking **OK**.

- On the **Main** page, click **-** to remove the payment (cashing a check is a payment event with no payments).
- Transfer to the **Payment Event, Tenders** page.
- **Insert** a row in the tender scroll (click the + button) for the cash refund tender. In this row enter a tender type of "cash" and a tender amount of negative the check amount.
- Save the event (note there is nothing to distribute or freeze).

How To Allocate The Tender Amount To Multiple Accounts

Copyright © 2026, Oracle and/or its affiliates.

When you add a payment event, the system automatically creates a single payment for the account that remitted the funds. If someone is remitting funds for someone other than themselves, you must change/add payments. This section describes how to do this.

> **Note**
>
> This section assumes you chose to add the payment event using the Payment Event - Add Dialog. Refer to How To Add A New Payment Event for the complete list of options.

In the Payment Event - Add Dialog, be sure to select Do Not Distribute before clicking **OK**.

On the payment event Main page, do the following:

- If the remitting account shouldn't have received any part of the payment, **Remove** it from the payments grid by clicking the - button. Alternatively, you can just change the account id to reflect the recipient.
- If multiple accounts receive the remitted funds, **Insert** one row in the payments grid (click the + button) for each additional account.
- When the payment event is balanced, **Save** it. Then **Distribute** and **Freeze** the payment(s). Refer to Payment Event Actions for more information on these action buttons.

How To Print Receipts And Endorsements

The system can be configured to allow you to endorse checks and print receipts using special cashiering station printers. This section describes the print options available.

> **Note**
>
> These options are only available if they have been installed in your system. Installing these options are a delivery and installation issue and are not within the domain of this document. For information on how to enable these print options, refer to:
>
> - The Customer Cloud Service Receipt Printing section of the Cloud Services Implementation Guide. Or,
> - The Receipt Printing section of the Optional Products Installation Guide, for on-premise implementations.

The print functions are available from the Payment Event, Payment Quick Add and Payment Event Quick Add transactions. Refer to those sections for more information.

The print dialog functions are described below.

Description of Page

Use the **Endorse** button to endorse checks using the cashier station printer. This is only enabled if at least one tender type is not like cash and does not generate an automatic payment. If there are multiple checks, use the scroll buttons to select them. Feed the appropriate check in the printer and an endorsement message is printed on the back.

Copyright © 2026, Oracle and/or its affiliates.

Use the **Receipt** button to print a receipt using the cashier station printer. Choose a **Short** or **Long** option. Use the **Duplicate** button to print a duplicate receipt. This prints the receipt with a "Duplicate" label.

Use the **Stub** button to print special information in bill stub format (e.g. account, name, amount). Feed a slip of paper into the printer and the special stub information will be printed.

> **Note**
>
> The various text strings that are printed on the receipt and the endorsement are defined on the Payment Receipt Message and Endorse Message messages on the Point Of Sale (POS) Printer Integration master configuration. Refer to the master configuration's embedded help for more details.
>
> The Stub function is available only on the Payment Event page.
>
> Refer to the implementation / installation guide mentioned above, for details on the print BPA script samples provided in base and for sample printouts using these scripts.

How To Cancel A Tender

If a tender is no longer valid (e.g., a check bounces), the tender must be canceled. The following steps explain how to do this.

> **Warning**
>
> These steps only make sense in the context of the page used to maintain tenders. Refer to Payment Event - Tenders for the details.

You navigate through the pages as follows:

1. Open the payment event - tenders page for the tender in question. Proceed to step 2.

2. Click **Cancel**. Proceed to step 3.

3. Before the system cancels the tender, you must specify the cancel reason. The tender cancel window opens.

In the tender cancel window, select a **Cancel Reason** to describe why the tender is being canceled. The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank** and **Bank Account Key** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

If the cancellation reason you supply is indicated as being "non-sufficient funds", the system will generate an adjustment to levy a NSF charge. Refer to SA Type - Main Information for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **When non-canceled tenders are still linked to the event.** If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.
>
> **Cancellation after cash back.** If a customer tenders a check for $100, but only owes $80, the system will recommend returning $20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not "like cash", a second tender is created for -$20 (the first tender is for $100). If the check subsequently bounces, both tenders must be cancelled.

How To Transfer A Payment From One Account To Another

If you need to transfer a payment amount from one account to another, you can use the **Transfer** button either on the payment event page or the payment page.

> **NOT_SUPPORTED**
>
> Refer to Transferring A Payment for a description of when to use each option.

How To Distribute A Payment To A Specific Service Agreement

When you click **Distribute**, the system uses the payment distribution algorithm defined on the account's customer class to allocate the payment amongst the account's service agreements. In this section, we describe how to override this distribution (e.g., when you need to allocate a payment to specific service agreement(s)).

We start this explanation in the middle of How To Add A New Payment Event.

In the **Payment Event Add** dialog, be sure to select Do Not Distribute before clicking **OK**.

- **Save** the payment event (this also saves the payment and tender information).
- Use the **Payment Amount** context menu to transfer to the **Manual Distribution** page.
- Fill in the desired **Distributed Amt** for each service agreement and save the payment.
- Click **Freeze** to freeze the payment when the amount distributed is equal to the payment amount.

How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under)

In order to balance a tender control that is out-of-balance, you must create a tender for an account associated with your company.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The over / under account.** Your organization must set up a company-use account with a service agreement whose SA type references the over/under distribution code. This account should be linked to the person ID associated with company usage.

- If you have more money in the drawer than you have tenders, then you're "over". In this situation, you need to record a payment for the "over" amount and the tender type configured for cash drawer over/under amounts. This will cause cash to be debited by the over amount, and the expense account associated with the account's service agreement to be credited.
- If you have less money in the drawer than you have tenders, then you're "under". In this situation, you need to record a negative payment for the "under" amount and the tender type configured for cash drawer over/under amounts. This will cause cash to be credited by the over amount, and the expense account associated with the account's service agreement to be debited.

> **Important**
>
> A separate tender type is used for recording cash drawer over/under amounts. Once the over/under amount has been determined, you must enter a new tender using your over/under tender type for the company-use account. Check out the following table for an example:

Tender Type	Starting Balance	Tenders Received	Turn Ins	Actual Ending Balance (Entered by Cashier)	Expected Ending Balance	Over/Under Amount
Cash	$150.50	$5,000	$4,000	$1151	$1150.50	Over $0.50
Check	-	$1,000	$750	$249	$250	Under $1
Over/Under	-	$-0.50	-	-	$-0.50	Over $0.50

In this example, you'd have to create a new tender for the company-use account for the amount that will be distributed to the company-use account's service agreement (debit over/under expense, credit cash):

- Tender type: Over/Under, Tender amount: -$0.50

> **Note**
>
> **You must reopen the tender control.** Before you can add an over/under tender to a tender control, you must re open the tender control (tenders may only be added to open tender controls).

Copyright © 2026, Oracle and/or its affiliates.

How To Add A Non-CIS Payment Using A Template

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the payment distribution and allowing non-CIS payments to be directly allocated against distribution codes. The following steps explain how to do this.

You navigate through the pages as follows:

- Open **Menu**, and then **Financial**, and then **Payment Portal.**
- The **Add Non CIS Payment Zone** shows a list of all active payment templates. Use the **Add** button to add a non-CIS payment for a specific payment template.
- Enter the payment and tender details (**Payment Amount**, **Payment Date** defaults to the current date, **Tender Source** is used to find an open tender control when adding the tender).

> **Note**
>
> **Authorizing Tenders.** If the **Tender Type** selected requires authorization (i.e. credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc.). Only one tender that requires authorization can exist for a single payment event.

> **NOT_SUPPORTED**
>
> **For information** about tender types requiring authorization, refer to How To Authorize A Credit Card Tender.

- Enter the non-CIS information for the customer remitting the payment (**Customer Name**, **Agreement Number** and **Comments** to describe anything unusual about the non CIS payment).

> **Note**
>
> **Payor Account ID.** The Non CIS Account Id specified on the payment template is used as the tendering account. If the account has exceeded your cash-only threshold and the **Tender Type** isn't marked as "like cash", a warning appears advising of such.
>
> **Including a CIS payment.** If a portion of the tender is for a CIS payment, ensure that the **Add CIS Payment** checkbox is selected. The system will process the non-CIS payment, and then navigate the user to the Payment Event page where the CIS payment can be included.

- The distribution details are defaulted from the payment template but may be overridden. You can add additional distribution lines. Enter the payment amount and Reference Number for each applicable distribution line.

Copyright © 2026, Oracle and/or its affiliates.

- When you click the **OK** button, the system creates the payment event and tender(s). A single payment is created for each non-zero amount entry in the distribution details list. The payment template is captured as a characteristic on the tender. The corresponding distribution code, GL account and reference number are captured as characteristics on each payment. When the financial transactions are created for these payments, the distribution code (and GL account) is retrieved from the payment characteristic collection.

How To Authorize A Credit Card Tender

You can use the Payment Portal to add a payment event for a tender that requires authorization. The following steps explain how to do this.

If your organization is using the base tender authorization functionality, you can navigate through the pages as follows:

- Open **Menu**, then **Financial**, then **Payment Portal.**
- The **Add CIS Payment** zone only expands if there's an account in context. To expand this zone, either search for an account using Control Central or use the Tender Search available on the **Payment Portal**.
- Use the **Add Payment** button to add a payment for the account displayed in the zone.
- The information displayed is the same as that on the Payment Event - Add Dialog, except for the following:
  - Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button. The dropdown contains the following values - Distribute and Freeze if OK, Manual Distribution and Manual Distribution - Multiple Payments

> **Note**
>
> The **Distribute and Freeze if OK** and **Manual Distribution** options are also present on the **Distribute Action** on the payment event add dialog screen. The third option, **Manual Distribution - Multiple Payments**, is special for credit card authorization processing. This option is similar to the **Manual Distribution** option except that the user is navigated to the **Payment Event** maintenance page after authorization has taken place.

- If the **Tender Type** selected requires authorization (e.g., credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc.). This information is displayed in the **Credit Card Information** section.

> **Note**
>
> **Authorizing Tenders.** Only one tender that requires authorization can exist for a single payment event.

- The **Authorization Override** checkbox is only displayed if the tender type selected requires authorization. Use the **Authorization Override** checkbox to indicate whether or not tender authorization should take place. If **Authorization Override** is checked, an

Copyright © 2026, Oracle and/or its affiliates.

authorization code must be entered and the system bypasses the authorization web service call.

- Click **OK** to save the payment and navigate to the appropriate page.
- Assuming the credit card is accepted, the authorization code returned by the credit card authorization provider is captured as a tender characteristic.
- If the credit card is declined, the tender will not be added.

How To Reverse A Credit Card Tender

The steps required to reverse a credit card tender are the same as the steps for How To Cancel A Tender.

The reversal code from the credit card authorization service is stored as a tender characteristic.

> **NOT_SUPPORTED**
>
> For more information on authorizing credit card payments, refer to the following business objects: Tender Type - Credit Card with Authorization, CyberSource - Credit Card Authorization and CyberSource - Credit Card Reversal.

How To Process An Alternate Currency Payment

You can use the Payment Portal to add a payment event for a tender(s) in an alternate currency, i.e. a currency that differs from the account's currency. The following steps explain how to do this.

- Open **Menu**, and then **Financial**, and then **Payment Portal.**
- The **Add CIS Payment Z one** only expands if there's an account in context. To expand this zone, either search for an account using Control Central or use the Tender Search available on the **Payment Portal**.
- Use the **Add Payment** button to add a payment for the account displayed in the zone.
- The information displayed is the same as that on the Payment Event - Add Dialog, except for the following:
  - If your organization accepts payment tenders in alternate currencies, the **Currency Converter** button appears in the tender list
- Use the **Currency Converter**button to enter the **Tendered Amount** and **Tendered Currency** in the alternate currency and click **OK**. The system will convert the alternate currency amount to the account's currency.
- Click **OK to save the payment and navigate to the appropriate page**.
- When you click the **OK** button, the system creates the payment event and tender(s). Each tender is linked to an appropriate tender control based on the currency of the tender.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Alternate Currency Payments for more information.

Financial Transactions On A Payment

Open **Menu**, and then **Financial Query**, and then **Financial Transactions On A Payment** to view the financial transactions on a payment.

Note

You can also open this page using the go to buttons that prefix the financial transaction summaries on Payment - Main.

Description of page

Payment Id is the system-assigned unique identifier of the payment.

Account ID is the payment's account.

The area beneath **Account ID** provides you with options that control which financial transactions appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
  - Address. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code.
  - All. Use this option if you do not wish to restrict financial transactions based on service agreement attributes.
  - Geographic Type. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.
  - SA ID. Use this option to restrict financial transactions to those of a specific service agreement.
  - SA Type. Use this option to restrict financial transactions to those whose service agreements are linked to a given **CIS Division** and **SA Type**.
- Use **Match Event Status Filter** to restrict the financial transactions based on the status of their match event. This filter only appears if the payment's account is an open-item customer. The following options are available:
  - All. This option shows all financial transactions.
  - Balanced. This option shows all financial transactions whose match event is balanced.
  - Disputed. This option shows all financial transactions whose match event is disputed.
  - Unbalanced. This option shows all financial transactions whose match event is unbalanced.

Copyright © 2026, Oracle and/or its affiliates.

– Unmatched. This option shows all financial transactions that are not linked to a match event.

> **Warning**
>
> Don't forget to click the search button after changing the filters or after selecting a new Payment Id.

The grid that follows contains the financial transactions that match your search criteria. The following information is displayed:

- **Match Event Status** shows the status of the financial transaction's match event. This column only appears if the account is an open-item customer.
- **FT Type** displays the type of financial transaction. Click on the hyperlink to transfer to Financial Transaction - Main. On this page, you can change certain aspects of the FT in question.
- **Accounting Date** is the date the system uses to determine the financial transaction's accounting period in your general ledger.
- **Current Amount** contains the financial transaction's effect on the service agreement's current balance.
- **Payoff Amount** contains the financial transaction's effect on the service agreement's payoff balance. The **Payoff Amount** will be dim if it equals the **Current Amount**.
- **Show on Bill** indicates if information about the financial transaction appears on the customer's bill.
- **SA Information** contains a summary of the respective service agreement.
- **Financial Transaction ID** is the system-assigned unique identifier of the financial transaction.

At the bottom of the page is a summary of the financial transactions that match the search criteria.

Payment History

The payment history page shows all payments that have been distributed to an account's service agreements. Open this page using **Menu**, and then **Financial Query**, and then **Account Payment History.**

Description of Page

One row is displayed for every payment that has been distributed to an account's service agreements. The payment date, amount, payment status and the tender source associated with the tender's tender control are displayed for each payment.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Tender Source** typically contains the description of the cash drawer in which the payment was made or the remittance processor that processed the payment. **Tender source** will be blank for automatic payments until they are interfaced to the financial institution. Refer to Downloading Automatic Payments for more information about interfacing automatic payments.

If you need to see more detailed information about the payment, click the Go To button to transfer to the payment event page.

- Payment Tender History

Payment Tender History

The **Payment Tender History** page shows a list of all tenders linked to an open tender control for the current user. Open this page using **Menu**, and then **Financial Query**, and then **Payment Tender History.**

Description of Page

One row is displayed for every tender that is linked to an open tender control to which the current user has access. The freeze date/time, payment date, pay tender ID, payor account, tender amount, tender type and tender status are displayed for each tender.

Payment / Tender Search

This page allows you to look for payments and/or tenders using a combination of search criteria. Open this page using **Menu**, and then **Financial Query**, and then **Payment / Tender Search.**

Description of Page

The top half of the page is where you enter the criteria used to search for payments and tenders.

> **Note**
>
> **Multiple search criteria may be specified.** You can search for payments and tenders using a combination of search criteria. For example, if you enter both a **Payment Account** name of Brandon and a **Payment Amount** between 150 and 170; only those payments for customers named Brandon whose amount is between 150 and 170 will be displayed.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times. For example, if you know the payments you're looking for have a payment date sometime in January 2003 and the customer's name is Brazil,John, then enter both of these criteria. If you also know the payment amount is between 150 and 170, enter these values too.

The following table describes each of the different search methods.

Search Method	Description
Search for	Use this field to define if you're searching for Payments, Tenders or both Payments and Tenders. The value entered in this field controls which of the remaining search methods are enabled.
Distribution Rule	Use this field if you're searching for a tender or a payment and know the distribution rule and value used to created it. **Note.** This section appears only if you have configured your system to allow the payment event distribution rules method to be used.
Payment Account	Use this field if you're searching for a payment and know the account ID or the name of any person linked to the account whose debt is relieved by the payment. - If you know the payment's account ID, first choose Account in the adjacent dropdown and then and enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the account, first choose Person Name in the adjacent dropdown and then enter the person's name. You can enter all or part of the person's name (the more you enter, the faster the search will be). The name search is not case sensitive. A search method of Person Name defaults. If you leave the person name or ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Tender.

Copyright © 2026, Oracle and/or its affiliates.

Payment Amount	Use this field if you're searching for a payment and know its amount. - If you know the exact amount, first choose Equal To in the adjacent dropdown and then enter the amount. - If you know the payment amount is between a given range of values, first choose Between and then enter the range of values. A search method of Between defaults. If you leave the amount(s) blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Tender.
Payor Account	Use this field if you're searching for a tender and know the account ID or the name of any person linked to the tendering account. - If you know the tender's account ID, first choose Account in the adjacent dropdown and then and enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the tendering account, first choose Person Name in the adjacent dropdown and then enter the person's name. You can enter all or part of the person's name (the more you enter, the faster the search will be). The name search is not case sensitive. A search method of Person Name defaults. If you leave the person name or ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Payment.
Tender Amount	Use this field if you're searching for a tender and know its amount. - If you know the exact amount, first choose Equal To in the adjacent dropdown and then enter the amount. - If you know the tender amount is between a given range of values, first choose Between and then enter the range of values. A search method of Between defaults. If you leave the amount(s) blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Payment.
Tender Source	If you're searching for a tender and you know the source of the tender (e.g., the lockbox, cashier, etc.), enter the tender source code. A search method of Equal To defaults. If you leave the tender source blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Payment.

Copyright © 2026, Oracle and/or its affiliates.

MICR ID	Use this field if you're searching for a tender and know its MICR (magnetic ink character recognition) ID. - If you know the entire MICR ID, first choose Equal To in the adjacent dropdown and then and enter the entire MICR ID. - If you know the first X characters of the MICR ID, first choose Like in the adjacent dropdown and then and enter the entire MICR ID. A search method of Like defaults. If you leave the MICR ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of Payment.
Payment Date	Use this field if you're searching for a tender or a payment and know its date. - If you know the exact date, first choose Equal To in the adjacent dropdown and then enter the date. - If you know the date is between a given range of values, first choose Between and then enter the date range. A search method of Between defaults. If you leave the date(s) blank, the system ignores this search method.

The system shows the total number of payments / tenders that satisfy your search results immediately below the grid.

The first group of payments / tenders is displayed in the grid at the bottom of the page. Different columns appear in the grid depending on the value of **Search for**.

- If you use a **Search for** of Payment, only payment-oriented columns appear in the grid.
- If you use a **Search for** of Tender, only tender-oriented columns appear in the grid.
- If you use a **Search for** of Payment and Tender, both payment-oriented and tender-oriented columns appear in the grid.

> **Warning**
>
> If you use the Payment and Tender search method, and the resultant tenders / payments are linked to payment events that have multiple payments or tenders, multiple rows may be displayed for the payment event's tenders and payments. For example, if a tender with multiple payments is selected, a separate row will be displayed for every payment that matches your payment search criteria. If you don't specify any payment search criteria, a row will be displayed for every payment linked to the tender.

Payment Date This contains the date of the payment event associated with the payment / tender. This column appears regardless of the value of **Search for**.

Copyright © 2026, Oracle and/or its affiliates.

Payment Account Info This contains a concatenation of important information about the account whose debt was relieved by the payment. This column does not appear if you use a **Search for** of Tender .

Payment Amount This contains the amount of the payment. This column does not appear if you use a **Search for** of Tender.

Payment Status This contains the status of the payment. Refer to Payment Lifecycle for more information. This column does not appear if you use a **Search for** of Tender.

Payment ID This contains the unique identifier of the payment. This column does not appear if you use a **Search for** of Tender.

Payor Account Info This contains a concatenation of important information about the account who tendered the payment. This column does not appear if you use a **Search for** of Payment .

Tender Amount This contains the amount of the tender. This column does not appear if you use a **Search for** of Payment.

Tender Source This contains the source of the tender of the tender. This column does not appear if you use a **Search for** of Payment.

MICR ID This contains the MICR (magnetic ink character recognition) ID of the tender. This column does not appear if you use a **Search for** of Payment .

> **Note**
>
> The system supports encryption for MICR ID. If your implementation has configured the system to encrypt MICR ID, the data will be displayed with a masked value; such as, *******.Refer to Encrypting Sensitive Data for more information.

Tender Status This contains the status of the tender. Refer to Tender Lifecycle for more information. This column does not appear if you use a **Search for** of Payment.

Pay Tender ID This contains the unique identifier of the tender. This column does not appear if you use a **Search for** of Payment.

Pay Event ID This contains the unique identifier of the payment event.

Maintaining Deposit Controls

Deposit controls exist to give you administrative control over your cash drawers (and all other tender sources) and the subsequent deposit of funds at banks.

> **NOT_SUPPORTED**
>
> Refer to Tender Management and Workstation Cashiering for background information.

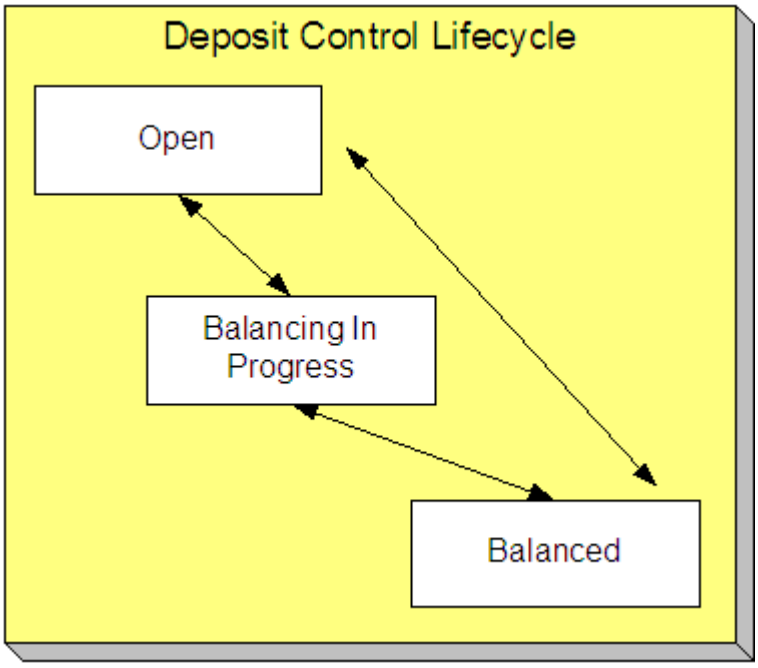
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The system creates most deposit controls behind-the-scenes.** Most deposit controls are created by the system when it processes tenders from your remittance processor and lock boxes. You should only have to access the deposit control pages if you record payments in cash drawers. For information about how the system creates deposit controls, refer to Managing Payments Interfaced From External Sources. Also note that the automatic payment activation process also creates tender and deposit controls. Refer to Activating Automatic Payments for more information.

- The Lifecycle Of A Deposit Control
- Deposit Control - Main
- Deposit Control - Tender Control
- Deposit Control - Tender Deposit
- Deposit Control - Turn Ins
- Deposit Control - Characteristics

The Lifecycle Of A Deposit Control

The following diagram shows the possible lifecycle of a deposit control.



Open A deposit control is initially created in the Open state. While in this state, you may add new deposits to it, change the deposit amount on its deposits, and transfer tender controls into and out of it.

Balancing In Progress You change a deposit control's status to Balancing In Progress when you're ready to balance its contents. While in this state, you can change the deposit amount on

Copyright © 2026, Oracle and/or its affiliates.

its deposits and transfer tender controls out of it. If you need to add new deposits to it or transfer tender controls into it you must return it to the Open state.

Balanced You change a deposit control's status to Balanced when the sum of its tender controls is consistent with the total of its deposits. While in this state, you cannot modify its deposits or its tender controls. If you need to make modifications, you must return it to the Open or Balancing In Progress state.

> **Note**
>
> **Background processes and state transition.** When payments are interfaced from external sources, the system automatically creates a deposit control and links a tender control to it. When all payments have been successfully loaded, the system changes the state of the respective deposit control to Balanced.

Deposit Control - Main

The Main page contains core deposit control information. Open this page using **Menu**, and then **Financial**, and then **Deposit Control**, and then **Search.**

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

User is the person who created the deposit control.

Create Date/Time is the date and time the deposit control was created.

Tender Source Type is the type of tender control that has been linked to the deposit control. Valid values are: Ad hoc, Auto Pay, Online Cashiering and Lockbox. The system uses this information to prevent tender controls from different sources from being included under the same deposit control. In other words, you can't mix automatic payment, cashiering and lockbox tenders under the same deposit control.

Currency Code is the currency in which the deposit control's tenders are denominated.

> **Note**
>
> **Default note.** The currency code defaults from the installation record.

The summary information that follows contains a summary of the starting balance and the tenders that are linked to the tender control.

- **Starting Balance** is the sum of the starting balances from all tender controls linked to the deposit control.
- **Total Tenders Amount** is the sum of tenders from all tender controls linked to the deposit control.

Copyright © 2026, Oracle and/or its affiliates.

- **Total Tender Controls** is the number and amount of tender controls linked to the deposit control.
- **Total Tender Deposits** is the number and amount of tender deposits linked to the deposit control.
- **Expected Ending Balance** is the **Total Tender Control** minus **Total Tender Deposits**.
- **Ending Balance** is the actual amount of money in the tender control minus the deposit tender amount. This amount must equal the **Expected Ending Balance** before the tender control can be marked as Balanced.
- **Outstanding Over/Under** is **Ending Balance** minus **Expected Ending Balance**.

Deposit Control Status shows the status of the deposit control. Valid values are Open, Balanced, and Balancing In Progress.

> **NOT_SUPPORTED**
>
> For more information, refer to The Lifecycle Of A Deposit Control.

The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to Balanced.

Use **Comments** to describing anything unusual about the deposit control.

Deposit Control - Tender Control

The Tender Control page contains a row for every tender control linked to the deposit control. Open this page using **Menu**, and then **Financial**, and then **Deposit Control**, and then **Search** and navigate to the **Tender Control** page.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The grid that follows contains a row for every tender control linked to the deposit control. No information about the tender controls may be modified on this page. To view or modify tender control information, click the Go To button.

Deposit Control - Tender Deposit

The Tender Deposit page contains a row for every tender deposit linked to the deposit control. Open this page using **Menu**, and then **Financial**, and then **Deposit Control**, and then **Search** and navigate to the **Tender Deposit** page.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

Copyright © 2026, Oracle and/or its affiliates.

The **Tender Deposit** scroll that follows contains one row for every financial institution into which the tenders will be deposited. To insert a new row, click the + button and fill in the following fields:

- **Tender Deposit ID** is maintained by the system. When a deposit is being created, there is no ID number displayed. Once a deposit is entered and saved, the system generates an ID and displays it here.
- **Deposit Amount** contains the amount to be deposited at the bank. **Currency Code** is the currency in which the deposit is denominated.
- **Reference ID** contains the ID of the deposit (if any).
- Use **Bank** and **Bank Account** to define where the tenders will be deposited.

Deposit Control - Turn Ins

The Turn Ins page contains a row for every turn-in event linked to the deposit control. Open this page using **Menu**, and then **Financial**, and then **Deposit Control**, and then **Search** and navigate to the **Turn Ins** page.

> **NOT_SUPPORTED**
>
> Refer to Turn Ins **for background information**.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The turn-ins grid contains one row for every turn-in event associated with the deposit control. A turn-in event is created when moneys originally deposited in respect of a tender control are turned in to the tender control's deposit control. Turn-in events are created and maintained on the Tender Control - Turn Ins page. Refer to Turn Ins for background information. The following information is displayed in the grid:

- **Turn In Status** defines if it is a new turn in Awaiting approval or has been Approved by the operator who is responsible for the Deposit Control. Once the turn-in is Approved, this field becomes display-only.
- **Tender Type** is the type of tender that has been turned in. This is a display-only field.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This is a display-only field.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This is a display-only field.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Deposit Control - Characteristics

Copyright © 2026, Oracle and/or its affiliates.

To update deposit control characteristics, open **Menu**, and then **Financial**, and then **Deposit Control**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The **Characteristics** collection contains information that describes miscellaneous information about the deposit control.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Maintaining Tender Controls

Tender controls exist to give you administrative control over your cash drawers (and all other tender sources).

NOT_SUPPORTED

Refer to Managing Your Cash Drawers for more information.

Note

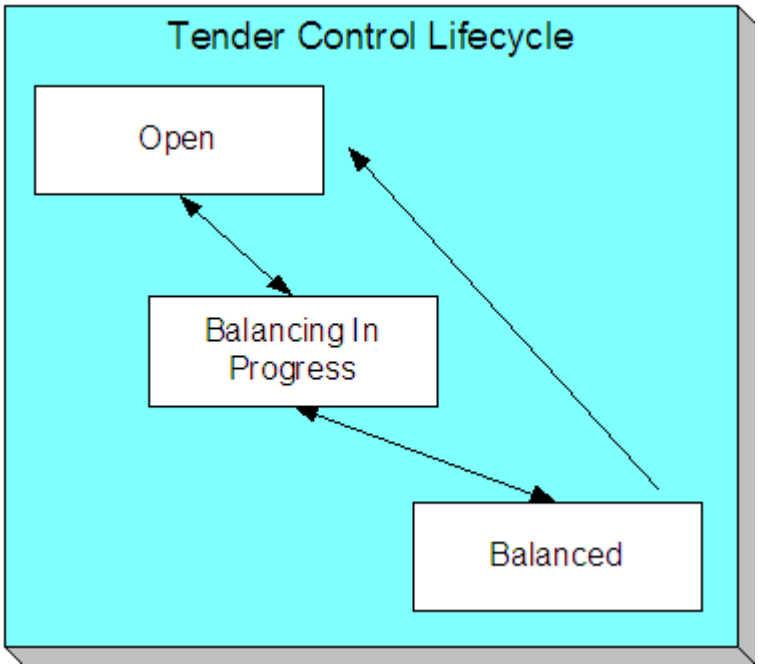
The system creates most tender controls behind-the-scenes. Most tender controls are created by the system when it processes tenders from the remittance processors and lock boxes. You should only have to access the tender control pages if you record payments in cash drawers. For information about how the system creates tender controls, refer to Managing Payments Interfaced From External Sources. Also note that the automatic payment activation process also creates tender and deposit controls. Refer to Activating Automatic Payments for more information.

- The Lifecycle Of A Tender Control
- Tender Control - Main
- Tender Control - Tenders
- Tender Control - Turn Ins
- Tender Control - Exceptions
- Tender Control - Characteristics

Copyright © 2026, Oracle and/or its affiliates.

The Lifecycle Of A Tender Control

The following diagram shows the possible lifecycle of a tender control.



Open A tender control is initially created in the Open state. While in this state, you may add new tenders to it, change the tender amount on its tenders, and transfer tenders into and out of it.

Balancing In Progress You change a tender control's status to Balancing In Progress when you're ready to balance its contents. While in this state, you can change the tender amount on its tenders and transfer tenders out of it. If you need to add new tenders to it or transfer tenders into it, you must return it to the Open state.

Balanced You change a tender control's status to Balanced when the sum of its tenders is consistent with the ending balance in the drawer. While in this state, you cannot modify it or its tenders. If you need to make modifications, you must return it to the Open or Balancing In Progress state.

If the tender control is part of a Balanced deposit control, you may not change its status.

> **Note**
>
> **Background processes and state transition.** When payments are interfaced from external sources, the system automatically creates a tender control and links tenders to it (one for each payment interfaced). When all payments have been successfully loaded, the system changes the state of the respective tender control to Balanced.

Copyright © 2026, Oracle and/or its affiliates.

Tender Control - Main

The Main page contains core tender control information. This page is used to balance the tender control.

Open this page using **Menu**, and then **Financial**, and then **Tender Control**, and then **Search.**

> **Note**
>
> **Searching For Tender Controls.** When you use the **Creation Date** to search for tender controls, the system returns tender controls that are created on or before the date you specify. If the number of records returned exceeds the search limit (i.e. 300 records), only the records that do not exceed search limit are displayed. Therefore, you may need to specify an earlier creation date to find the record you are looking for. If you do not specify a creation date and search based on other criteria, the system uses the most recent date.

> **NOT_SUPPORTED**
>
> Refer to Managing Your Cash Drawers, Turn Ins and Balancing By Tender Type for more information.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

Tender Source is the source of the tenders (e.g., cash drawer 22, lockbox 1 at Bank of America).

Deposit Control ID is the system-assigned, unique identifier of the deposit control that the tender control is part of. The deposit control's **Currency Code** is displayed below.

> **NOT_SUPPORTED**
>
> For more information about deposit controls and tender controls, refer to Managing Your Cash Drawers.

Turn on **All Users** if any user may insert tenders into this tender control. Turn this switch off and specify the appropriate **User ID** if only a single user may insert tenders into this tender control.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When the **All Users** switch is off, only the associated user can update the tender control, unless the system is configured to allow users to update tender controls owned by other users. This configuration is the **Allow Change Tender Control User** option in Financial Transaction feature configuration. Its value would be set to **Y**.

Create Date/Time is the date and time the tender control was created.

The **Starting Balance** of the tender control is defaulted from the tender source. This may be overridden while the **Tender Control Status** is Open or Balancing in Progress.

The **Tender Control Status** shows the status of the tender control. The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to Balanced.

> **NOT_SUPPORTED**
>
> For more information, refer to The Lifecycle Of A Tender Control.

The **Balance** button is enabled when the **Tender Control Status** is Balancing in Progress. When this button is clicked, the system checks the following:

- All Turn-Ins have been Approved.
- The sum of all **Ending Balances** equals the sum of all **Expected Ending Bal**ances.

If the above conditions are true, the status of the tender control is changed to Balanced and all modifiable fields become protected.

The tenders grid contains a summary of the tenders linked to the tender control. One row is displayed for each tender type. Each row contains the number of tenders of this type and their total amount. In order to Balance the tender control, you must enter an appropriate **Ending Balance** for each **Tender Type**. Refer to Balancing By Tender Type for more information. The following information appears in the grid:

- **Tender Type** represents the type of tender (e.g., cash, credit card). This is a display-only field.
- **Nbr of Tenders** contains the total number of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control.
- **Total Tenders** contains the total amount of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control. Navigate to the **Tenders** page to see the individual tenders.
- **Turn In Amount** contains the total amount of turn-ins of this type. This is a display-only field and is calculated by the system by accumulating the turn-ins linked to the tender control. Navigate to the **Turn Ins** pages to see the individual turn ins.
- **Starting Balance** contains the starting balance of this type of turn in. This field is only populated on the row defined as the Starting Balance tender type on the Installation Record. This is a display-only field and is equal to the tender control's **Starting Balance**.
- **Expected Ending Bal** is the **Starting Balance** plus **Total Tenders** minus **Turn In Amount** for this type of tender.

Copyright © 2026, Oracle and/or its affiliates.

- **Ending Bal** is the actual amount of money of this tender type. The cashier must enter this field in order to balance the tender control. This field is protected unless the **Status** of the tender control is Balancing In Progress. This amount must equal the Expected Ending Balance before the tender control can be marked as Balanced. Refer to How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under) for more information.
- **Outstanding Over/Under** is the difference between **Expected Ending Bal** and **Ending Bal**. This is a display-only field.

The summary information beneath the tender type grid contains a summary of the tenders that are linked to the tender control.

- **Total Tender Amount** is the total amount of all tenders linked to the tender control.
- **Ending Balance** is the actual amount of money in the tender control (as defined in the above grid).
- **Expected Ending Balance** is the expected amount of money in the tender control (as defined in the above grid).
- **Outstanding Over/Under** is Ending Balance minus Expected Ending Balance. If this amount is zero, the field is not displayed.

If the tender controls were created for a background process, such as the automatic payment extract, the following information is displayed:

- **Batch Code** is the batch code for which the tender control was created. This field only appears if it contains a value.
- **Batch Number** is the batch number for which the tender control was created. This field only appears if it contains a value.

Use **Comments** to describe anything unusual about the tender control (e.g., to explain why a large over/under amount was created).

Tender Control - Tenders

The Tenders page contains a row for every tender linked to the tender control. Open this page using **Menu**, and then **Financial**, and then **Tender Control**, and then **Search** and navigate to the **Tenders** page.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

This grid at the top of the page contains a row for every tender linked to the tender control. When first displayed, the grid is ordered by the Create Date/Time column. No information about the tenders may be modified. To view or modify tender information, click the Go To button.

The grid at the bottom of the page contains a summary by tender type of all tenders linked to the tender control.

Tender Control - Turn Ins

Copyright © 2026, Oracle and/or its affiliates.

Every time you turn in funds to the deposit control that is associated with your tender control, you create a turn-in event using this page. Open this page using **Menu**, and then **Financial**, and then **Tender Control**, and then **Search** and navigate to the **Turn Ins** page.

> **NOT_SUPPORTED**
>
> Refer to Managing Your Cash Drawers and Turn Ins for more information.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

The turn-ins grid contains one row for every turn-in event associated with the tender control. A turn-in event is created when moneys are physically transferred to the deposit control associated with the tender control. Turn-in events are approved on the Deposit Control - Turn Ins page. The following information is displayed in the grid:

- **Turn In Status** defines if the turn-in is Awaiting Approval or has been Approved by the operator who is responsible for the Deposit Control. When a turn-in is first created, its status is Awaiting Approval and the following fields should be defined. Once the turn-in is Approved by the user responsible for the deposit control associated with the tender control, all of the following fields become display-only and the turn-in event cannot be deleted.
- **Tender Type** is the type of tender that has been turned in. This field may only be modified when the **Turn In Status** is Awaiting Approval.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This field may only be modified when the **Turn In Status** is Awaiting Approval.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This field may only be modified when the **Turn In Status** is Awaiting Approval.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Tender Control - Exceptions

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events. Open this page using **Menu**, and then **Financial**, and then **Tender Control**, and then **Search** and navigate to the **Exceptions** page.

> **NOT_SUPPORTED**
>
> Refer to Exceptions for more information.

Description of Page

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events.

Copyright © 2026, Oracle and/or its affiliates.

Tender Control - Characteristics

To update tender control characteristics, open **Menu**, and then **Financial**, and then **Tender Control**, and then **Search** and navigate to the **Characteristics** page.

> **Note**
>
> If your tender control is balanced, you cannot update tender control characteristics.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the tender control.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Interfacing Payments From External Sources

Most payments are NOT added by an operator using the Payment Event page. Rather, they are interfaced from an external source (e.g., a lock box or a remittance processor).

The base-package provides two interfaces to upload payments, each based on a different method of creating payment events.

> **NOT_SUPPORTED**
>
> Refer to Distributing A Payment Event for more information about how payment event distribution is handled in the system.

The topics in this section describe how these payment interfaces work.

- Interfacing Payments
- Interfacing Payments Using Distribution Rules

Interfacing Payments

The topics in this section describe how the payment interface using the system default method of creating payment events works.

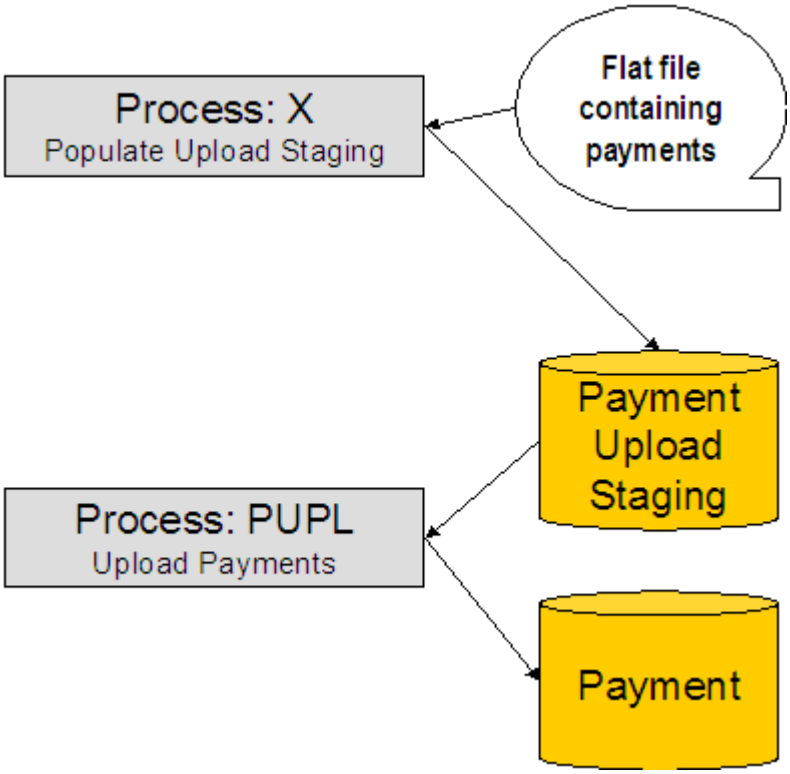
- Populating The Payment Upload Staging Records

Copyright © 2026, Oracle and/or its affiliates.

- PYUP-PRG - Purge Payment Upload Objects
- Maintaining Deposit Control Staging
- Payment Upload Staging
- Payment Upload Exception

Populating The Payment Upload Staging Records

The following diagram illustrates the processes involved in the uploading of payment into the system.



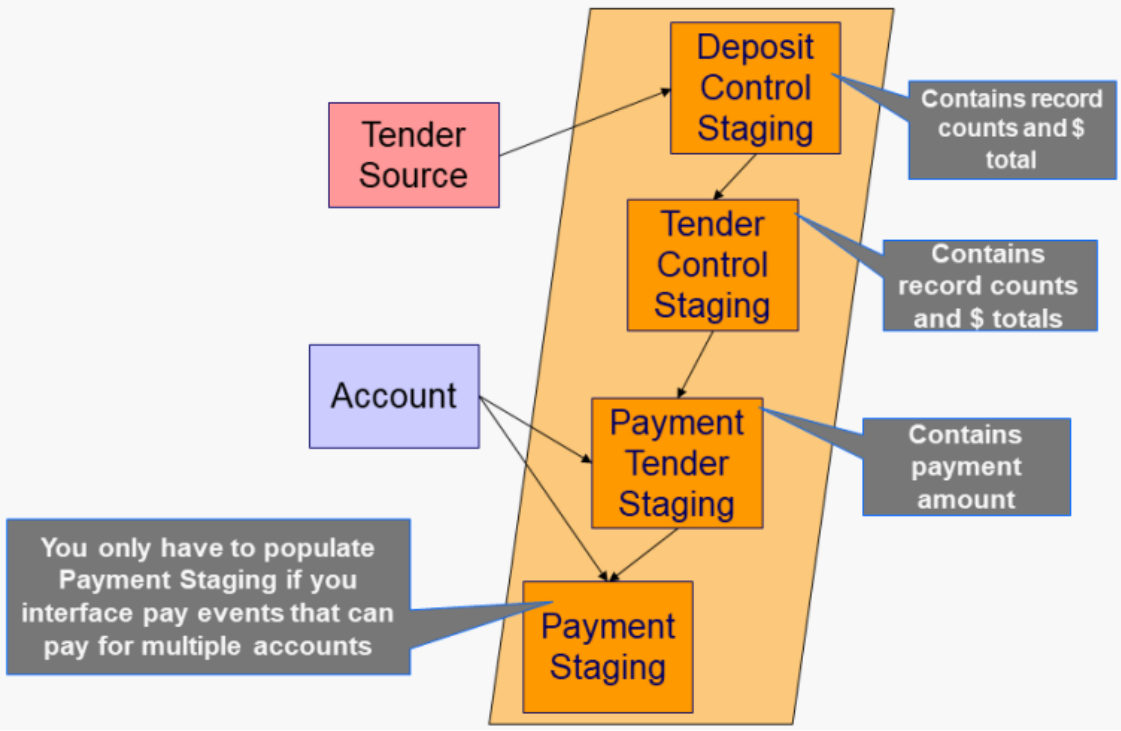
The topics in this section describe each background process referenced above.

- Process X - Populate Payment Upload Staging
- Process PUPL - Upload Payments

Process X - Populate Payment Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).

Copyright © 2026, Oracle and/or its affiliates.



The following topics in this section describe each of these tables:

- Deposit Control Staging
- Tender Control Staging
- Payment Tender Staging
- Payment Staging

Deposit Control Staging

You must create a deposit control staging record for each batch of payments to be uploaded into the system. The name of this table is CI_DEP_CTL_ST. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.

Copyright © 2026, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
EXT_TRANSMIT_ID	50	Y	A/N	This is the unique identifier of the transmission from the external source. This must be a unique value for each transmission from the source.
DEP_CTL_STG_ST_FLG	2	Y	A/N	This must be set to 20 (20 is the lookup value that corresponds with the Pending state)
DEP_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a deposit control record.
TRANSMIT_DTTM	15	Y	DateTime	Date and time that the file was transmitted.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be USD for United States dollars).
TOT_TNDR_CTL_AMT	13.2	Y	N	This column must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging.
TOT_TNDR_CTL_CNT	10	Y	N	This column must equal the number of tender control staging records associated with this deposit control staging.
LAST_UPDATE_INST	10	N	N	This field is populated during the upload. It is the process scheduler instance ID of the process performing the upload.

You must create one or more Tender Control Staging for this deposit control staging record.

Copyright © 2026, Oracle and/or its affiliates.

Top of Page

Tender Control Staging

You must create at least one tender control staging record for each batch of payments to be uploaded into the system. The name of this table is CI_TNDR_CTL_ST. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	50	Y	A/N	This must correspond with the external transmission ID on the parent deposit control staging record.
EXT_BATCH_ID	30	Y	A/N	This is the unique identifier of the batch of payments in respect of the external transmission ID.
TNDR_CTL_STG_ST_FLG	2	Y	A/N	This must be set to 20 (20 is the translate value that corresponds with the Pending state)
TNDR_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
TOT_TNDR_AMT	13.2	Y	N	This column must equal the sum of the payment amounts on the payment tender staging records associated with this tender control staging.
TOT_TNDR_CNT	10	Y	N	This column must equal the number of payment tender staging records associated with this tender control staging.

Copyright © 2026, Oracle and/or its affiliates.

You must create one or more Payment Tender Staging records for this tender control staging record.

Top of Page

Payment Tender Staging

You must create at least one payment tender staging record for each payment associated with the tender control staging record. The name of this table is CI_PAY_TNDR_ST. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	50	Y	A/N	This must correspond with the external transmission ID on the parent tender control staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent tender control staging record.
EXT_REFERENCE_ID	30	Y	A/N	This is the unique identifier of the payment in respect of the external batch ID.
PAY_TND_STG_ST_FLG	2	Y	A/N	This must be set to 10 (10 is the translate value that corresponds with the Pending state)
PAY_TENDER_ID	12	N	N	Leave this column blank. It will be assigned by the system when it creates a tender record.
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).

Copyright © 2026, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
ACCOUNTING_DT	10	Y	Date	This is the date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must correspond with the prime key of one of your tender types. Refer to Setting Up Tender Types for more information.
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old account number that matches this value, the account ID of the tender source's suspense SA will be used on the corresponding tender and payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment.
NAME1	40	N	A/N	This is the customer name on the payment.
CHECK_NBR	10	N	A/N	This is the check number on the payment.

Top of Page

Payment Staging

You need only populate rows on this table if any of the following conditions apply:

- If you need to distribute a payment tender to an account other than that defined with the CUST_ID on the payment tender staging record, you must create a payment staging record. You may distribute a tender to multiple accounts by creating multiple payment

Copyright © 2026, Oracle and/or its affiliates.

staging records. Note, if you want to distribute the payment tender to the same account, you do NOT need a payment staging record.

- If you want to restrict a payment to a specific service agreement, you must insert a row on this table to indicate the specific the service agreement in question. You do this by populating MATCH_TYPE_CD with a value that indicates that you are paying for a specific service agreement and MATCH_VALUE with the unique ID of the service agreement.
- If you practice open-item accounting, you must insert a row on this table for each to indicate the open-item to which the payment should be matched. Note, because open-item customer typically match payments to bills, you would populate MATCH_TYPE_CD with a value to indicate that you are matching by bill ID and MATCH_VALUE with the unique ID of the bill.

The name of this table is CI_PAY_ST. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent payment tender staging record.
EXT_TRANSMIT_ID	50	Y	A/N	This must correspond with the external transmission ID on the parent payment tender staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent payment tender staging record.
EXT_REFERENCE_ID	30	Y	A/N	This must correspond with the external reference ID on the parent payment tender staging record.

Copyright © 2026, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer to which the payment should be distributed. If the system cannot find an account ID or old account number that matches this value, the account ID of the payor is used on the corresponding payment. If the payor's account ID is invalid, the tender source's suspense SA is used.
PAY_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.

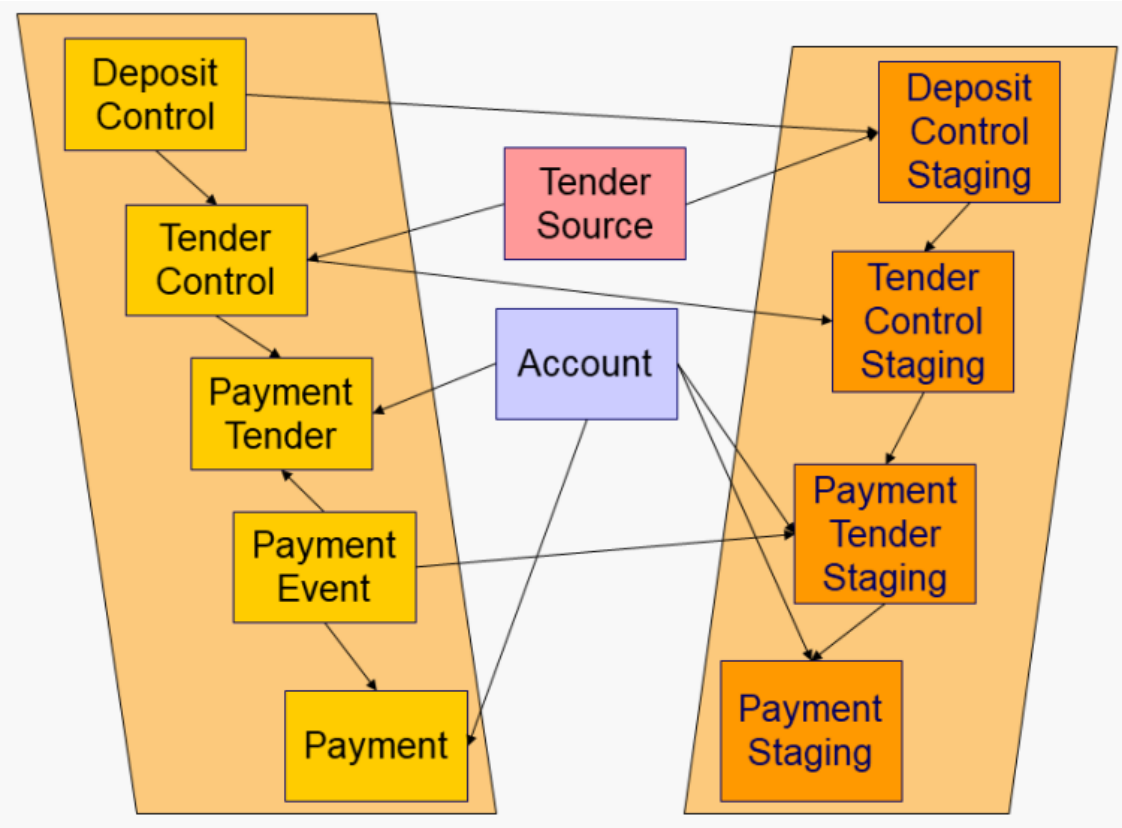
Copyright © 2026, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type.

Process PUPL - Upload Payments

The background process (PUPL) loads the contents of the various payment staging records into the various payment event tables. The tables that are populated by this process are shown in the left orange section of the following ERD (the right orange section are the staging tables populated by the process described above)

Copyright © 2026, Oracle and/or its affiliates.



The topics in this section describe how these tables are populated.

Phase 1 - Create Deposit Control

Phase 2 - Create Tender Control

Phase 3 - Create Payment Events, Tenders, Payments, and Payment Segments

Phase 1 - Create Deposit Control

The following points describe, at a high level, the first phase of the payment upload process:

- PUPL checks that the record counts and money totals of tender control stagings add up to the expected amount on deposit control staging. If not,
  - PUPL sets the status of the deposit control staging to be Error . None of the tender controls within the deposit control will be processed until everything adds up. You can fix these on the Deposit Control Staging page.
  - When PUPL runs next, it will recheck the totals on deposit control stagings that are in Error or Pending
- If the record and dollar amounts are clean,
  - PUPL creates the corresponding deposit control
  - PUPL sets the status of the deposit control staging to be In Progress

Top of Page

Phase 2 - Create Tender Control

The following points describe, at a high level, the second phase of the payment upload process:

Copyright © 2026, Oracle and/or its affiliates.

- PUPL checks that record counts and money totals of payment tender staging(s) adds up to expected amount on tender control staging. If not,
  - PUPL sets the status of the tender control staging to be Error . None of the tender controls within the deposit control will be processed until everything adds up for ALL tender controls. You can fix these on the Tender Control Staging page.

> **Note**
>
> that the Deposit Control Staging record is **not** updated - its status is unchanged. Neither is the Pay Tender Staging record updated - its status also remains unchanged. Only the Tender Control Staging record is updated to be in Error .

- When PUPL runs next, it will recheck the totals of tender control stagings that are in Error or Pending.
- If the record and dollar amounts are clean,
  - PUPL creates the corresponding tender control.
  - PUPL sets the status of the tender control staging to be In Progress .

Top of Page

Phase 3 - Create Payment Events, Tenders, Payments, and Payment Segments

At this point, all deposit control stagings and tender control stagings are in the state of In Progress. Next, PUPL starts the upload of payment tender staging and payment staging. The following points describe, at a high level, this phase of the payment upload process:

- If the payment tender staging record has a future accounting date, the processing for the record is skipped. This prevents uploaded payments from being created and subsequently frozen until their accounting date is reached. (Some external sources may provide advance notification of payments to be made in the future.) A skipped staging record remains in the Pending state until its accounting date is reached.
- PUPL checks money totals of payment advices (if any) adds up to expected amount on payment tender staging.
  - If not, PUPL sets the payment tender staging's status to Error .
  - Any errors are written to the Payment Upload Exception table. You can fix these errors on the Payment Upload Staging page and change the record's status back to Pending.
  - When PUPL runs next, it will recheck the totals of the payment tender staging
- If payment tender staging record is clean:
  - PUPL creates a corresponding payment event, tender, and payment.
  - If the account on payment tender staging is wrong, the account on the corresponding tender will be the tender source's suspense SA's account. Refer to Setting Up Tender Sources for more information. Refer to How To Transfer A Payment From One Account To Another for how to transfer to payment to the correct account.
  - If the account on payment advice is wrong, the account on the corresponding payment will be the account on the payment tender.
  - PUPL distributes the payment(s) amongst the account's service agreements, and payment segments are created. Note, the payment could be in error if there are no SAs for the account (as well as other reasons). Payments in error are written to the payment exception (CI_PAY_EXCP).

Copyright © 2026, Oracle and/or its affiliates.

  - PUPL changes the payment tender staging's status to Complete.
- If all payment tender stagings are Complete:
  - PUPL changes the tender control staging's status to Complete.
  - PUPL changes the deposit control staging's status to Complete .
- If there are payment tender staging that are not Complete
  - The status of the tender control staging will still be In progress .
  - The status of the deposit control staging will still be In progress .
- PUPL will attempt to upload the offending payment tender staging records when it next runs.

> **Note**
>
> **Balancing Control Tables.** The balancing of the deposit and tender control tables during payment upload can either be done by PUPL, or after PUPL successfully completes. The latter might be useful for high volume payment uploads that could result in deadlocks on these control tables. If your implementation wishes to execute control table balancing after PUPL completes, ensure that the necessary batch parameter is set to exclude balancing when PUPL runs and also that the separate balancing batch process (C1-PUBAL) is scheduled after PUPL.

PYUP-PRG - Purge Payment Upload Objects

Completed payment upload staging objects should be periodically purged from the system by executing the PYUP-PRG background process. This background process allows you to purge all Completed payment upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of Completed payment upload objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge Completed payment upload objects as they exist only to satisfy auditing and reporting needs.

Maintaining Deposit Control Staging

The Deposit Control Staging page has three purposes:

- You can view historical deposit and tender control staging records associated with uploaded payments.
- You can correct deposit and tender control records that are in error.
- You can add deposit and tender control records to be uploaded by the payment upload background process.

The topics in this section describe this page.

- Deposit Control Staging - Main
- Deposit Control Staging - Tender Control Staging

Copyright © 2026, Oracle and/or its affiliates.

Deposit Control Staging - Main

This page shows the details of a deposit control staging record.

> **NOT_SUPPORTED**
>
> Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Menu**, and then **Financial**, and then **Deposit Control Staging**, and then **Search.**

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to Setting Up Tender Sources for more information.

External Transmit ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Status shows the state of the deposit control staging records. Potential values are: Incomplete , Pending, In Progress, Partial Load, Complete, Error.

Deposit Control ID is the system-assigned, unique identifier of the related deposit control. This value is populated after the system creates a deposit control for the upload staging record.

Transmission Date/ Time are when the information was interfaced into the system.

Total Tender Controls must equal the number of tender control staging records associated with this deposit control staging.

Total Tender Control Amount must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging. The **Currency Code** related to the amount is adjacent.

Deposit Control Staging - Tender Control Staging

This page shows the details of a tender control staging record.

> **NOT_SUPPORTED**
>
> Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Menu**, and then **Financial**, and then **Deposit Control Staging**, and then **Search** and navigate to the **Tender Control Staging** page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

External Source ID is the external source ID on the parent deposit control staging record.

External Transmit ID is the external transmission ID on the parent deposit control staging record.

The grid that follows contains a row for every tender control staging record linked to the deposit control staging record. The following information is displayed.

External Batch ID This is the unique identifier of the batch of payments in respect of the external transmission ID.

Status This is the state of the tender control staging records. Potential values are: Pending, In Progress, Complete, Error.

Tender Control ID This is the system-assigned, unique identifier of the related tender control. This value is populated after the system creates a tender control for the upload staging record.

Total Tenders Amount This is the sum of the payment amounts on the payment staging records associated with this tender control staging.

Total Number Of Tenders This is the number payment tender staging records associated with this tender control staging.

Payment Upload Staging

The Payment Upload Staging page has three purposes:

- You can view historical payment tender and payment staging records associated with uploaded payments.
- You can correct payment tender records that are in error.
- You can add new payment tender and payment staging records to be uploaded by the payment upload process.

The topics in this section describe this page.

- Payment Upload Staging - Tender Detail
- Payment Upload Staging - Payment Advice
- Payment Upload Staging- Characteristics

Payment Upload Staging - Tender Detail

This page shows the details of a payment tender staging record.

> **NOT_SUPPORTED**
>
> Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Menu**, and then **Financial**, and then **Payment Upload Staging**, and then **Search** to open the **Tender Detail** page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

Ext. Reference ID is the external source's unique identifier of the payment tender.

Customer ID is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old account number that matches this value, the account ID of the tender source's suspense SA will be used on the corresponding tender and payment.

The **Tender Amount** is the amount tendered (i.e., the payment amount).

Tender Type defines the type of tender. Refer to Setting Up Tender Types for more information.

> **Note**
>
> **Authorizing Tenders.** The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

> **NOT_SUPPORTED**
>
> For information about tender types requiring authorization, refer to Credit Card Payments.

MICR ID is the MICR ID associated with the tender.

Check Number is the check number on the payment.

Name is the customer's name (as it appeared on the uploaded tender).

Accounting Date is the date that should be used for accounting purposes.

Pay Tender Staging Status shows the state of the tender control staging records. Potential values are: Pending, Complete, Error.

Payment Event ID is the system-assigned, unique identifier of the related payment event. This value is populated after the system creates a payment event for the upload staging record.

Payment Upload Staging - Payment Advice

If a tender was distributed to customers other than that defined on the Tender Detail page, the customers that the tender was distributed to are defined on this page. This page will not contain information if the tender is distributed to the tender detail's customer.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Menu**, and then **Financial**, and then **Payment Upload Staging**, and then **Search** and navigate to the **Payment Advice** page.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

External Reference ID is the external reference ID of the payment tender.

The grid that follows is only populated if the tender is distributed to customer(s) other than the tendering customer. The following information is displayed.

Customer ID This is the account ID or the old account number of the customer to which the payment should be distributed.

Customer Info If the **Customer ID** is an account ID that exists in the system, the name of the primary person and the customer class of the account are displayed here.

Payment Amount This is the amount of the tender to be distributed to the customer.

Match Type and **Match Value** These fields are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). **Match Type** indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), **Match Value** indicates the ID of the restriction (e.g., the service agreement ID). Valid values of **Match Type** are Service Agreement.

Payment Upload Staging- Characteristics

Select **Menu**, and then **Financial**, and then **Payment Upload Staging**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

Ext. Reference ID is the external source's unique identifier of the payment tender.

Use the characteristics grid to capture additional information about the payment tender staging record.

- Use **Payment Characteristics Entity** to define where each characteristic will be populated on upload. Valid values are: **Payment, Payment Event** or **Payment Tender**

Copyright © 2026, Oracle and/or its affiliates.

- Select a **Characteristic Type** for each payment characteristic entity. You can only choose Characteristic Types defined as permissible for the applicable entity. Refer to Setting Up Characteristic Types & Their Values for more information.
- Enter a valid **Characteristic Value** for the selected characteristic type. You may enter more than one characteristic row for the same characteristic type, each associated with a unique Sequence number. If not specified, the system defaults it to the next **sequence** number for the characteristic type.

Payment Upload Exception

If errors are detected during the payment upload process, a record is written to the payment upload exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-PYUPL background process. This process generates a To Do entry for every record in the payment upload exception table.

You can fix this error using the Payment Upload Staging page and change the status of the record from Error to Pending. When the payment upload process next runs, it attempts to upload this record again.

Interfacing Payments Using Distribution Rules

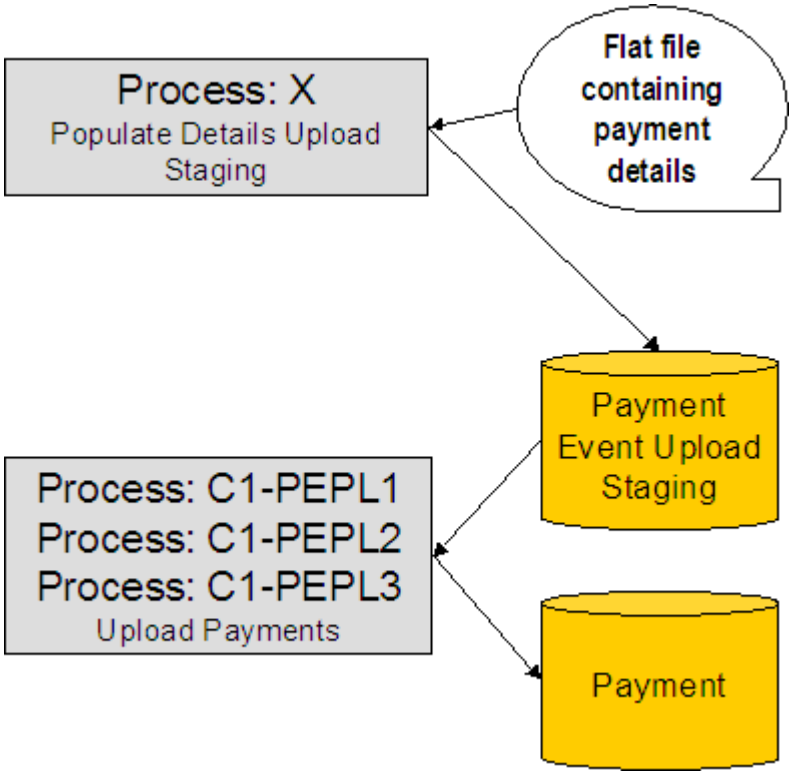
The topics in this section describe how the payment interface using distribution rules works.

- Populating The Payment Event Upload Staging Records
- Payment Event Upload Staging

Populating The Payment Event Upload Staging Records

The following diagram illustrates the processes involved in the uploading of payment event distribution details into the system.

Copyright © 2026, Oracle and/or its affiliates.



The topics in this section describe each background process referenced above.

- Process X - Populate Payment Event Upload Staging
- Data Setup Examples of Payment Distribution Details
- The Lifecycle of a Payment Event Upload Staging Record
- To Do Entries Instead of Exception Records
- Process C1-PEPL1 - Upload Payments (Step 1)
- Process C1-PEPL2 - Upload Payments (Step 2)
- Process C1-PEPL3 - Upload Payments (Step 3)

Process X - Populate Payment Event Upload Staging

Process X refers to the mechanism used by your organization to populate the payment event upload staging table.

payment Event Upload Staging Table

You must create a payment event upload staging record for each payment distribution detail to be uploaded into the system. The name of this table is CI_PEVT_DTL_ST. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments

Copyright © 2026, Oracle and/or its affiliates.

EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.
EXT_TRANSMIT_ID	50	Y	A/N	This is the unique identifier of the transmission from the external source. This must be a unique value for each transmission from the source.
PEVT_DTL_SEQ	12	Y	N	Unique identifier of the detail record within the transmission. The C1-PEPL1 process uses this field to organize the parallel threads.
PEVT_STG_ST_FLG	4	Y	A/N	This must be set to 10 (10 is the lookup value that corresponds with the Incomplete state)
DST_RULE_CD	12	Y	A/N	This must be a valid distribution rule. The distribution rule contains the set of algorithms designed to process the staging detail.
DST_RULE_VALUE	254	Y	A/N	This must be a valid value for the characteristic type defined on the distribution rule.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be USD for United States dollars).
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).

Copyright © 2026, Oracle and/or its affiliates.

ACCOUNTING_DT	10	Y	Date	This is the payment date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must be a valid tender type. Refer to Setting Up Tender Types for more information.
CHECK_NBR	10	N	A/N	This is the check number on the payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment tender.
CUST_ID	15	N	A/N	This field may be used to record customer information.
NAME1	30	N	A/N	This field may be used to capture additional payment tender information.
EXT_REFERENCE_ID	30	N	A/N	This field may be used to capture external information associated with the payment tender.
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.

Copyright © 2026, Oracle and/or its affiliates.

MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), MATCH_VALUE contains the ID of the restriction (e.g., the SA ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type.
TNDR_CTL_ID	10	N	A/N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
ACCT_ID	10	N	A/N	This is the tender account. If left blank, the C1-PEPL1 process will populate this field by calling the Determine Tender Account algorithm defined on the distribution rule. Note that this Account ID is not necessarily unique as multiple staging details can reference the same tender account.

Copyright © 2026, Oracle and/or its affiliates.

PAY_EVENT_ID	12	N	A/N	Leave this column blank. It will be assigned by the system when it creates a payment event record.
PEVT_PROCESS_ID	10	N	A/N	If left blank, the C1-PEPL1 process will set this field equal to the Tender Account ID. This field is used for grouping staging records **and** for organizing parallel threads (in the C1-PEPL2 process). Therefore, it is **strongly encouraged** that it bears a relationship to the tender account ID.
APAY_SRC_CD	12	N	A/N	This must be a valid auto-pay source code.
EXT_ACCT_ID	50	N	A/N	This is the customer's account number at the financial institution
EXPIRE_DT	10	N	Date	This field is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment)
ENTITY_NAME	64	N	A/N	This is the customer's name in the financial institution's system

Data Setup Examples of Payment Distribution Details

Typically, each staging record will represent a single payment event - with a corresponding payment tender and payment. However, it is possible to specify complex relationships within a set of staging records. For example, it will be straightforward to define a set of staging records that represent a single tender but multiple payments (to model the single payment tender of a welfare agency which covers payments for multiple accounts). Similarly, it is equally possible to

Copyright © 2026, Oracle and/or its affiliates.

define a set of staging records that represent a single payment but multiple tenders (although not a common requirement).

Each staging record can represent:

- Zero or one new payment events. Typically, each detail staging record will represent a single payment event. However, it is possible to define multiple records for a single payment event. All details for a single payment event are identified by a common value on the staging record: **Pay Event Process ID**.

> **Note**
>
> **Pay Event Process ID.** This field is used for grouping staging records **and** for organizing parallel threads (in the C1-PEPL2 process). Therefore, it is **strongly encouraged** that it bears a relationship to the tender account ID.

- A partial or one new payment tenders. Typically, each detail staging record will represent a single payment tender. However, it is possible to define multiple staging records for a single pay tender. All details for a single tender are identified by a common payment event ID as well as common tender information (tender type, tender account ID, check number, external reference, etc.).
- A partial or many new payments. Typically, each detail staging record will represent a set of one or many new payments. However, it is possible to define multiple staging records for a single payment. All details for a single payment are identified by a common payment event ID as well as common distribution rule and characteristic value information.

The sections below provide examples of a few of these complex payment event configurations. Note that these examples assume the same distribution rule is referenced in all staging records.

- Staging Entry Example 1: One Payment Event, Two Tenders, Two Payments
- Staging Entry Example 2: One Payment Event, One Tender, Two Payments
- Staging Entry Example 3: One Payment Event, Two Tenders, One Payment

Staging Entry Example 1: One Payment Event, Two Tenders, Two Payments

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112B	1/1/2006	575-40-3030	40.00	Check	102

Top of Page

Staging Entry Example 2: One Payment Event, One Tender, Two Payments

Copyright © 2026, Oracle and/or its affiliates.

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112A	1/1/2006	575-40-3030	40.00	Check	101

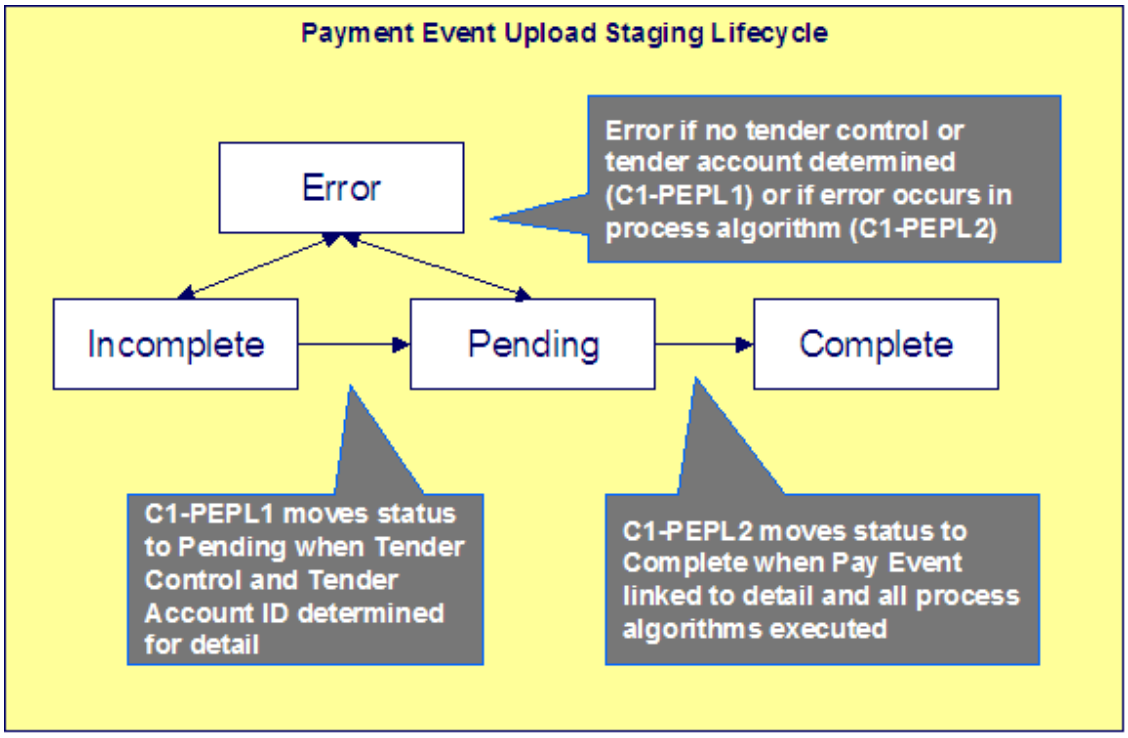
Top of Page

Staging Entry Example 3: One Payment Event, Two Tenders, One Payment

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	575-40-3030	30.00	Check	101
2	1234567890	7878787870	888Q	1/1/2006	575-40-3030	40.00	Check	9872

The Lifecycle of a Payment Event Upload Staging Record

The following diagram shows the possible lifecycle of a payment event upload staging record.



Copyright © 2026, Oracle and/or its affiliates.

- Incomplete. A payment event staging record is initially created in incomplete state. The C1-PEPL1 process sets it to pending once it links it to a tender control and determines its tender account.
- Pending. The C1-PEPL2 process sets a pending record to complete once all processing logic is executed and a payment event is linked to it.
- Complete. When processing of the staging record is complete the record is in the complete state. This is a final state.
- Error. A payment event staging record may be set to Error from Incomplete or Pending states by the C1-PEPL1 and C1-PEPL2 processes respectively.

NOT_SUPPORTED

Refer to To Do Entries Instead of Exceptions for more information on how To Do entries are used to capture processing errors.

To Do Entries Instead of Exception Records

Instead of creating an exception record for staging records in the Error state the C1-PEPL1 and C1-PEPL2 background processes create To Do entries and link them to the offending staging records.

Each process determines the To Do type to use for reporting errors by looking for the To Do Type defined with this process as its creation process.

Process C1-PEPL1 - Upload Payments (Step 1)

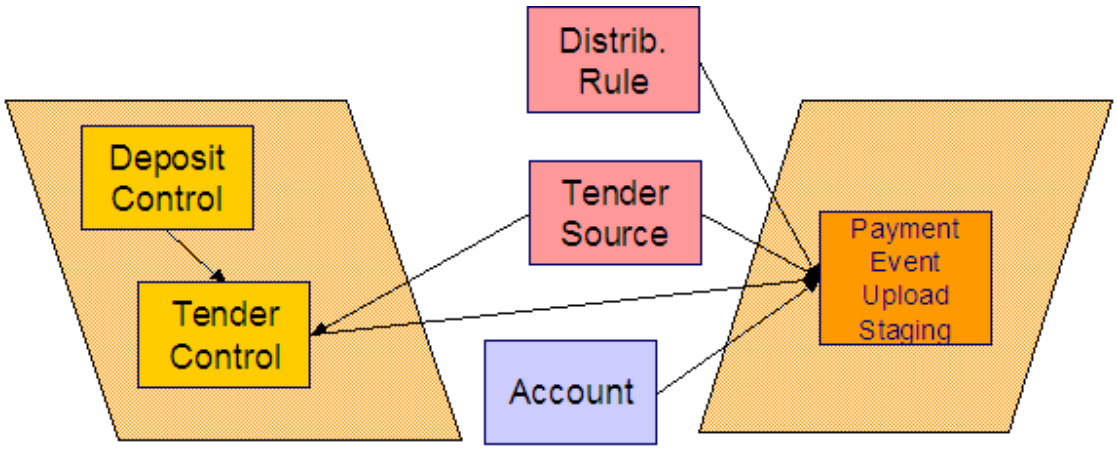
The batch process C1-PEPL1 refers to the first of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of this process is to transition the status of the staging records from Incomplete to Pending. The status of a staging record must remain Incomplete until:

- It is linked to a tender control. This process creates a new deposit control record. It creates a new tender control record if the staging record is not currently linked to one. It then updates the payment event upload staging records with the corresponding **Tender Control ID**.
- The **Tender Account ID** field is populated. The Determine Tender Account algorithm defined on the distribution rule is called and the returned account ID is posted on the staging record.
- The **Pay Event Process ID** field is populated. If left blank this field is set equal to the **Tender Account ID**.

The following diagram and the sections below describe at a high level the processing phases of the C1-PEPL1 background process.

Copyright © 2026, Oracle and/or its affiliates.



- Phase 1 - Create Tender Control
- Phase 2 - Determine Tender Account

Phase 1 - Create Tender Control

> **Note**
>
> This step cannot be bounded by thread range but must execute across the entire population of staging details. Therefore this step in the process is designed so that all parallel threads attempt to execute it at the same time but only one thread succeeds to avoid creating duplicate tender controls.

This step creates all tender controls required by the upload records as follows:

- For each distinct tender source transmissions represented within the incomplete set of upload staging details a deposit control and a tender control are created in an open state. (If the upload record is already linked to a tender control, then that tender control is used and a new one is not created.)
- If an error occurs at this stage
  - A designated staging record for the transmission group (one is picked at random) is set to Error and a To Do entry is created and linked to it to capture the error message that applies for the whole transmission group. Other records in the group remain incomplete .
  - The process stops.

> **Note**
>
> **Group Error.** This technique allows for an easier recovery from a setup error that may affect a large volume of records in a single transmission. Capturing the error only on a single designated record requires only this record to be set back to Incomplete once the setup issue is corrected. It is important to note that the transmission group is not processes if at least one of the records in the group is in Error status.

- Once a transmission group of records is fully processed, a To Do cleanup processing takes place to complete To Do entries previously raised for its designated staging record.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to To Do Entries Instead of Exceptions for more information on how To Do entries are used to capture processing errors.

Top of Page

Phase 2 - Determine Tender Account

After all tender controls have been created the second step attempts to transition all incomplete records to the pending state.

Each incomplete staging record is processed as follows:

- If not yet associated with a tender control ID, the process looks for an Open tender control that matches the batch code, batch number and external source ID and links it to the staging record. An error is raised if a matching tender control is not found.
- Execute the Determine Tender Account algorithm defined on the distribution rule and populate the record with the returned account ID. Note that the algorithm is called even when the tender account is populated to provide for a potential override of the initial value when necessary. An error is raised if a tender account may not be determined.
- If not already populated, the **Pay Event Process ID** is set equal to the Tender Account ID. The C1-PEPL2 process uses this field to organize the parallel threads and to group multiple staging details into a single payment event.
- The staging record is moved to pending state.
- If any errors occur set the record to Error and create a To Do entry for the error message.
- If no error occurred, a To Do cleanup processing takes place to complete To Do entries previously raised for the staging record.

NOT_SUPPORTED

Refer to To Do Entries Instead of Exceptions for more information on how To Do entries are used to capture processing errors.

Note

This step is designed to support execution in parallel threads based upon the sequence number portion of the staging table prime key.

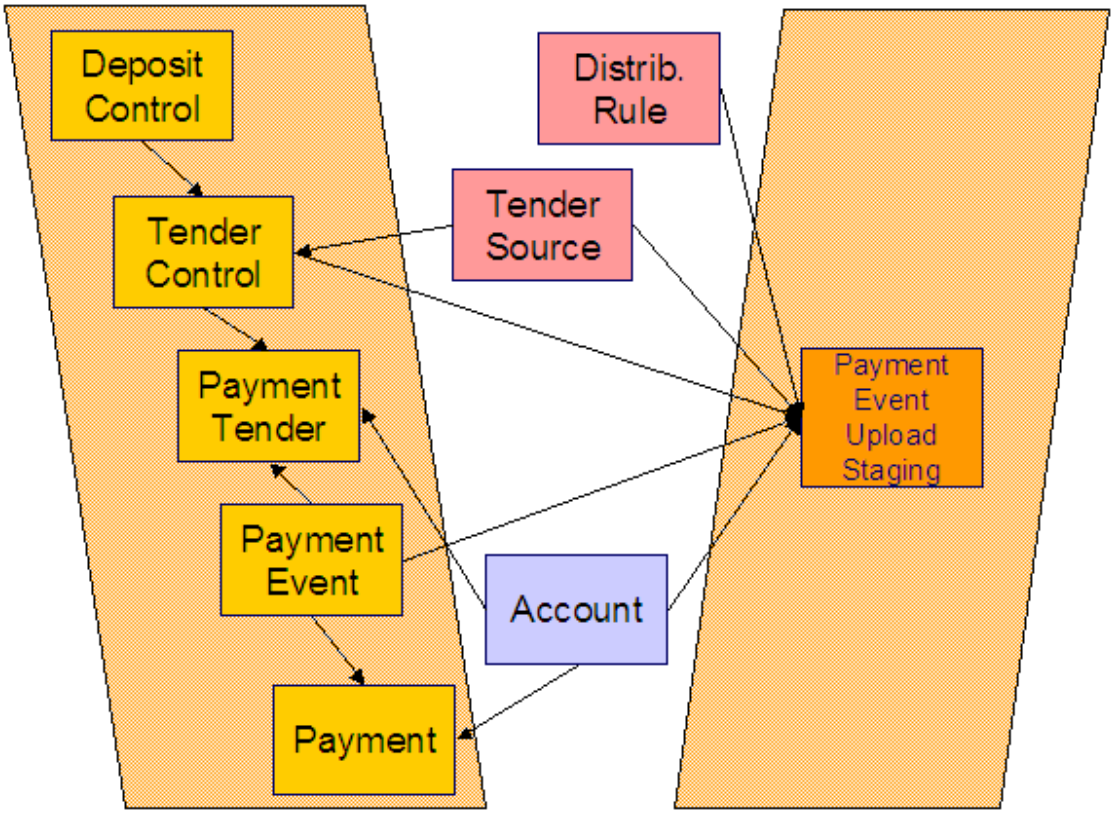
Process C1-PEPL2 - Upload Payments (Step 2)

The batch process C1-PEPL2 refers to the second of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the C1-PEPL2 process is to create payment events, payment tenders and payments and transition the corresponding staging records from Pending to Complete.

Copyright © 2026, Oracle and/or its affiliates.

The following diagram and the section below describe at a high level the processing phases of the C1-PEPL2 background process.



Each distinct group of pending staging records associated with the same external source ID, external transmit ID, accounting date, and pay event process ID is processed as follows:

- A payment event is created for the group and stamped on each of its records.
- A payment tender is created for each distinct set of staging records having the same tender account, tender type and other tender information fields, except for the tender amount.
- The Create Payment algorithm is then called for each distinct group of staging records having the same tender account, distribution rule and rule value providing it with the total amount for the group.
- The staging record is moved to the complete state.
- If any error occurs a designated staging record for the payment event group (one is picked at random) is set to Error and a To Do entry is created and linked to it to capture the error message that applies for the whole group. Other records in the group remain incomplete .

> **Note**
>
> **Group Error.** This technique allows for an easier recovery from an error that affects all staging records for a single payment event. Capturing the error only on a single designated record requires only this record to be set back to pending once the issue is corrected. It is important to note that the whole set of records is not processes if at least one of the records in the group is in Error status.

Copyright © 2026, Oracle and/or its affiliates.

- If no error occurred, a To Do cleanup processing takes place to complete To Do entries previously raised for the designated staging record.

> **NOT_SUPPORTED**
>
> Refer to To Do Entries Instead of Exceptions for more information on how To Do entries are used to capture processing errors.

> **Note**
>
> This process is designed to support execution in parallel threads based upon the payment event process ID field.

Process C1-PEPL3 - Upload Payments (Step 3)

The batch process C1-PEPL3 refers to the last of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the C1-PEPL3 process is to update the status of the related deposit and tender controls from open to balanced.

Each distinct tender control for which all associated staging records are in complete status is processed as follows:

- The tender deposit is created.
- The tender control is set to balanced.
- The deposit control is set to balanced.

> **Note**
>
> This process is designed to support execution in parallel threads based upon the Tender Control ID field.

Payment Event Upload Staging

The Payment Event Upload Staging page has three purposes:

- You can view historical payment event upload staging records associated with uploaded payments.
- You can correct payment event upload staging records that are in error.
- You can add new payment event upload staging records to be uploaded by the payment event upload processes.

The topics in this section describe this page.

- Payment Event Upload Staging - Main

Copyright © 2026, Oracle and/or its affiliates.

Payment Event Upload Staging - Main

This page shows the details of a payment event upload staging record.

> **NOT_SUPPORTED**
>
> Refer to Populating The Payment Event Upload Staging Records for more information about this record.

Open this page using **Menu**, and then **Financial**, and then **Payment Event Upload Staging**, and then **Search.**

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to Setting Up Tender Sources for more information.

External Transmission ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Sequence is the identifier of the record within the transmission. The C1-PEPL1 process uses this field to organize the parallel threads.

Accounting Date is the payment date that should be used for accounting purposes.

Distribution Rule is the rule by which the payment detail is to be processed. A default distribution rule is displayed if you have set one.

Rule Value is a value associated with the payment and expected by the distribution rule.

Match Type and **Match Value** are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). **Match Type** indicates how the payment should be distributed (e.g., only distribute to a specific service agreement), **Match Value** indicates the ID of the restriction (e.g., the service agreement ID).

Payor Account ID is the tender account. If not populated the C1-PEPL1 process populates this field by calling the Determine Tender Account algorithm defined on the distribution rule.

Tender Amount is the amount tendered (i.e., the payment amount).

Currency Code is the currency of the tendered amount. This should be the same as currency defined on the tender source.

Tender Type defines the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the installation record.

> **Note**
>
> **Authorizing Tenders.** The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For information about tender types requiring authorization, refer to Credit Card Payments.

Check Number is the check number on the payment.

MICR ID is the value of the magnetic ink character recognition (MICR) line on the payment.

External Reference ID may be used to record external information associated with the payment tender.

Customer ID may be used to record additional customer information.

Name may be used to record additional payment tender information.

Tender Control ID is the tender control associated with the payment. This field is should typically be left for the C1-PEPL1 process to populate.

If the **Tender Type** is associated with an automatic payment, the **Auto Pay** section displays. The system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.

- **Auto Pay Source** is the financial institution / credit card company that receives the automatic payment request.
- **External Account ID** is the customer's account number at the financial institution.
- **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
- **Name** is the customer's name in the financial institution's system.

Pay Event Process ID is used to group multiple staging records into a single payment event. If not populated, the C1-PEPL1 process sets this field equal to the **Payor Account ID**.

Pay Event Staging Status shows the state of the staging record. Refer to The Lifecycle of a Payment Event Upload Staging for a state transition overview.

Payment Event ID is the system-assigned, unique identifier of the related payment event. The C1-PEPL2 process populates this field when it creates a payment event for the upload staging record. You can use this field to navigate to the payment event page.

If a staging record is in Error state then the error message associated with the corresponding To Do entry is displayed.

Adjustments

In this section, we describe when and how to adjust a service agreement's balance.

- The Big Picture Of Adjustments
- Maintaining Adjustments

Copyright © 2026, Oracle and/or its affiliates.

- How and When To Use An Adjustment
- Interfacing Adjustments From External Sources

The Big Picture Of Adjustments

The topics in this section provide background information about a variety of adjustment issues.

> **NOT_SUPPORTED**
>
> We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate the place of an adjustment in the system's financial architecture. In particular, refer to Setting Up Adjustment Types and Setting Up Adjustment Type Profiles.

- Adjustments - Current Balance versus Payoff Balance
- Canceling Adjustments
- Transfer Adjustments
- Calculated Adjustments
- Adjustment Amount May Be Positive, Negative, Or Zero
- Adjustment Type Controls Everything
- Unbilled Adjustments And Aged Debt
- An Adjustment May Affect More Than Just Customer Balances

Adjustments - Current Balance versus Payoff Balance

Adjusting how much a customer owes involves changing a service agreement's payoff balance and/or current balance by creating an adjustment. In this section we describe these two balances.

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- When Current Balance Equals Payoff Balance
- When Current Balance Differs From Payoff Balance
- Adjustment Type And Balances

When Current Balance Equals Payoff Balance

Copyright © 2026, Oracle and/or its affiliates.

For most service agreements, payoff balance and current balance are always the same (or in colloquial speech - the amount the customer thinks they owe equals what they really owe). In this situation, an adjustment is easy: both payoff balance and current balance are adjusted by the same value.

Let's run through a typical example. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied (i.e., the running balance):

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125	125	125
15-Jan-99	Payment: $150	-25	-25
2-Feb-99	Bill: $175	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200	200	200
15-Mar-99	Payment: $150	50	50
2-Apr-99	Bill: $225	275	275
27-Apr-99	Adj: Late Payment Charge $10	285	285

As you can see, payoff balance and current balance are always in sync.

When Current Balance Differs From Payoff Balance

For some service agreements, payoff balance and current balance differ (or in colloquial speech - the amount the customer thinks they owe differs from what they would owe if they wanted to payoff their account).

Let's run through an example of a customer on a budget to illustrate a classic service agreement where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0
2-Apr-99	Adj: Mar 3, 97 bill incorrect due to meter exchange discrepancies; manual correction of -$33.	17	0

Notice the adjustment of -$33 affects only payoff amount. This is because the customer is on a budget. The adjustment will not affect their monthly payment and therefore it doesn't affect how much they think they owe.

Copyright © 2026, Oracle and/or its affiliates.

The following table describes the various types of service agreements where these balances may differ.

Type Of Service Agreement	What Payoff Balance Holds	What Current Balance Holds
Deposit	Payoff balance holds the amount of deposit you are holding on behalf of the customer's debt. This amount is showed as a credit (with a negative sign) because you truly owe the customer this money.	Current balance holds the amount the customer owes for a deposit.
Charitable contribution	N/A - charitable contribution service agreements never have a payoff balance.	Current balance holds the amount the customer owes for a charitable contribution.
Payment arrangements	Payoff balance holds the total amount of delinquent debt that remains unpaid. When a payment arrangement is started, you transfer the payoff balance from the utility service agreement to the payment arrangement service agreement. Each billing period, the customer pays down this balance.	Current balance holds the amount the customer owes in respect of the billed payment arrangement amounts.
Utility agreements on a budget plan	Payoff balance holds the actual amount owed (actual bills minus payments).	Current balance holds the amount the customer owes in respect of the billed budget amount.
Loans	Payoff balance holds the payoff amount for the loan, which is the current balance (the billed loan amount) plus the principal balance (the unbilled loan amount) and any accrued interest charges.	Current balance holds the amount the customer owes in respect of the billed amount (the periodic payment amount).

Adjustment Type And Balances

When you create an adjustment, you must define its adjustment type. The adjustment type controls how payoff balance and current balance are affected by the adjustment amount.

You can only pick adjustment types that make sense for the service agreement. For example, if your organization doesn't allow budget billing, all adjustments for a utility service agreement will affect payoff balance and current balance the same (you can't get the balances out-of-sync).

The various types of adjustments that may be linked to a service agreement are controlled by the adjustment profiles defined on the service agreement's SA type.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Adjustment Type Controls Everything for more information.

Canceling Adjustments

You may cancel any frozen adjustment. Canceling the adjustment creates another financial transaction that reverses the financial effects of the original adjustment. The impact of the cancellation appears on the customer's next bill. You may view both the original financial transaction and its cancellation on the Adjustment page.

Transfer Adjustments

A convenient mechanism exists to transfer moneys between two service agreements. The net effect of such a request is the creation of two adjustments, each of which is linked to the other. Both adjustments are created together, frozen together, and posted to the GL together. This is useful because you can't create one side of the transfer and without the other. Refer to How To Create A Transfer Adjustment for more information.

Note

Inter or Intra Account. It's important to be aware that the transfer can be inter or intra account (i.e., the accounts on the two adjustments may be different).

Calculated Adjustments

An adjustment can calculate the adjustment amount using an algorithm. A calculated adjustment can be used to calculate:

- Sales or other taxes for a base amount
- Any charge amount based on a user-supplied quantity, such as feet, miles, hours, etc.
- Others.... Any calculation that can be made by a rate can be applied to an adjustment.

When you create a calculated adjustment, you may be asked to supply a base amount. The user may supply the base amount or the amount could be defaulted from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

When the adjustment is generated, the calculation algorithm may use the base amount and calculation date. The base package algorithm calls the rate application and returns the calculated amount with calculation lines and GL distribution codes.

An adjustment's adjustment type controls whether the adjustment is calculated and the algorithm that performs the calculation.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to Adjustment Type Controls Everything and Setting Up Calculated Adjustment Types.

Adjustment Amount May Be Positive, Negative, Or Zero

An adjustment's amount may be positive, negative or zero:

- A positive amount causes the customer's balance(s) to increase.
- A negative amount causes the customer's balance(s) to decrease.
- A zero amount will not affect the customer's balance(s). A zero amount is odd, but necessary when you need to use an adjustment to correct the GL distribution. Refer to How To Use An Adjustment To Change Amounts Booked In Your GL for more information.

Adjustment Type Controls Everything

When you create an adjustment, you must define its adjustment type. The topics in this section describe how adjustment type controls the behavior of an adjustment.

- Controls Which Balance(s) Are Affected
- Defines The GL Account Affected By The Adjustment
- Defaults The Adjustment Amount
- Calculates The Adjustment Amount
- Controls The Interface To A/P & 1099 Reporting
- Controls Information Printed On The Bill
- Controls If The Adjustment Can Be Frozen Prior To Bill Completion
- Controls If The Adjustment Requires Approval

Controls Which Balance(s) Are Affected

The adjustment type's financial transaction (FT) algorithm controls how payoff balance and current balance are affected by the adjustment amount.

Defines The GL Account Affected By The Adjustment

Most adjustments affect the general ledger (GL) in some way. The following points describe the source of these GL accounts.

- For normal adjustments (i.e., non-calculated adjustments that affect a single service agreement), there is a single accounting entry generated:

Copyright © 2026, Oracle and/or its affiliates.

  - One side of the accounting entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment. For example, if you are adjusting the payoff balance on a normal service agreement, the A/R account is constructed from the distribution code on the service agreement's SA type.
  - The other side of the accounting entry is taken from the distribution code on the adjustment's adjustment type. For example, if a late payment charge is created, the "late payment charge" adjustment type references this fee's revenue account.
- For transfer adjustments (i.e., adjustments used to transfer moneys between two service agreements), there are two accounting entries generated - one for the "from" side and one for the "to" side. Each adjustment carries its own set of balanced GL accounting details.
  - For each adjustment, one side of the entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment (just like for normal adjustments).
  - The other sides of both accounting entries have the same GL account. This account should be the intermediate clearing GL account that is to be used for the transfer. The source of this clearing GL account is the distribution code on the adjustment type used to transfer the funds.
- For calculated adjustments (i.e., adjustments where rates may be applied to calculate the adjustment amount), there is a single accounting entry:
  - For each adjustment, one side of the entry is taken from the distribution code on the SA type of the service agreement affected by the adjustment (just like for normal adjustments).
  - The other side of the entry is taken from the distribution codes that resulted from the rate calculation if the adjustment financial transaction algorithm is set up to use the calculation lines as the distribution code source. If the adjustment financial transaction algorithm is set up to use the adjustment type as the distribution code source, the other side of the accounting entry is taken from the distribution code on the adjustment type.

> **Note**
>
> **Not all adjustments affect the GL.** As a general rule of thumb, only those adjustments that affect a service agreement's payoff balance affect the GL. So, if you adjust a budget customer's current balance, but don't adjust their payoff balance, the GL is not affected.

Defaults The Adjustment Amount

The adjustment type may default the adjustment amount in one of the following ways:

- The adjustment type may specify a default amount. This would be used for those adjustment types that have a standard charge for all customers that receive this adjustment, for example a non-sufficient funds charge.
- The adjustment type may specify a default adjustment amount algorithm. This would be used for those adjustment types that have a charge that varies based on other factors. For example, a non-sufficient funds charge may be based on a customer's credit rating.

When an amount is defaulted onto a new adjustment it may be overridden by a user.

Copyright © 2026, Oracle and/or its affiliates.

Calculates The Adjustment Amount

Some adjustment types calculate the adjustment amount. If the adjustment type uses a calculated amount, a base amount is passed to the rate application, which then returns the calculated amount with calculation lines and GL distribution codes.

The adjustment type's generate adjustment algorithm controls which rate is applied to the base amount. A user supplied calculation date controls which version of the rate is used. The user may supply the base amount or it may be defaulted from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

Controls The Interface To A/P & 1099 Reporting

If the adjustment type is associated with a payment (e.g. a refund or a loan disbursement) of money to a customer, the adjustment type indicates such with a reference to an A/P request type.

When an adjustment that references an A/P request type is frozen, an A/P download request record is created. This record is the interface request to ask your A/P system to cut a check. This interface record is marked with a batch process ID and run number.

- The batch process ID is the background process responsible for creating the flat file that contains check request that is interfaced to your account's payable system. The batch process ID is defined on Installation Options - Financial. The base package is supplied with a skeletal background process (APDL) that must be populated with logic to format the records in the format compatible with your accounts payable system.
- The run number is the batch process ID's current run number.

> **NOT_SUPPORTED**
>
> Refer to Accounts Payable Check Request for more information.

If the resultant check needs to be reported for income tax purposes under a specific 1099 category, the category is also specified on the adjustment type. The 1099 category is in turn interfaced to the A/P system (the system does NOT manage 1099 reporting).

Controls Information Printed On The Bill

If the adjustment appears on the customer's next bill, the verbiage is specified on the adjustment type (and may NOT be overridden on the adjustment).

Controls If The Adjustment Can Be Frozen Prior To Bill Completion

Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for how adjustment type can control whether adjustments of a given type can be frozen prior to bill completion.

Copyright © 2026, Oracle and/or its affiliates.

Controls If The Adjustment Requires Approval

If an adjustment's adjustment type references an approval profile, the system will not allow the initiating user to freeze the adjustment. Rather, the initiating user can submit the adjustment for approval. When an adjustment is submitted for approval, the system determines the necessary approval levels and notifies the first approver. The system will freeze the adjustment when last approver approves the adjustment (if the adjustment's adjustment type allows it to be frozen prior to the completion of the next bill).

> **Note**
>
> **Only online adjustments are subject to approval**. The system assumes that no approval is necessary for adjustments created by batch processes even those whose adjustment type references an approval profile.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Adjustment Approvals for more information.

Unbilled Adjustments And Aged Debt

A Frozen adjustment waits in limbo until the customer's next bill is produced. This limbo period could be several weeks if the customer is billed infrequently. When the customer's next bill is completed, all limbo adjustments are linked to the bill. A brief description (as defined on the adjustment type) appears on the printed bill.

It is important to stress the following in respect of these limbo adjustments:

- If the adjustment decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the adjustment appears on a bill.
- If the adjustment increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the adjustment. There is a switch on the financial transaction created for the adjustment called New Charge that controls the arrears behavior. When the FT for this type of adjustment is created, the switch is set to on and the customer's aged debt will not reflect the adjustment amount until the adjustment is swept onto a bill. The moment the adjustment is swept onto the customer's bill, the debt starts aging. A user may turn the switch off to cause the debt to start aging sooner. For this case, the date on which the adjustment starts aging must be defined in the Arrears Date field.
- The amount a customer owes in total is immediately affected by the adjustment regardless of whether the adjustment appears on a bill. This means that amount of aged debt may not be in sync with the total amount owed. This seems odd but is useful from a credit and collections perspective. You see, you probably don't want to start aging an adjustment until the customer has actually seen it.

Copyright © 2026, Oracle and/or its affiliates.

An Adjustment May Affect More Than Just Customer Balances

NOT_SUPPORTED

When an adjustment is frozen or cancelled, a customer's current and payoff balances are affected. However, several other objects may be affected when such events occur. Refer to Obscure Things That Can Happen for more information.

Maintaining Adjustments

An adjustment is used to change the amount of debt stored on a service agreement. The topics in this section describe the pages on which adjustments are maintained.

NOT_SUPPORTED

For more information about adjustments, refer to The Big Picture Of Adjustments.

- The Lifecycle Of An Adjustment
- Adjustment - Main Information
- Adjustment - Characteristics
- Adjustment - Transfer Adjustment
- Adjustment - A/P Request
- Adjustment - Adjustment Portal
- Adjustment - Approval
- Financial - Adjustment Calculation Line Characteristics

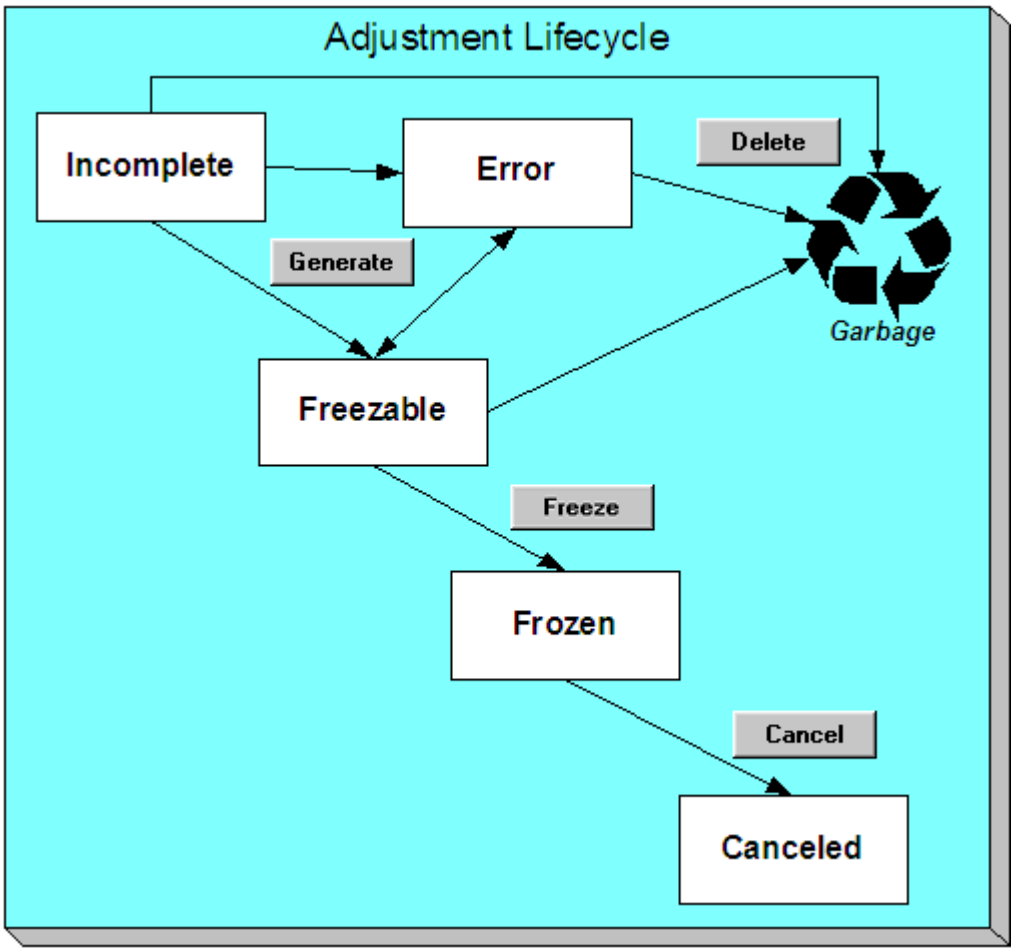
The Lifecycle Of An Adjustment

The following diagram shows the possible lifecycle of an adjustment.

Warning

This diagram only makes sense in the context of the page used to maintain Adjustments. Refer to Adjustment - Main Information for the details.

Copyright © 2026, Oracle and/or its affiliates.



> **Note**
>
> Even though New is considered a valid adjustment state, this is only used as default when adding an adjustment. Once the adjustment is saved, it is assigned the appropriate state in the adjustment life cycle. An adjustment will never exist in the New state.

Adjustments are initially created in the Incomplete state. Adjustments in this state don't have a financial transaction. This means, you can change the adjustment type and amount at will.

Click **Generate** to generate a financial transaction for the adjustment. The financial transaction contains the adjustment's effect on the general ledger and on the customer's payoff and current balances. If the adjustment is calculated, the algorithm on the adjustment type's Generate Adjustment event controls how the adjustment is calculated. If the adjustment is calculated, you must specify the calculation date, which is passed as a parameter to the calculation algorithm and is used for calculations that are effective dated (e.g., rate version or bill factor value). The financial transaction (FT) algorithm defined on the adjustment type Adj. Financial Transaction event controls how the financial transaction is constructed. For calculated adjustments, the distribution code may be taken from the adjustment type or the calculation lines.

After generating the financial transaction, the adjustment becomes Freezable. While in this state, you may change the service agreement, adjustment type and amount at will. However, if

Copyright © 2026, Oracle and/or its affiliates.

you change this information, you will have to regenerate the financial transaction (by clicking the Generate button).

In the very rare situation when the system cannot generate the financial transaction because of inconsistent setup data, the adjustment is moved to the Error state. You may regenerate the financial transaction after correcting the source of the error. You may also delete such an adjustment.

Click the delete button to physically remove an Incomplete, Error or Freezable adjustment from the database.

> **Warning**
>
> Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is completed. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

Click **Freeze** to freeze the adjustment and its financial transaction. After doing this, the adjustment's state becomes Frozen. While in this state, you cannot change the adjustment's type or amount, but you may change:

- When the adjustment starts aging.
- Whether the adjustment appears on a customer's bill.
- The accounting date used to derive the general ledger accounting period(s) to which the financial transaction is booked.

> **Note**
>
> **Adjustments may require approval**. If the adjustment type referenced on the adjustment has an approval profile, the Freeze button is replaced with a Submit for Approval button. If an adjustment is currently being approved, all action buttons are disabled because the adjustment must be either approved or rejected using the Adjustment - Approval page.
>
> However, the Delete button is enabled for an adjustment in a freezable state and linked to an approval request that is in a “no approval necessary” state. If the adjustment is deleted a log entry is created on the associated approval request.
>
> Refer to The Big Picture of Adjustment Approvals for more information.

If you need to remove the financial effects of an adjustment, click **Cancel**. Canceling an adjustment causes the generation of another financial transaction. This new financial transaction reverses the financial impact of the original adjustment.

Adjustment - Main Information

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The Main page contains core adjustment information. Open this page using **Menu**, and then **Financial**, and then **Adjustment**, and then **Search.**

> **Warning**
>
> All adjustments reference a very important field that controls much validation and processing; this field is called Adjustment Type. Take special care when adding a new adjustment to specify the appropriate Adjustment Type as it affects how the adjustment appears on the customer's bill, how the adjustment is reflected in your general ledger, and much more. After the adjustment is frozen, you may not change its Adjustment Type.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

> **Note**
>
> **Formatting may be performed by a plug-in.** The contents of **Adjustment Info** may be formatted by a plug-in algorithm on the Adjustment Type. Refer to the base package's C1-ADT-INFO for an example. If such an algorithm is not plugged-in on the Adjustment Type, the system looks for a corresponding algorithm on the installation record. Refer to the base package's C1-ADI-INFO for examples. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Service Agreement** ID to define the service agreement whose value needs to be adjusted. Basic information about the service agreement appears adjacent.

Premise is a display-only field that shows the **Service Agreement**'s characteristic premise (i.e., the primary premise associated with the service agreement).

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger. This field is gray after the adjustment is frozen.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You can only choose certain adjustment types.** The service agreement's SA type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the SA type.
>
> The list of adjustment types is also restricted to those without a CIS Division and those with a CIS Division that matches the service agreement's CIS Division.

Enter the **Amount** of the adjustment. If the adjustment type is a calculated adjustment, the **Calculated Amount** and **Calculation Date** are displayed. The values of the calculated amount and calculation date are displayed after the adjustment is generated. The calculated amount shows the result of the generate adjustment algorithm. The calculation date is specified when you click Generate for a calculated adjustment type. It is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version calculation group or bill factor value).

> **Note**
>
> **Default note.** The adjustment amount defaults based on the adjustment type. When you change the adjustment type, the amount changes accordingly. You may change the adjustment amount after it is defaulted.

On Behalf SA ID is a display-only field that is only populated on system-generated adjustments instigated by one customer, but attributed to another customer. For example, if chargeable field work is performed for a customer, but is payable by the customer's meter maintenance service provider, the adjustment will be linked to the service provider and will reference the customer's SA as its **On Behalf SA ID**. This field is only visible if it is populated on the adjustment.

You may not change an adjustment's **Adjustment Status** directly. Rather, you use the buttons in the **Adjustment Actions** area. Refer to The Lifecycle Of An Adjustment for the details.

If the status is canceled, the **Cancel Reason** is displayed.

Use the **Comments** to describe anything unusual about the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

If the adjustment is subject to approval, a message indicating such appears. Clicking this message navigates the user to the **Approvals** tab. Refer to The Big Picture of Adjustment Approvals for more information.

The financial transaction (FT) grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.

Copyright © 2026, Oracle and/or its affiliates.

- **Current Amount** contains the FT's effect on the service agreement's current balance.
- **Payoff Amount** contains the FT's effect on the service agreement's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the service agreement's account.

The **Calculation Lines** grid contains the details of the calculations associated with a calculated adjustment. It only appears if the rate that calculated the adjustment amount created at least one calculation line. One row exists for every calculation involved in the process. This information is for audit purposes only and cannot be modified. The following information displays in the grid:

- If at least one of the calculation lines has characteristics, **Calc Line Char** displays go to buttons, allowing you to go to characteristics that are linked to a specific adjustment calculation line. This column is only displayed if at least one of the calculation lines has characteristics.
- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the calculation line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the calculation line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **Appears in Summary** switch defines if this line's amount also appears on a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column.
- **Unit of Measure** (UOM) is the unit of measure of the service quantity priced on the calculation line.
- **Time of Use** (TOU) is the time-of-use code of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity differs from the measured consumption if there are pre-processing calculation groups in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.
- **Rate Component Sequence** refers to the sequence number of the rate component on the applicable rate version calculation group that was used to calculate the line. This is only populated if the original rate engine calculated the charges.
- **Calculation Rule** is the rule that calculated the line. This is only populated if the new rate engine calculated the charges.
- **Measures Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt.

Copyright © 2026, Oracle and/or its affiliates.

- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the line was calculated using a bill factor (because only bill factors use characteristic values). Refer to An Illustration Of A Bill Factor And Its Characteristics for more information.

> **NOT_SUPPORTED**
>
> For more information, refer to Calculated Adjustments .

The **Adjustment Actions** area contains buttons that you use to commit and cancel the adjustment's financial impact. The adjustment's status controls the button you can see and select. Refer to The Lifecycle Of An Adjustment for the details.

> **Warning**
>
> Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is completed. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.
>
> **Cancel may not be enabled**. The cancel button is not enabled if the adjustment is linked to a bill that is written off.

> **Note**
>
> **Canceling An Accounts Payable (A/P) Adjustment.** If you need to cancel an A/P adjustment (i.e. an adjustment of an adjustment type that has an A/P request type) that has already been extracted by A/P, refer to How To Cancel An A/P Adjustment After It's Been Selected By A/P.

Adjustment - Characteristics

You use this page to link additional information to the adjustment. Open using **Menu**, and then **Financial** , and then **Adjustment**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

The characteristics collection contains information that describes miscellaneous information about the adjustment.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined on the adjustment's adjustment type.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

> **Note**
>
> **Default Note.** An adjustment's characteristics default from the adjustment type.

Adjustment - Transfer Adjustment

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The Transfer Adjustment is used to define the reciprocal adjustment associated with a transfer adjustment (there are always two adjustments associated with a transfer). Open using **Menu**, and then **Financial** , and then **Adjustment**, and then **Search**.

The Main page describes one side of the transfer adjustment; for example, the “transfer from” adjustment. As well, the Transfer Adjustment page describes the reciprocal adjustment; for example, the “transfer to” adjustment.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Service Agreement** ID to define the transfer to service agreement. Basic information about the service agreement appears adjacent.

Premise is a display-only field that shows the **Service Agreement**'s characteristic premise (i.e., the primary premise associated with the service agreement).

The **Adjustment Type** and adjustment **Amount** from the first page are displayed. If you need to change either value, return to the first page.

Copyright © 2026, Oracle and/or its affiliates.

Adjustment Status shows the status of the transfer to adjustment. If you need to change the adjustment's status, use the action buttons.

Use the **Comments** to describe anything unusual about the adjustment.

Transfer Adj ID is the system-assigned, unique identifier of the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

The financial transaction grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Current Amount** contains the FT's effect on the service agreement's current balance.
- **Payoff Amount** contains the FT's effect on the service agreement's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the account.

Adjustment - A/P Request

> **NOT_SUPPORTED**
>
> The **Description of Page** below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The A/P Request page contains information about adjustments used to refund money via an A/P check request. This page is only relevant if:

- The adjustment's adjustment type references an A/P Request Type (i.e., it will be interfaced to your accounts payable system which will cut the check), AND
- The adjustment is frozen.

Open this page using **Menu**, and then **Financial** , and then **Adjustment**, and then **Search** and navigate to the **A/P Request** page.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Adjustment type controls A/P check requests.** The A/P Request information is populated when an adjustment used to refund money via an A/P check request is frozen. Whether or not an adjustment is interfaced to A/P is controlled by the adjustment's adjustment type. The adjustment type's A/P Request Type controls the payment date and the bank.

Name is the name printed on the check. This name is derived from the account's main person:

- If this person has an override mailing name, the first line of the override mailing name is used.
- If this person does not have an override mailing name, the person's primary name is used.

Payment Selection Status is the status of the check request. The values are:

- Not Selected for Payment before it's selected by A/P for payment
- Requested for Payment after it's selected by A/P
- Paid after it's paid by A/P
- Canceled if it's been canceled in A/P
- Hold if it's been held in A/P

Scheduled to Pay is the date on which the check is scheduled to be cut. This is equal to the adjustment date plus the Due Days on the adjustment type's A/P request type.

The following fields are provided in order to support a two-way interface with an A/P system. The system provides the inbound web service C1-UpdateAPCheckRequest that can be used to update these fields.

- **Payment Date** is the date on which the check was cut in A/P. This field is only populated after A/P cuts the check.
- **Paid Amount** is the amount of the check. This field is only populated after A/P cuts the check.
- **A/P Request ID** is the system-assigned, unique identifier of the A/P check request.
- **Payment Number** is the system-assigned number of the payment in A/P (this number typically appears on the printed check). This field is only populated after A/P cuts the check.

The address information in the bottom frame is the address to which the check is mailed. This is the billing address of the main customer linked to the account. After the adjustment is frozen, this information is automatically populated based on the address information for the account. This information is only modifiable when the Payment Selection Status is Not Selected for Payment.

Adjustment - Adjustment Portal

Select **Menu**, and then **Financial** , and then **Adjustment**, and then **Search** and navigate to the **Adjustment Portal** page to view additional information associated with an Adjustment.

Depending on your implementations requirements, you may define additional zones to appear on this Portal.

The contents of this section describe the zones that are available on this portal page.

Copyright © 2026, Oracle and/or its affiliates.

General Information

Adjustment ID, **Adjustment Info** are displayed.

Adjustment - Approval

This page only appears after an adjustment has started the approval process. Open this page using **Menu**, and then **Financial** , and then **Adjustment**, and then **Search** and navigate to the **Approval** page.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Adjustment Approvals for more information about the approval process.

The topics in this section describe the base-package zones that appear on the Approval Profile portal.

- Adjustment Information
- Approval Request
- Approval Request Log

Adjustment Information

The Adjustment Information zone contains display-only information about the adjustment.

Approval Request

The Approval Request zone shows the current and future approvers of an adjustment. This zone only appears if the adjustment is in the approval process.

If the current user has approval authority, the following functions are available:

- **Approve**: to approve the adjustment.
- **Reject**: to reject the adjustment.

Approval Request Log

The Approval Request Log zone contains the history of the adjustment's approval.

Financial - Adjustment Calculation Line Characteristics

This page displays the characteristics that are linked to a specific adjustment calculation line. The information on this page is for audit purposes only and cannot be changed.

Copyright © 2026, Oracle and/or its affiliates.

Open this page by clicking on the characteristics go to button on the calculation lines grid of the Adjustment - Main page.

Description of Page

Account is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

The **SA** is the service agreement whose value needs to be adjusted. Basic information about the service agreement appears adjacent.

Premise shows the service agreement's characteristic premise (i.e., the primary premise associated with the service agreement).

The **Adjustment Info** displays information to identify the adjustment, including amount, type, and status.

Adjustment ID displays the identification of the adjustment, and **Calc Line** displays the calculation line with which the characteristics are associated.

The **Description on Bill** displays the description of the calculation line that appears on the customer's bill.

The characteristics grid displays the **Characteristic Type**s and the **Characteristic Value**s associated with the calculation line.

> **NOT_SUPPORTED**
>
> For more information about characteristics, refer to Setting Up Characteristic Types and Their Values.

How and When To Use An Adjustment

> **Note**
>
> **A last resort.** Adjustments should be viewed as a last resort because you should correct the cause of errors, rather than the effect. For example, if a customer's bill is incorrect, you should fix the customer's rate / meter reads / contract terms and then perform a cancel / rebill. Issuing an adjustment to fudge the balance is really just that - a fudge.

The topics in this section describe how to perform common adjustment maintenance functions.

- How To Create A Transfer Adjustment
- How To Create A Calculated Adjustment
- How To Cancel An A/P Adjustment After It Has Been Selected By A/P
- How To Correct Normal Service Agreement Debt
- How To Apply Ad Hoc Fees To A Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

- How To Apply Interest To A Deposit
- How To Apply A Deposit To Outstanding Debt
- How To Refund A Deposit With A Check
- How To Write-Down Deposit Due
- How To Refund A Credit Balance With A Check
- How To Transfer A Credit Balance To A Service Agreement In Arrears
- How To Change The Age Of Debt
- How To Write-Down A Charitable Contribution Service Agreement Debt
- How To Write-Off Debt
- How To Cancel A Service Agreement's Budget
- How To Adjust A Service Agreement On A Budget
- How To Adjust An Installment Plan Service Agreement
- How To Use An Adjustment To Change The GL Distribution

How To Create A Transfer Adjustment

The following steps describe how to create a transfer adjustment on the Adjustment page.

> **Note**
>
> You cannot use an adjustment type with a calculated amount for transfer adjustments.

> **Warning**
>
> These steps only make sense in the context of the page used to maintain Adjustments. Refer to Adjustment - Main Information for the details.

To create a transfer adjustment:

- Specify the "transfer from" account and service agreement on the Adjustment - Main page.
- Specify the adjustment type and the amount and transfer to the Transfer Adjustment page.
- Specify the "transfer to" service agreement and click the **Generate** and **Freeze** buttons to generate and freeze the adjustments' financial transactions. Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account's next bill is completed. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

How To Create A Calculated Adjustment

The following steps describe how to create a calculated adjustment on the Adjustment page.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> These steps only make sense in the context of the page used to maintain Adjustments. Refer to Adjustment - Main Information for the details.

To create a calculated adjustment:

1. Open **Menu**, and then **Financial**, and then **Adjustment**, and then **Add** and specify the **Service Agreement** that will receive the adjustment.

2. Specify the **Adjustment Type** that calculates the amount, and enter a base **Amount**.

3. Click **Generate**. The **Generate** dialog box opens.

4. In the resulting dialog box, specify an **Accounting Date** and **Generation Reference Date**. Click **Calculate**.

5. Click **Freeze** to freeze the calculated transactions.

The **Calculation Date** that you supply is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version calculation group or bill factor value).

Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account's next bill is completed. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

How To Cancel An A/P Adjustment After It Has Been Selected By A/P

Adjustments that are interfaced to A/P (because A/P needs to cut a refund check) sometimes need to be canceled. You may cancel an A/P adjustment in the CIS system while its A/P request is Not Selected. If the A/P request is Requested for Payment or Paid you must first cancel the payment in A/P. Canceling the payment in A/P changes the A/P request in CIS to Canceled. At this point, you can cancel the adjustment in CIS.

How To Correct Normal Service Agreement Debt

If a customer is unhappy with the balance on a "normal" service agreement and you cannot fix the source of their unhappiness (i.e., the bills and payments), create an adjustment. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number if you are crediting the customer).

> **Note**
>
> **What's normal?** There really is no such thing as a "normal" service agreement. We use this term to reference service agreements used to bill for services like utility service, land leases, home warranties, security monitoring, etc. where the current amount is the same as the payoff amount.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> A service agreement that is part of a budget plan is NOT normal. Refer to How To Adjust A Service Agreement On A Budget **for more information**.

How To Apply Ad Hoc Fees To A Service Agreement

If you need to apply a fee (e.g., late payment charge, connection charge, NSF charge) to a "normal" service agreement, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a positive number so that the customer's balance increases.

> **Warning**
>
> Take care if you apply a fee to a service agreement that's on a budget. The adjustment affects current balance and payoff balance by the amount of the fee. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Apply Interest To A Deposit

> **NOT_SUPPORTED**
>
> You should only have to issue adjustments to calculate and refund interest if you need to override the system's automated procedures. Refer to Controls Interest Calculation for the details.

If you need to manually apply interest to a deposit, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a negative number (you are crediting the customer's deposit balance).

If you want to apply the interest to the customer's outstanding balance, refer to How To Apply A Deposit To Outstanding Debt.

If you want to cut a check for the deposit, refer to How To Refund A Deposit With A Check.

How To Apply A Deposit To Outstanding Debt

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures. Refer to Refunding Deposits for the details.

Applying a deposit is a two-step process:

- The deposit service agreement's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you apply the deposit balance, we recommend making the deposit's current balance equal to the amount to be transferred. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.
- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be applied to the customer's debt), use the Transfer Adjustment process to apply the deposit's credit balance to another service agreement's debit balance. The deposit service agreement is the transfer from service agreement; the delinquent service agreement is the transfer to service agreement. The adjustment amount should be a positive number. Refer to How To Create A Transfer Adjustment for more information.

Warning

Take care if you apply a deposit to a service agreement that's on a budget. The transfer adjustment affects current balance and payoff balance by the same amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Refund A Deposit With A Check

NOT_SUPPORTED

You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures. Refer to Refunding Deposits for the details.

Refunding a deposit with a check (cut by your A/P system) is a two step process:

- The deposit service agreement's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you refund the deposit balance, we recommend making the deposit's current balance equal to the amount to be refunded. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.
- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be refunded), issue a refund adjustment. The adjustment amount should be a

Copyright © 2026, Oracle and/or its affiliates.

positive number. This adjustment causes a check request to be interfaced to your A/P system.

How To Write-Down Deposit Due

If a customer doesn't pay their deposit and you don't expect them to, create an adjustment to write-down the current balance. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Refund A Credit Balance With A Check

If you need to refund a credit balance on a "normal" service agreement, issue a refund adjustment using the appropriate adjustment type. The adjustment amount should be a positive number. This adjustment causes a check request to be interfaced to your A/P system.

> **Warning**
>
> Take care if you refund a credit balance from a service agreement that's on a budget. The adjustment affects current balance and payoff balance by the amount of the refund. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Transfer A Credit Balance To A Service Agreement In Arrears

Use the Transfer Adjustment process to apply a "normal" service agreement's credit balance to another service agreement's debit balance. The credit service agreement is the transfer from service agreement; the delinquent service agreement is the transfer to service agreement. The adjustment amount should be a positive number. Refer to How To Create A Transfer Adjustment for more information.

> **Warning**
>
> Take care if you apply a credit to a service agreement that's on a budget. The transfer adjustment affects current balance and payoff balance by the transfer amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Change The Age Of Debt

If a service agreement's debt is older (or younger) than what it should be (for whatever reason), you should do the following.

Copyright © 2026, Oracle and/or its affiliates.

- Use the Transfer Adjustment process to re-age the debt. The transfer from and transfer to service agreements will be the same. The adjustment amount should be a negative number. Refer to How To Create A Transfer Adjustment for more information.
- After creating the transfer adjustment, drill into the transfer to adjustment and populate the arrears date with the day on which the debt should start aging (e.g., if the debt should be 7 days old, the arrears date should be the current date minus 7 days).

> **Warning**
>
> Be careful if you transfer debt from a service agreement that's on a budget. Before doing the transfer, you should make the customer's current balance equal their payoff balance.

How To Write-Down A Charitable Contribution Service Agreement Debt

If a customer doesn't want to contribute as much to the charity as the agreed to, create an adjustment to fix the current balance (charity service agreements shouldn't have a payoff balance as the "debt" is voluntary). Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Write-Off Debt

To write-off debt:

> **Warning**
>
> You don't have to do the tasks outlined below if you use the Write Off - Main page. This page does these tasks for you.

- Create a write-off service agreement using the Start Service page (you could also use the Service Agreement page to do this).
- Use the Transfer Adjustment process to transfer the delinquent debt to the to the write-off service agreement. The original service agreement is the transfer from service agreement; the write-off service agreement is the transfer to service agreement. The adjustment amount should be a negative number. Refer to Transfer Adjustments for more information.

> **Warning**
>
> Take care if you transfer debt from a service agreement that's on a budget. Before doing the transfer, you should make the service agreement's current balance equal their payoff balance.

Copyright © 2026, Oracle and/or its affiliates.

How To Cancel A Service Agreement's Budget

> **Warning**
>
> You don't have to do the tasks outlined below if you use the Account - Budget panel. This panel does these tasks for you.

Canceling a service agreement's budget involves the following steps:

- Update the service agreement to indicate the service agreement is no longer on a budget.
- Issue an adjustment to make the service agreement's current balance equal the payoff balance.

> **Note**
>
> **Credit and collections (C&C).** You can change the arrears date on the adjustment that synchronizes current and payoff balance to reflect the true age of the debt. You do this on the financial transaction associated with the adjustment. If you do this, C&C will monitor the old debt accordingly.

How To Adjust A Service Agreement On A Budget

If you need to adjust the debt on a service agreement that's on a budget plan, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you need to adjust both balances equally, choose an adjustment type that affects both.

How To Adjust An Installment Plan Service Agreement

If you need to adjust the debt on a service agreement that's used to bill a total amount in installments, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.

If you need to adjust both balances equally, choose an adjustment type that affects both.

Copyright © 2026, Oracle and/or its affiliates.

How To Use An Adjustment To Change The GL Distribution

Assume you have a frozen bill segment that affected an incorrect GL distribution code (e.g., it booked revenue to commercial - time-of-use - revenue and it should have booked it to industrial - time-of-use - revenue. You have two ways to correct such a situation:

- You could correct the cause of the incorrect distribution code (probably incorrect GL information on a calculation rule/pricing rule) and cancel / rebill the bill segment. Refer to Cancel / Rebill for more information about this method.
- You could create an adjustment that does not impact the customer's balance but does have a GL entry. This GL entry would reverse the effect of the original distribution code and have an offsetting entry to the correct distribution code. The following points explain how to do this:
- Choose an adjustment type that only impacts the GL (i.e., the adjustment's adjustment type has an FT algorithm that doesn't impact the customer's current or payoff balance, it only impacts the GL).
- Enter an adjustment amount of zero.
- Generate the adjustment.
- Drill into the adjustment's financial transaction and enter two (or more) GL distribution rows that reflect the redistribution of the amounts.
- Return to the adjustment and freeze it.

> **Note**
>
> **Adjusting statistical amounts.** You can use the above approach to change the statistical amount that is interfaced to the general ledger. The statistical amount is simply another column on the financial transaction's GL distribution rows.

Interfacing Adjustments From External Sources

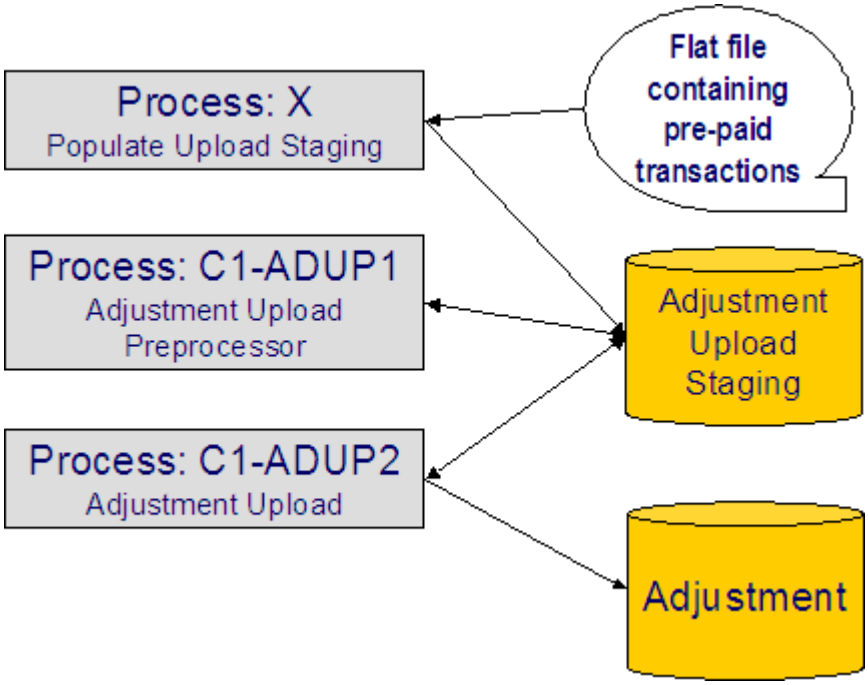
The topics in this section describe how adjustments are uploaded from an external source.

- Interfacing Adjustments
- Suspense Adjustments
- Maintaining Adjustment Staging Control
- Maintaining Adjustment Upload Staging

Interfacing Adjustments

The following diagram illustrates the processes involved in the uploading of adjustments into the system.

Copyright © 2026, Oracle and/or its affiliates.



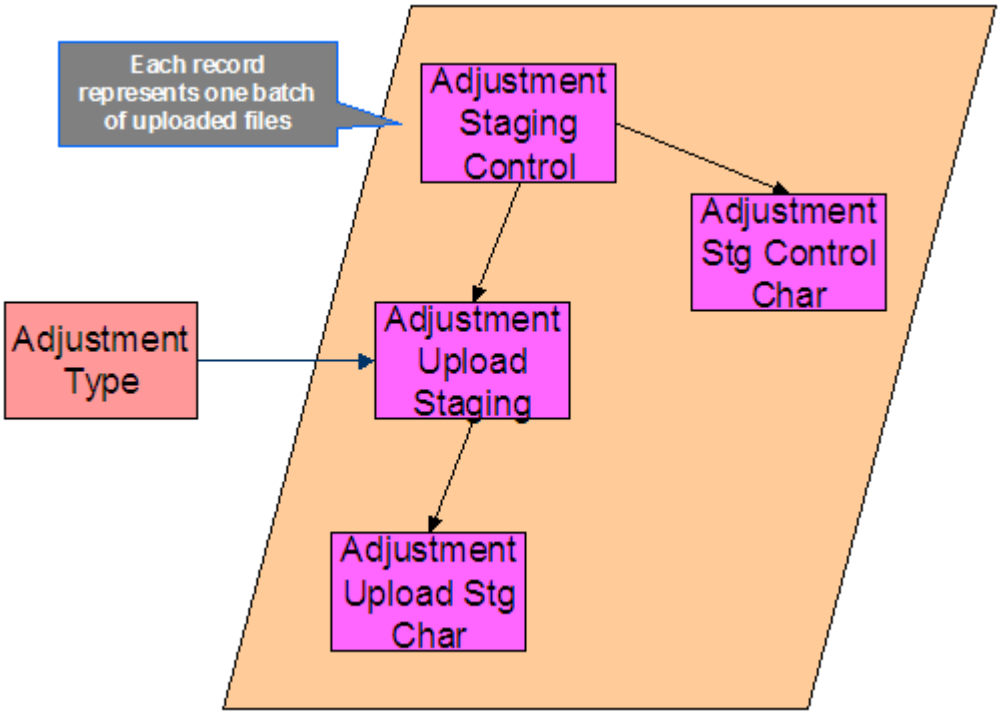
The topics in this section describe how these processes work.

- Process X - Populate Adjustment Upload Records
- Process C1-ADUP1 - Preprocess Adjustment Uploads
- Process C1-ADUP2 - Upload Adjustments

Process X - Populate Adjustment Upload Records

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).

Copyright © 2026, Oracle and/or its affiliates.



The topics in this section describe each of these tables.

- Adjustment Staging Control
- Adjustment Staging Control Characteristic
- Adjustment Upload Staging
- Adjustment Characteristic Upload Staging
- The Lifecycle of an Adjustment Staging Control Record
- The Lifecycle of an Adjustment Upload Staging Record

Adjustment Staging Control

You must create an adjustment staging control record for each batch of adjustments to be uploaded into the system. The name of this table is CI_ADJ_STG_CTL. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This is the unique identifier of the adjustment staging control record. This key is a sequential number so you can use a database function to assign the value when populating the table.

Copyright © 2026, Oracle and/or its affiliates.

CRE_DTTM	15	Y	DateTime	This is the date / time on which the adjustment staging control record was created. This must be populated with the current date / time.
ADJ_STG_CTL_STATUS_FLG	4	Y	A/N	This must be set to P for Pending.
ADJ_STG_UP_REC_CNT	10	Y	N	This is the total number of adjustment upload staging records that are linked to this adjustment staging control.
TOT_ADJ_AMT	13.2	Y	N	This column must equal the sum of adjustment amounts on the adjustment upload staging records that are linked to this adjustment staging control.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code in the system. Refer to Defining Currency Codes for more information.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.

Adjustment Staging Control Characteristic

Copyright © 2026, Oracle and/or its affiliates.

You must create an adjustment staging control characteristic record for each characteristic that you would like to link to the adjustment staging control. The name of this table is CI_ADJ_STG_CTL_CHAR. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_CTL record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for adjustment staging control. Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to 10 unless you have multiple values for a given adjustment staging control and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is predefined.
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is ad-hoc or file location.
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is foreign key reference. Up to five columns of 50 bytes each are provided to accommodate compound keys.

Adjustment Upload Staging

You must create an adjustment upload staging record for each adjustment you want to upload. The name of this table is CI_ADJ_STG_UP. The following describes each column on this table.

Copyright © 2026, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This is the unique identifier of the adjustment upload staging record. This key is a sequential number so you can use a database function to assign the value when populating the table.
ADJ_STG_CTL_ID	12	Y	N	The ID of the adjustment staging control that is linked to this adjustment upload staging record.
ADJ_TYPE_CD	8	Y	A/N	This must correspond to the prime key of one of your adjustment types. Refer to Setting up Adjustment Types for more information.
ADJ_STG_UP_STATUS_FLG	4	Y	A/N	This must be set to P for Pending.
CREATE_DT	10	Y	Date	The date when the adjustment occurred.
ADJ_AMT	13.2	Y	N	The amount of the adjustment.
ADJ_SUSPENSE_FLG	4	Y	N	This must be set to NSUS for Not In Suspense.

Copyright © 2026, Oracle and/or its affiliates.

SA_ID	10	N	A/N	This must correspond to a valid service agreement in the system. If you leave this blank and opt to let the system find the SA during adjustment upload preprocessing, a Determine SA algorithm must be plugged in on the associated adjustment type. This plug-in is meant to derive the SA ID based on supplied miscellaneous information - e.g. information supplied via adjustment characteristic upload staging.
ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.
SUSPENSE_ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.

Copyright © 2026, Oracle and/or its affiliates.

Adjustment Characteristic Upload Staging

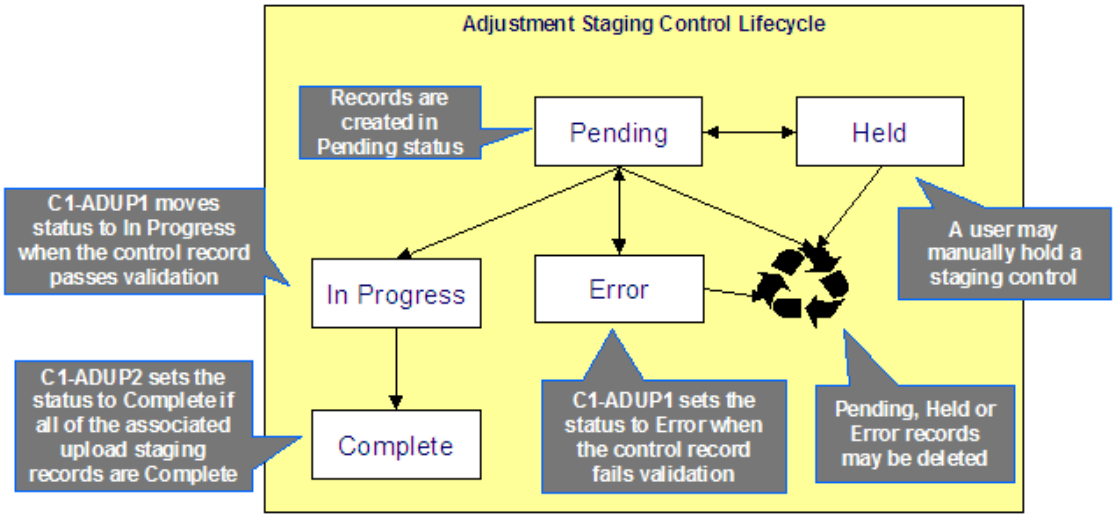
You must create an adjustment characteristic upload staging record for each characteristic that you would like to link to the adjustment upload staging. The name of this table is CI_ADJ_STG_UP_CHAR. The following describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_UP record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for adjustment Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to 10 unless you have multiple values for a given adjustment upload staging and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is predefined.
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is ad-hoc or file location.
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is foreign key reference. Up to five columns of 50 bytes each are provided to accommodate compound keys.

Copyright © 2026, Oracle and/or its affiliates.

The Lifecycle of an Adjustment Staging Control Record

The following diagram shows the possible lifecycle of an adjustment staging control record.

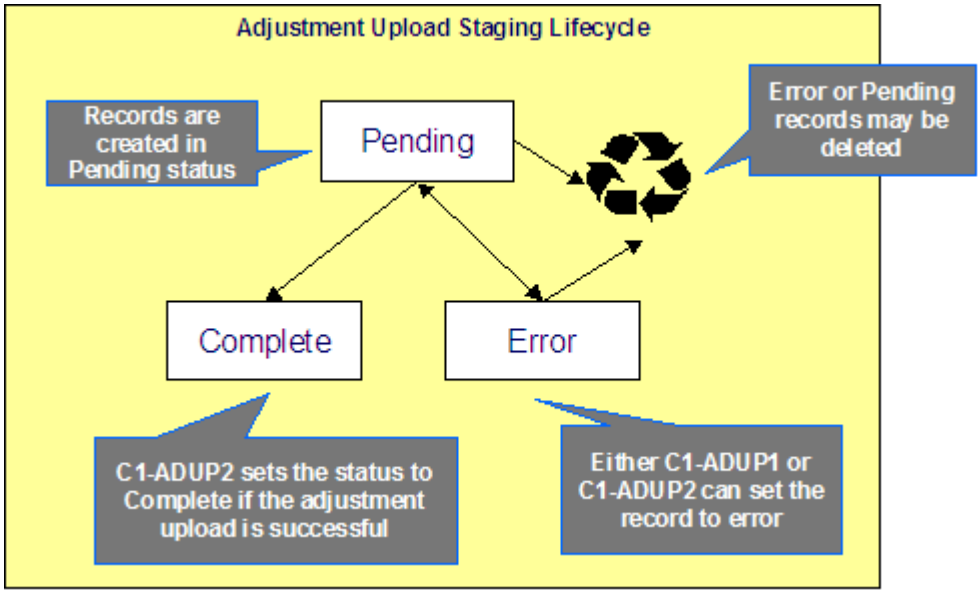


- Pending. An adjustment staging control record is created in this state. The C1-ADUP1 process selects pending adjustment staging control records for validation.
- In Progress. The C1-ADUP1 process sets a pending record to in progress when the totals on the adjustment staging control are successfully validated against the totals from the associated adjustment upload staging records.
- Complete. The C1-ADUP2 process sets the adjustment staging control's status to complete when all adjustment upload staging records linked to the adjustment staging control are complete.
- Error. The C1-ADUP1 process sets a pending record to error if the adjustment staging control fails validation. The status may be set back to pending after the error is fixed.
- Held. The held status is available for situations where you want to prevent or delay the upload of a batch of adjustment staging records. The status may be set back to pending when the batch of records is ready for upload.
- Pending, error and held records may be deleted.

The Lifecycle of an Adjustment Upload Staging Record

The following diagram shows the possible lifecycle of an adjustment upload staging record.

Copyright © 2026, Oracle and/or its affiliates.

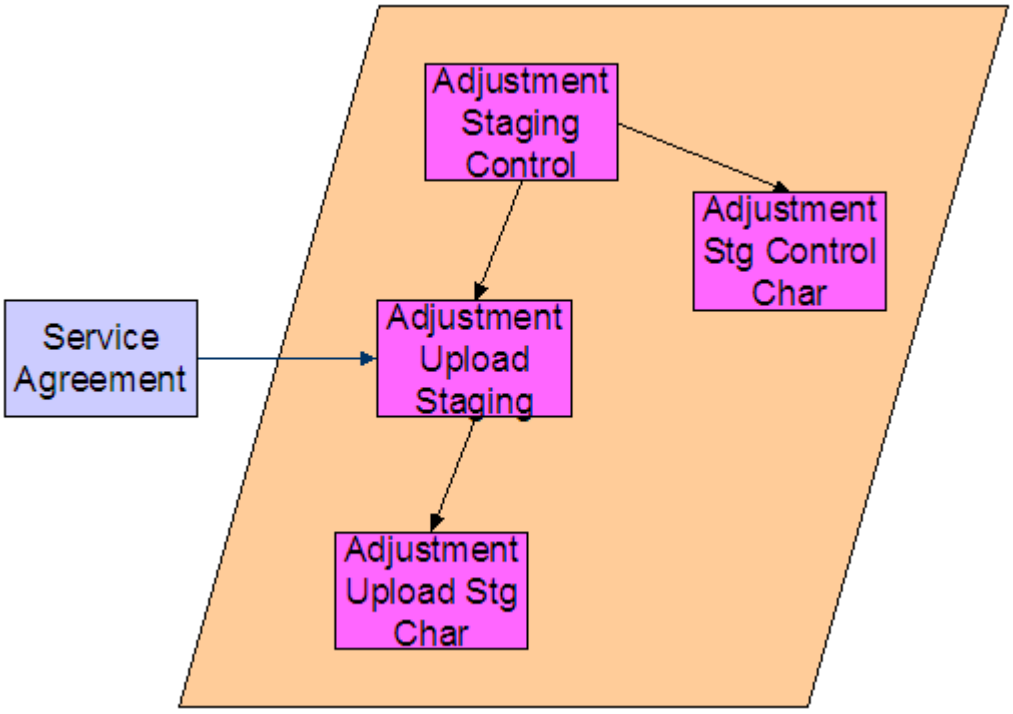


- Pending. Adjustment upload records are created in this state. The C1-ADUP2 process selects pending adjustment upload records and creates adjustments for each of them.
- Complete. The C1-ADUP2 process sets a pending record to complete if an adjustment is successfully created.
- Error. Either C1-ADUP1 or C1-ADUP2 process sets pending record to error when it encounters an error during the upload process. The status may be set back to pending after the error is fixed.
- Pending and error records may be deleted.

Process C1-ADUP1 - Preprocess Adjustment Uploads

The background process (C1-ADUP1) validates adjustment staging control records and populates service agreement IDs on adjustment upload staging records that do not specify an SA ID.

Copyright © 2026, Oracle and/or its affiliates.



- Phase 1 - Validate Adjustment Staging Controls
- Phase 2 - Populate SA ID

Phase 1 - Validate Adjustment Staging Controls

The following points describe, at a high level, the first phase of the adjustment upload pre-process:

- For each Pending adjustment staging control,
- Check that the record count on the adjustment staging control record equals the number of adjustment upload staging records that are linked to the adjustment staging control.
- Check that the total adjustment amount on the adjustment staging control record equals the sum of the adjustment amounts from the adjustment upload staging records that are linked to the adjustment staging control.
- If the adjustment staging control passes validation, set its status to In Progress. Otherwise, set the status to Error. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment staging control errors. (Complete any outstanding To Do entries for the adjustment staging control before creating a new To Do entry.)
- If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment staging control.

> **Note**
>
> You can fix errors by going to the To Do entry and drilling into the adjustment staging control page. Don't forget to change the adjustment staging control's status back to Pending after fixing the error.

Copyright © 2026, Oracle and/or its affiliates.

Phase 2 - Populate SA ID

The following points describe, at a high level, the second phase of the adjustment upload pre-process:

- For each Pending adjustment upload staging that is linked to an In Progress adjustment staging control AND does not have the SA ID,
- Execute the Determine SA algorithm that is plugged in on the adjustment type.
- If the algorithm returns a valid SA ID, stamp that SA ID onto the adjustment upload staging. If the algorithm also returns an indication that the adjustment is to be put into suspense, set the suspense flag on the adjustment upload staging to In Suspense. Refer to Suspense Adjustments for more information on how suspense adjustments are handled.
- If the algorithm returns an error, set the adjustment upload staging record's status to Error. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment upload staging errors. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
- If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

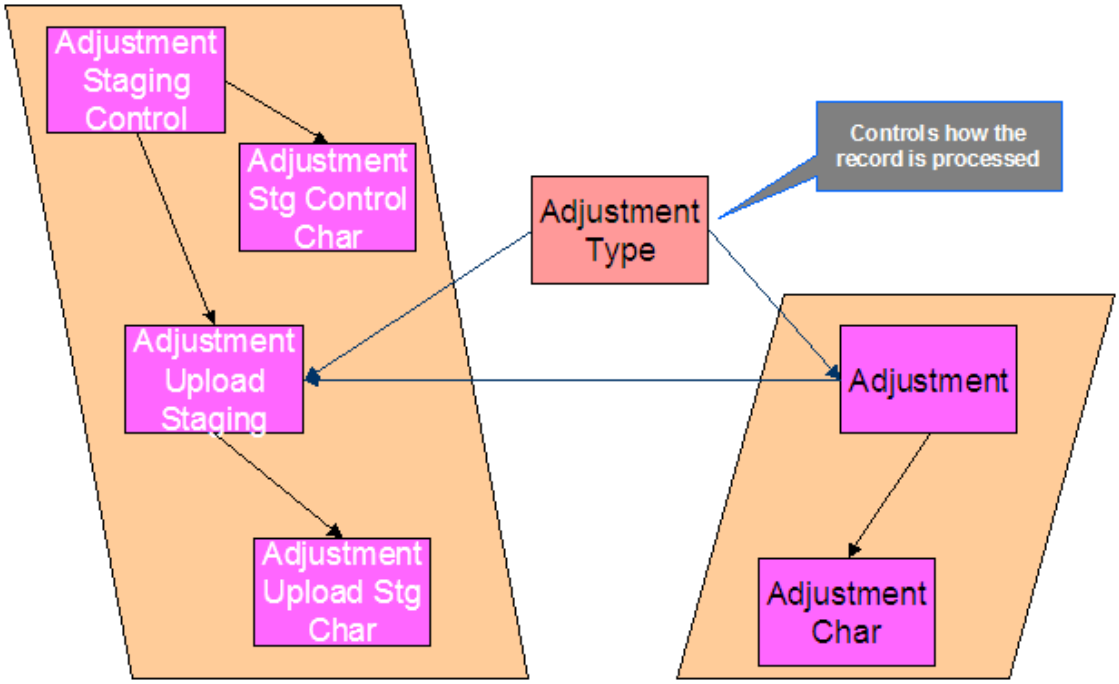
> **Note**
>
> You can fix errors by going to the To Do entry and drilling into the adjustment upload staging page. Don't forget to change the adjustment upload staging's status back to Pending after fixing the error.

Process C1-ADUP2 - Upload Adjustments

The background process (C1-ADUP2) creates adjustments for all adjustment upload staging records that are stamped with an SA ID.

The following diagram and section describe, at a high level, the processing done in the C1-ADUP2 background process.

Copyright © 2026, Oracle and/or its affiliates.



- For each Pending adjustment upload staging that has an SA ID,
- Add a frozen adjustment using the amount, creation date and adjustment type on the staging record.

> **Note**
>
> If the type of adjustment being uploaded is one that is calculated, the algorithm that's plugged in on the adjustment type will calculate the adjustment amount and generate associated calculation lines (if applicable). See Calculated Adjustments for more information.

- If adjustment creation is successful, update the adjustment ID on the staging record and set the staging record's status to Complete.
- If adjustment creation results in an error, set the staging record's status to Error. Create a To Do entry using the To Do type on which this batch process (C1-ADUP2) is defined as creation process. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
- If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

Suspense Adjustments

- What Are Suspense Adjustments?
- How Are Suspense Adjustments Resolved?

Copyright © 2026, Oracle and/or its affiliates.

What Are Suspense Adjustments?

When the adjustment upload preprocessor is unable to identify a valid SA ID, it can post the adjustment to a suspense SA. This is similar to how a payment is posted to a suspense account when a valid account ID could not be identified during payment upload.

Putting adjustments in suspense is optional. Therefore, this logic sits in a plug-in spot. If suspense adjustments are applicable to you, you must plug in your suspense logic in a Determine SA algorithm on Adjustment Type.

How Are Suspense Adjustments Resolved?

A background process can be run periodically to automatically resolve suspense adjustments. Since rules for resolving suspense may vary, this logic sits in a plug-in spot. You must plug in your resolve logic in a Resolve Suspense algorithm on Adjustment Type.

- Process C1-ADURS - Resolve Suspense Adjustments

Process C1-ADURS - Resolve Suspense Adjustments

The batch process refers to the background process (C1-ADURS) that resolves suspense adjustments.

For each adjustment upload staging that is In Suspense, execute the Resolve Suspense algorithm that is plugged in on the adjustment type.

Maintaining Adjustment Staging Control

Use this page to add, view or modify an adjustment staging control record. Open this page using **Menu**, and then **Financial**, and then **Adjustment Staging Control**, and then **Search**

Description of Page

Adjustment Staging Control ID is the unique identifier of the adjustment staging control record.

Create Date/Time is when the adjustment staging control was created. This field is protected after the record is saved.

Number of Staging Records is the number of adjustment upload staging records that are linked to this adjustment staging control record.

Total Adjustment Amount is the total of adjustment amounts from the adjustment upload staging records that are linked to this adjustment staging control record.

Enter the **Currency** for the total adjustment amount on the adjustment staging control. This is also the currency of the adjustment amounts on the associated adjustment upload staging records.

Status shows the state of the adjustment staging control. Possible values are: Pending, In Progress, Complete, Error, and Held.

Copyright © 2026, Oracle and/or its affiliates.

A **Hold** button appears adjacent to the status description if **Status** is Pending. Click this button to set the status to Held. The adjustment upload process will ignore staging controls in this state.

A **Pend** button appears adjacent to the status description if **Status** is either Held or Error . Click this button to set the status to Pending. An error message box also appears below the status description if **Status** is Error.

Use the **Characteristics** collection to capture miscellaneous information about an adjustment staging control.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

> **Note**
>
> You can only choose characteristic types defined as permissible on an adjustment staging control. Refer to Setting Up Characteristic Types & Their Values for more information.

Maintaining Adjustment Upload Staging

Use this page to add, view, or modify an adjustment upload staging record. Open this page using **Menu**, and then **Financial**, and then **Adjustment Upload Staging**, and then **Search.**

Description of Page

Adjustment Upload ID is the unique identifier of the adjustment upload staging record.

Adjustment Staging Control ID is the identifier of the adjustment staging control to which the adjustment upload staging is linked.

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger.

> **Note**
>
> You can only choose certain adjustment types. The service agreement's SA type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the SA type.

Enter an **Adjustment Amount**.

Creation Date is the date when the adjustment occurred. This will be the creation date on the uploaded adjustment. This field is protected after the record is saved.

Service Agreement indicates the SA to which the adjustment was posted. If the adjustment is in suspense, the label shows **Suspense SA** instead.

Adjustment is a reference to the adjustment that got created from the upload.

Copyright © 2026, Oracle and/or its affiliates.

Status shows the state of the adjustment upload staging. Possible values are: Pending, Complete, and Error.

If **Status** is Error, a **Pend** button appears adjacent to the status description. In addition, an error message box displays below the status description.

If the adjustment is in suspense, a message appears to indicate the condition and the **Suspense Adjustment** is shown. A different message appears when the suspense is resolved.

Use the **Characteristics** collection to keep miscellaneous information about an adjustment upload staging.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

> **Note**
>
> **Default Note.** An adjustment upload staging's characteristics default from the adjustment type.

Credit & Collection

The system periodically monitors how much your customers owe to ensure they haven't violated your collection criteria. When a violation is detected, the system generates the appropriate events (e.g., letters, disconnect field activities, To Do entries, write-off outstanding debt, etc.). This section describes how to manage your credit & collections processing.

> **Note**
>
> In the section, The Big Picture Of Credit & Collection (C&C), we describe how to set up the control tables that automate most of your credit & collections processing. If you set these tables up properly, the system will manage most of your collection, severance and write-off processes; little or no user intervention should be necessary. You will only access the C&C pages when you need to override what the system does.

- The Big Picture Of Credit & Collection
- Collection Process Maintenance
- Severance Process Maintenance
- Write-Off Process Maintenance
- Writing Off Uncollectable Debt
- Collection Referral
- Payment Arrangements and Pay Plans

Copyright © 2026, Oracle and/or its affiliates.

The Big Picture Of Credit & Collection

The topics in this section provide background information about a variety of credit & collections issues.

- Automating Your C&C Activities
- The Lifecycle Of A Collection Process And Its Events
- The Collection Process Background Processes
- The Lifecycle Of A Severance Process And Its Events
- The Severance Process Background Processes
- The Lifecycle Of A Write-Off Process And Its Events
- The Write-Off Process Background Processes

Automating Your C&C Activities

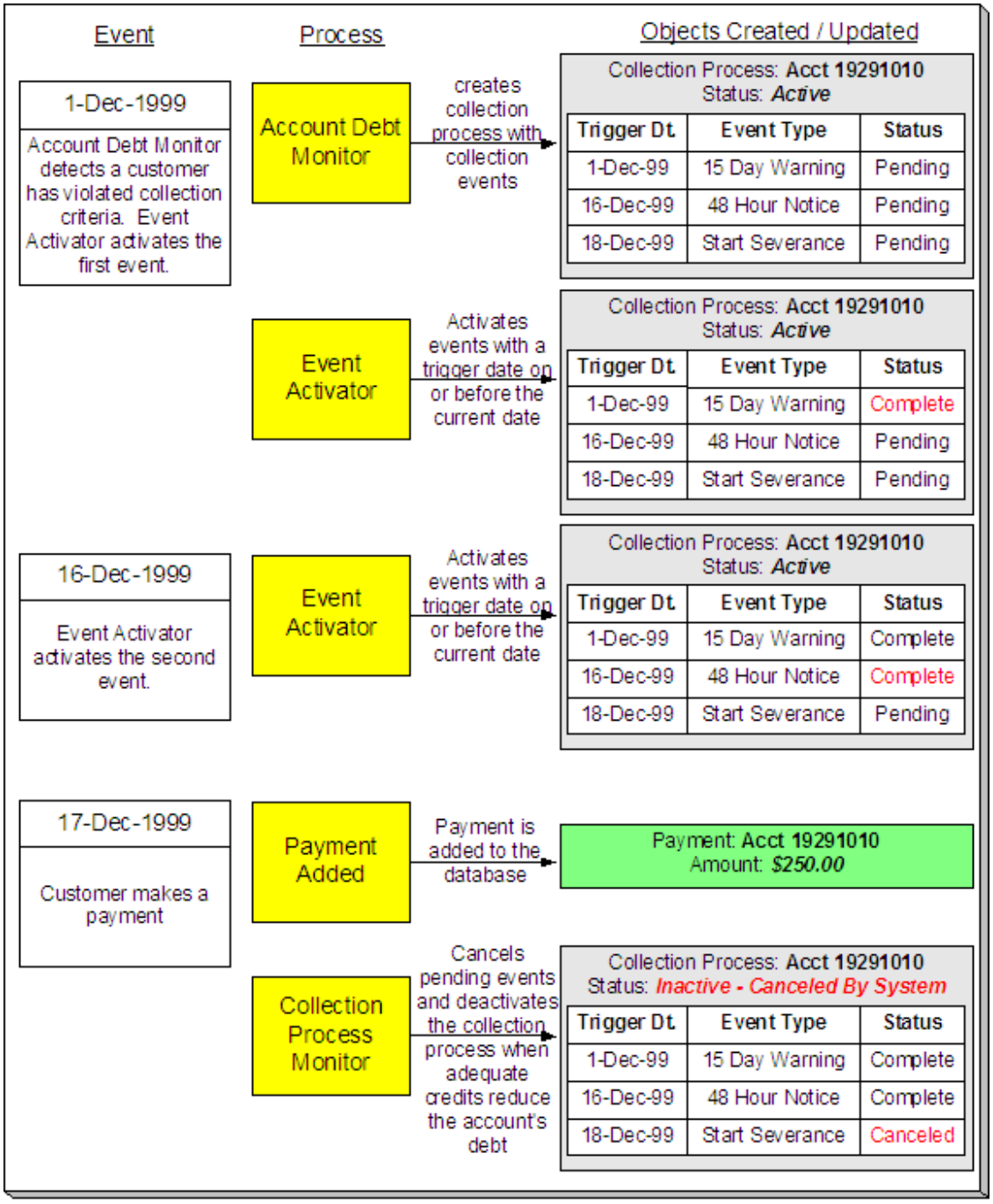
NOT_SUPPORTED

Refer to The Big Picture Of Credit & Collections (C&C) to understand how the system monitors overdue debt and initiates the appropriate collection, severance and write-off events when a customer exceeds your debt tolerances.

The Lifecycle Of A Collection Process And Its Events

The following diagram illustrates the lifecycle of a typical collection process and its collection events:

Copyright © 2026, Oracle and/or its affiliates.



The preceding diagram is referenced in the following topics.

- What Is A Collection Process?
- How Are Collection Processes Created?
- How Are Collection Events Completed?
- How Are Collection Processes (and their Events) Canceled?
- What Happens If A Payment Is Not Received?

Copyright © 2026, Oracle and/or its affiliates.

What Is A Collection Process?

A collection process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific service agreements that contributed to the delinquent debt.

How Are Collection Processes Created?

A collection process and its events may be created as follows:

- The Account Debt Monitor is a background process that creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The collection process template associated with the collection process controls the number and type of events.
- A user may create an ad hoc collection process at will. The collection process template controls the number and type of events linked to the collection process.

In the above diagram, the event on 1-Dec-99 shows the creation of a collection process with three collection events. Notice that the events all exist in the Pending state. The events will exist in this state until their trigger date is on or before the current date.

NOT_SUPPORTED

Refer to How Does The Account Debt Monitor Work? to understand how the system monitors overdue debt and initiates the appropriate collection process when a customer exceeds your debt tolerances. Refer to The Big Picture Of Collection Events for more information about a collection process' events.

When you enable the Control Central alert algorithm, C1-COLLPR-AC, an alert displays when an account has an active collection process. This algorithm is plugged-in on the installation record.

How Are Collection Events Completed?

The event activator runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the Complete state.

NOT_SUPPORTED

Refer to The Big Picture Of Collection Events for more information about the various types of collection events and what happens when they are completed.

Copyright © 2026, Oracle and/or its affiliates.

How Are Collection Processes (and their Events) Canceled?

NOT_SUPPORTED

Refer to How Are Collection Processes Cancelled for a description of how and when collection processes are automatically cancelled by the system.

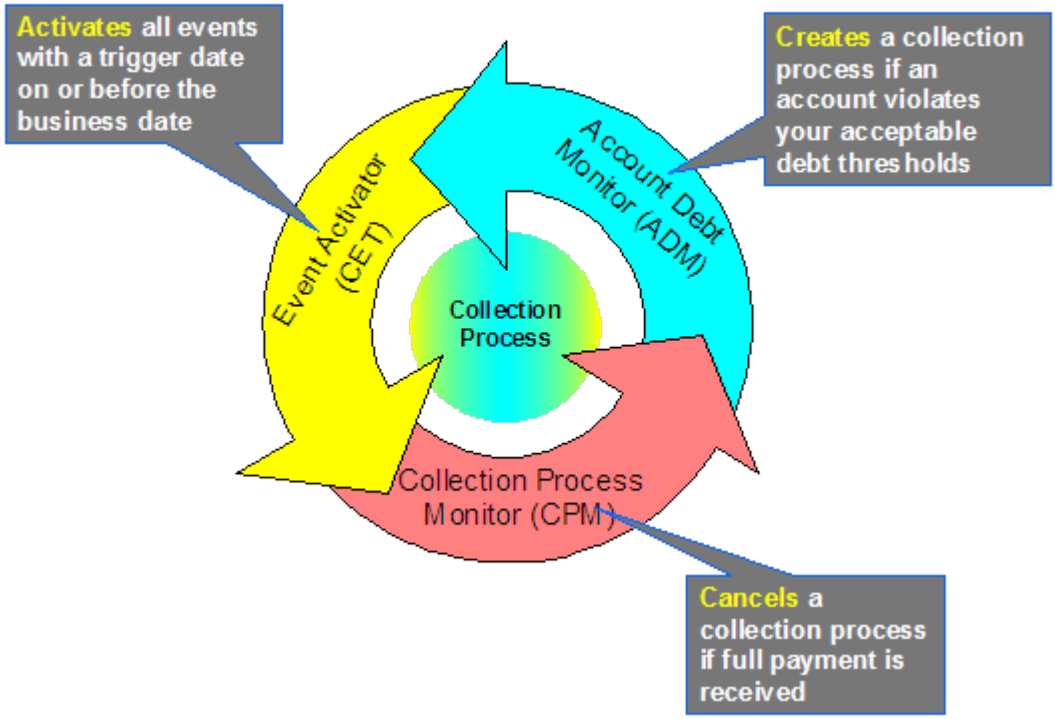
Besides the automated cancellation process, a user may cancel a collection process at will. Refer to How To Cancel A Collection Process for more information.

What Happens If A Payment Is Not Received?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the process, a severance process is initiated for one or more service agreements. Refer to The Last Collection Event Should Sever Service Agreement(s) for more information.

The Collection Process Background Processes

There are three background processes that automate the creation, activation, and cancellation of collection processes:



Copyright © 2026, Oracle and/or its affiliates.

- Account Debt Monitor
- Collection Event Activator
- Collection Process Monitor

Account Debt Monitor

The Account Debt Monitor (ADM) creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The number and type of events is defined on the collection process's template.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

> **Note**
>
> Monitoring account debt is usually done based on a schedule triggered by system events such as a bill or canceled payment. If too much time has gone by since the last debt review, the account can be selected for debt monitoring as well. ADM has an optional input parameter **Include Minimum Days Review**. When set, accounts that have gone too long since their last review are selected along with those scheduled to be reviewed. The minimum days between review is defined on the customer class
>
> A separate background process (ADM2) only selects accounts based on the minimum days. This process can be used when the option to include minimum days on ADM has not been set.
>
> ADM2 does not have an option to be run by CIS Division.

In the above diagram, the account debt monitor creates a collection process (with 3 events) on 1-Dec-99. Notice that the events all exist in the Pending state. The events will exist in this state until their trigger date is on or before the current date.

> **NOT_SUPPORTED**
>
> Refer to How Does The Account Debt Monitor Work for more information.

Collection Event Activator

The Collection Event Activator (CET) runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

Copyright © 2026, Oracle and/or its affiliates.

In the above diagram, the collection event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the Complete state after they are activated.

Collection Process Monitor

The Collection Process Monitor (CPM) runs periodically (at least daily). This process reviews a collection process whenever one of its service agreements' debt is reduced.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

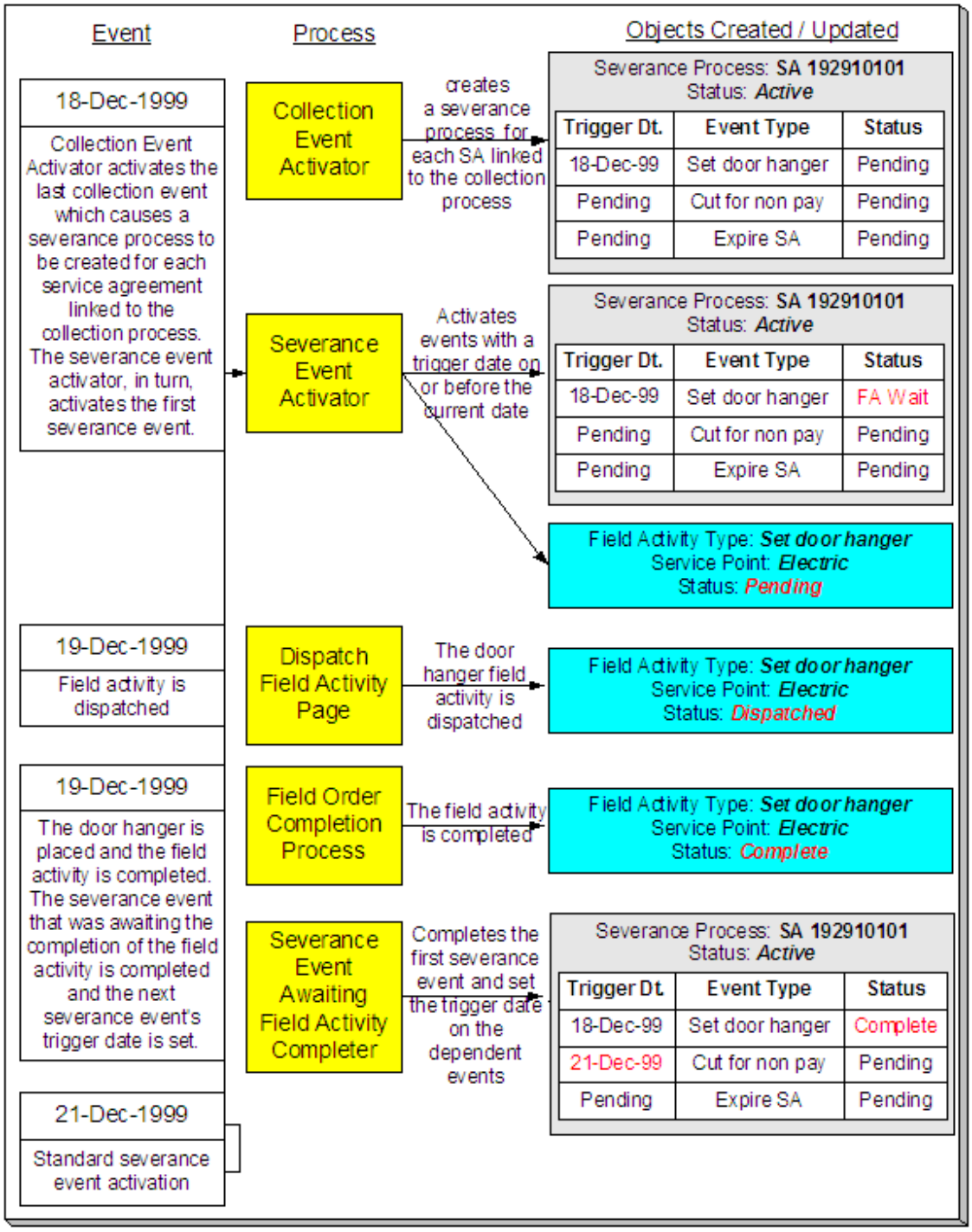
At review time, the Collection Process Monitor determines if it can cancel the collection process (because the customer's debt is no longer in a bad state). Refer to How Does The Collection Process Monitor Work for details describing how this process works.

In the above diagram, the collection process monitor cancels the process on 17-Dec-99. Notice that the events are moved to the Cancelled state and the process becomes Inactive.

The Lifecycle Of A Severance Process And Its Events

The following diagram illustrates the first part of the lifecycle of a typical severance process and its events:

Copyright © 2026, Oracle and/or its affiliates.



The preceding diagram is referenced in the following topics.

- What Is A Severance Process?
- How Are Severance Processes Created?
- How Are Severance Events Completed?
- How Are Severance Processes (and their Events) Canceled?
- What Happens If The Customer Does Not Pay?

Copyright © 2026, Oracle and/or its affiliates.

What Is A Severance Process?

A severance process is a series of events (e.g., letters, To Do entries, field activities) that lead to the severance of a service agreement. A separate severance process is required for each service agreement to be severed.

How Are Severance Processes Created?

Severance processes may be created as follows:

- The Collection Event Activator creates a severance process when it completes a "start severance process" collection event. The severance process has one or more severance event(s). The severance process template associated with the severance process controls the number and type of events.
- A user may create an ad hoc severance process at will. The severance process template defined when the severance process is created controls the number and type of events linked to the severance process.

> **Note**
>
> **Manual Creation**. A user can create a severance process for an account that does not qualify to be on severance according to the cancel criteria algorithm. For example, perhaps your cancel criteria algorithm cancels a severance process when the account's debt falls below a threshold amount. A user can create a severance process for an account whose debt is already below this threshold. Because cancellation is real time, there is no action that will cause this severance process to be canceled. When a manual severance process is created, the system executes the appropriate cancellation criteria algorithm. If the algorithm indicates that the system would have canceled this severance process, a warning is issued. Refer to How Are Severance Processes Cancelled for more information about cancellation algorithms.

In the above diagram, the event on 18-Dec-99 shows the creation of a severance process with three collection events. Notice that only the first event has a trigger date, this is because the later events' trigger date can only be determined after the first event is completed. Also, notice that the events all exist in the Pending state. The events will exist in this state until their trigger date is on or before the current date.

> **NOT_SUPPORTED**
>
> Refer to How Are Collection Events Completed? for more information about the completion of collection events.
>
> When you enable the Control Central Alert installation algorithm, C1-SEVPR-ACT, an alert displays when an account has an active severance process. This algorithm is plugged-in on the installation record.

Copyright © 2026, Oracle and/or its affiliates.

How Are Severance Events Completed?

The Severance Event Activator runs periodically (at least daily). It looks for severance events with a trigger date on or before the current date. For each such event, the system executes its activity. The type of severance event dictates what happens to the severance event's state. Refer to Field Events Versus Office Events for the details.

In the above diagram, the event activator activates a pending event on 18-Dec-99. Take note of the state of the severance event (it becomes Awaiting Field Activity Completion).

NOT_SUPPORTED

Refer to The Big Picture Of Severance Events for more information about the various types of severance events and what happens when they are completed.

How Are Severance Processes (and their Events) Canceled?

NOT_SUPPORTED

Refer to How Are Severance Processes Cancelled for a description of how the system automatically cancels a severance process.

Note

Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no equivalent of the collection process monitor for severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered. The logic to cancel the field activities is in an algorithm plugged in on the installation record.

NOT_SUPPORTED

Refer to Setting Up Severance Process Templates for more information about severance process templates. Refer to Designing Your Reconnection Procedures for more information about how severance processes are used to reconnect services.

Besides the automated cancellation process, a user may cancel a severance process at will.

Copyright © 2026, Oracle and/or its affiliates.

What Happens If The Customer Does Not Pay?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the severance process, the service agreement is stopped (as long as you've setup your severance process to include an Expire Service Agreement event as the last event). The expiration of the last service agreement will, in turn, trigger a final bill. If the account doesn't pay the final bill, a write-off process will be started.

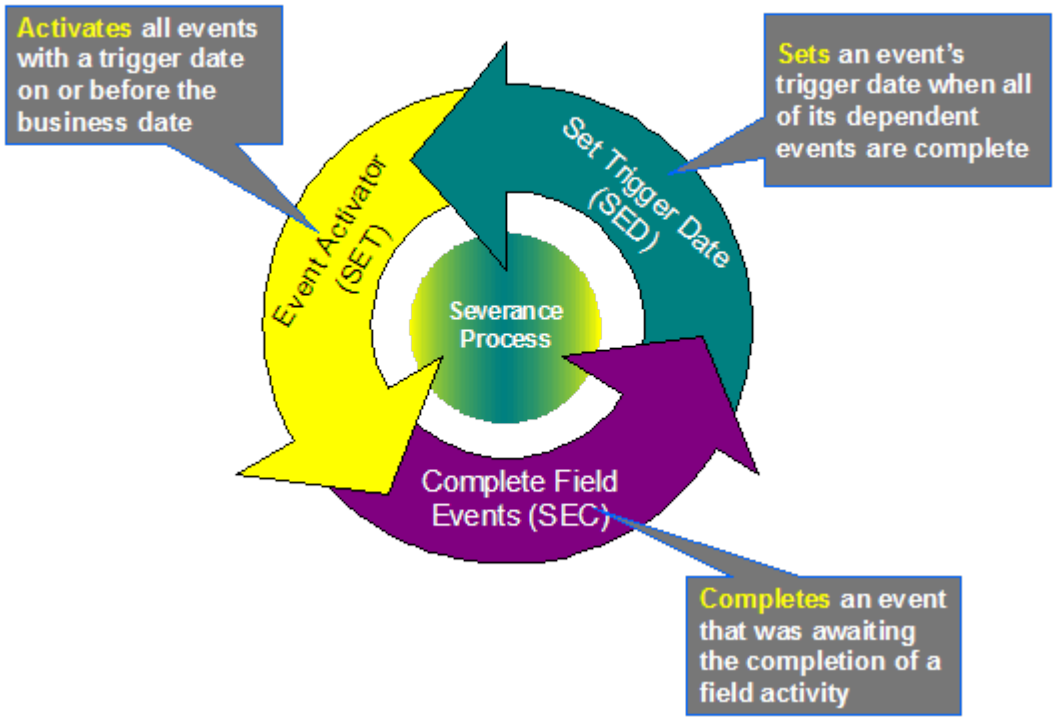
> **Note**
>
> **Final bills.** The account will be setup for final billing when the LAST service agreement linked to the account is stopped. Refer to Finalizing Pending Stops for more information about how the system changes the account's bill cycle when the last service agreement is stopped.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Write Osff Processing for more information about the write off process.

The Severance Process Background Processes

The Collection Event Activator creates a severance process when it completes a "start severance process" collection event. The severance process has one or more severance event(s). The severance process template associated with the severance process controls the number and type of events. There are 3 background processes that automate the activation and cancellation of severance events:

Copyright © 2026, Oracle and/or its affiliates.



- Severance Event Activator
- Completing Severance Events That Are Dependent On The Completion Of A Field Activity
- Set Trigger Date

Severance Event Activator

The Severance Event Activator (SET) runs periodically (at least daily). It looks for severance events with a trigger date on or before the current date. For each such event, the system executes its activity. The type of severance event dictates what happens to the severance event's state.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

In the above diagram, the event activator activates a pending event on 18-Dec-99. Take note of the state of the severance event (it becomes Awaiting Field Activity Completion).

Completing Severance Events That Are Dependent On The Completion Of A Field Activity

The Complete Field Events (SEC) process runs periodically (at least daily). It looks for severance events that are dependent on the completion of field activities before they can be completed. The system completes each such event when the field activities linked to the event are complete.

Copyright © 2026, Oracle and/or its affiliates.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

In the above diagram, this process completes the event that was Awaiting Field Activity Completion on 19-Dec-99. Take note of the state of the severance event (it becomes Complete).

Set Trigger Date

The Set Trigger Date (SED) process runs periodically (at least daily). It looks for severance events that are dependent on the completion of other events before their trigger date can be set. The system sets the trigger date of each such event when all of its dependent events are complete.

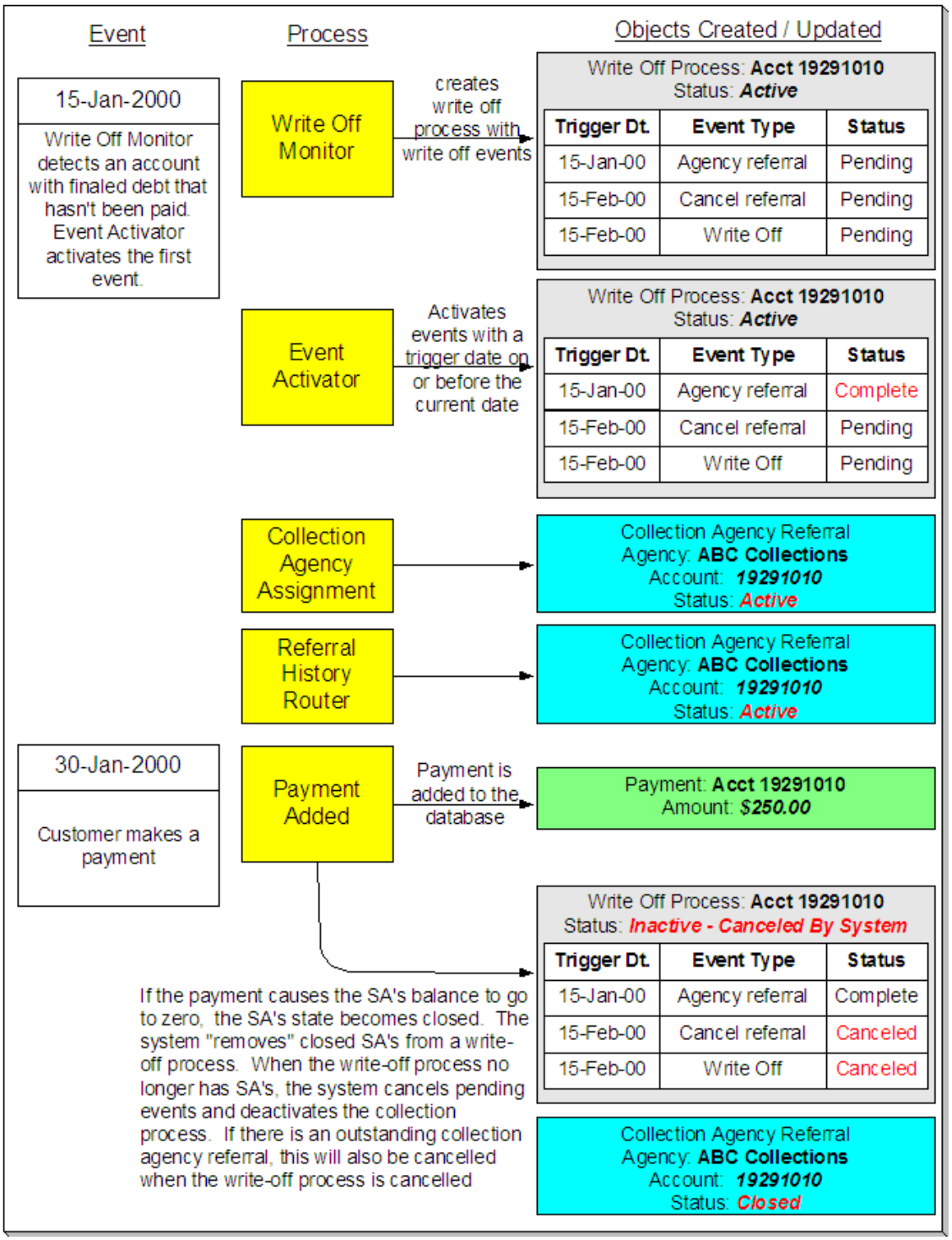
This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

In the above diagram, this process sets the trigger date of the second event on 19-Dec-99.

The Lifecycle Of A Write-Off Process And Its Events

The following diagram illustrates the lifecycle of a typical write-off process and its write-off events:

Copyright © 2026, Oracle and/or its affiliates.



The above diagram is referenced in the following topics.

- What Is A Write-Off Process?
- How Are Write-Off Processes Created?
- How Are Write-Off Events Completed?
- How Are Write-Off Processes (and their Events) Canceled?
- What Happens If The Debt Is Not Paid?

Copyright © 2026, Oracle and/or its affiliates.

- How Is Debt Referred To Collection Agencies

What Is A Write-Off Process?

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its delinquent debt. Linked to the write-off process are the specific service agreements that contributed to the delinquent debt.

How Are Write-Off Processes Created?

A write-off process and its events may be created as follows:

- The Write Off Monitor is a background process (WPM) that creates a write-off process when an account has finaled, unpaid service agreements. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.
- A user may create an ad hoc write-off process at will. The write-off process template controls the number and type of events linked to the write-off process.

In the above diagram, the event on 15-Jan-00 shows the creation of a write-off process with three write-off events. Notice that the events all exist in the Pending state. The events will exist in this state until their trigger date is on or before the current date.

NOT_SUPPORTED

Refer to How Does The Write-Off Monitor Work? to understand how the system monitors overdue debt and initiates the appropriate write-off process when a customer has unpaid, finaled service agreements. Refer to The Big Picture Of Write-off Events for more information about a write-off process' events.

How Are Write-Off Events Completed?

The event activator runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates a pending event on 15-Jan-00. Notice that the events are moved to the Complete state.

NOT_SUPPORTED

Refer to The Big Picture Of Write-off Events for more information about the various types of write-off events and what happens when they are completed.

Copyright © 2026, Oracle and/or its affiliates.

How Are Write-Off Processes (and their Events) Canceled?

When a stopped service agreement's balance is reduced to zero (i.e., it closes), the system checks if there is an ongoing write-off process associated with the service agreement. If so:

- The system "removes" the service agreement from the write-off process.
- When the last service agreement is removed from a write-off process, the system cancels all pending write-off events and deactivates the write-off process.
- If the canceled process has collection agency referrals, the system cancels the referrals.

> **Note**
>
> **Real time cancellation.** Unlike collection processes, the system cancels write-off processes real time (i.e., there is no equivalent of the collection process monitor for write-off processes). This means that if a customer (or collection agency) pays for a service agreement that is linked to write-off process, the system will cancel the process and its collection agency referrals (if any) the moment the payment is entered.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Write Off Process Templates for more information about write off process templates.

Besides the automated cancellation process, a user may cancel a write-off process at will.

What Happens If The Debt Is Not Paid?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the process, the debt will be written off (assuming you designed your write off process template to do such).

> **Note**
>
> **The last write-off event should cause debt to be written off.** Debt will only be written off if the last write-off event is the type that writes off debt. If the write-off process template contains no such event, nothing further will transpire until the next time the write-off monitor runs (when it will create another write-off process for the unpaid, finaled debt).

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to How Is Debt Financially Written-Off? for a description of exactly how debt is written off.

How Is Debt Referred To Collection Agencies

NOT_SUPPORTED

Refer to How Do Collection Agency Referrals Work? for more information about how the system refers debt to a collection agency.

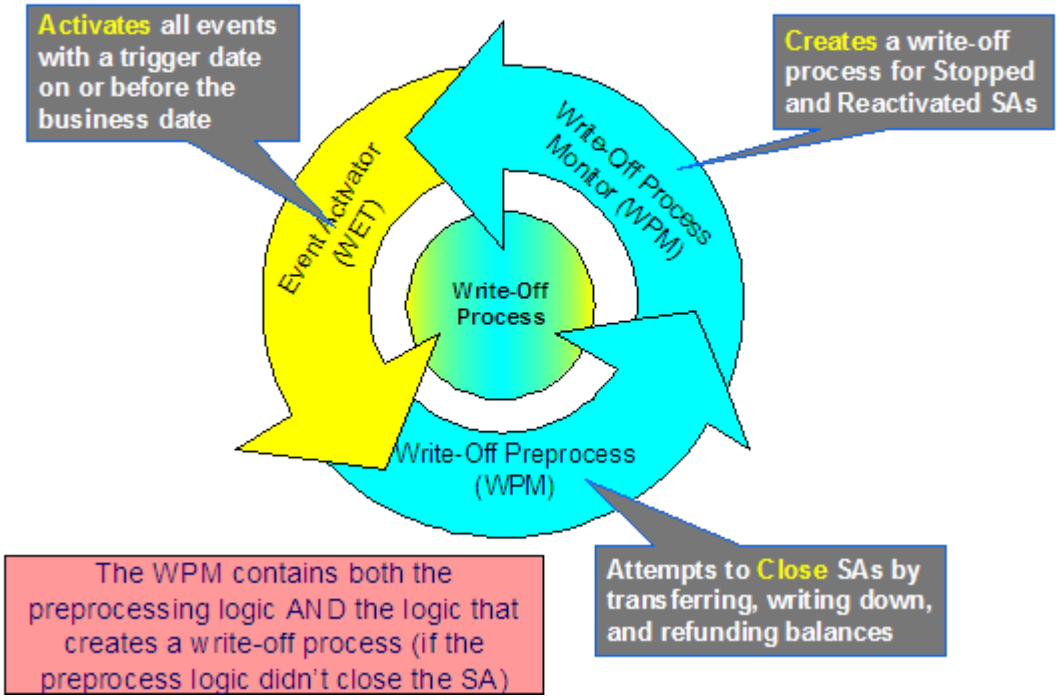
The Write-Off Process Background Processes

There are 2 background processes that automate the creation, activation, and cancellation of write-off processes:

Note

If necessary, you can write-off debt outside of the automated write-off process by simply transferring bad debt to a write-off service agreement using the Write Off transaction.

Copyright © 2026, Oracle and/or its affiliates.



- Write-Off Monitor
- Write-Off Event Activator

Write-Off Monitor

The Write-Off Monitor (WPM) is a background process that has two functions:

- It attempts to close Stopped and Reactivated service agreements using several methods as described under Attempt To Close The SA Before Creating A Write Off Process.
- If it is unsuccessful at closing a SA, it creates a write-off process. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

In the above diagram, the event on 15-Jan-2000 shows the creation of a write-off process with three write-off events. Notice that the events all exist in the Pending state. The events will exist in this state until their trigger date is on or before the current date.

Write-Off Event Activator

The Write-Off Event Activator (WET) runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

Copyright © 2026, Oracle and/or its affiliates.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

In the above diagram, the write-off event activator activates a pending event on 15-Jan-2000. Notice that the event is moved to the Complete state.

Collection Process Maintenance

A collection process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific service agreements that contributed to the delinquent debt. The topics in this section describe the pages on which collection process-related information is maintained.

> **NOT_SUPPORTED**
>
> For more information about collection processes, refer to The Lifecycle Of A Collection Process And Its Events.

- Collection Process - Main Information
- Collection Process - SAs (Service Agreements)
- Collection Process - Events
- How To Perform Common Collection Process Functions

Collection Process - Main Information

The Main page contains core collection process information. Open this page using **Menu**, and then **Credit & Collection**, and then **Collection Process**, and then **Search.**

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

Description of Page

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

Collection Process ID is the unique system-generated identifier for this collection process.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed next to the ID number.

Copyright © 2026, Oracle and/or its affiliates.

Collection Status defines the state of the collection process. The following values may exist:

Active The system creates a collection process in the Active state. A collection process remains in this state until the last event linked to the process is complete OR the customer pays the collection amount on the service agreements linked to the process OR you want to stop the process for whatever reason.

Inactive A collection process becomes Inactive when the last event linked to the process is completed OR the customer pays the collection amount on all service agreements linked to the process OR you want to stop the process for whatever reason. The **Reason** field tells you how a collection process became inactive.

> **NOT_SUPPORTED**
>
> To completely understand the significance of a collection process' status, refer to The Lifecycle Of A Collection Process And Its Events.

Reason defines how a collection process became Inactive or why a collection process is Active. The following values may exist:

Canceled by User This reason should be used when a user manually inactivates a collection process (for whatever reason).

Canceled by System This reason is populated by the system when it cancels a collection process. The system automatically cancels a collection process when sufficient credits have been posted to the process' service agreements since the process was started.

Events Pending This reason is populated by the system when a process has pending events.

Completed This reason is populated by the system when it completes the last event linked to the collection process.

Collection Class Control defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which service agreements can be linked to the collection process (every service agreement is associated with a specific debt class). This field is gray when there are service agreements linked to the collection process.

> **Note**
>
> **Multiple collection processes may be linked to an account.** It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process must be created for each debt class that you want to collect.
>
> **One collection process per debt class.** A given debt class for an account may only have one ongoing collection process at any point in time.

Beneath **Collection Class Control** is displayed a message showing why the system generated the collection process. This information is only displayed for collection processes created by the Account Debt Monitor. If you have plugged in a Collection Process Additional Information algorithm on the Installation record, the results of the algorithm, if any, is displayed on a separate line.

The following line displays the minimum value payment to the collection process debt class that will cause cancellation of the collection process.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to How Does The Account Debt Monitor Work?.

Collection Process Template defines the collection events in the process. You can override these events on the **Events** page.

Note

Changing the template. If you change the template when all events are pending, the system will remove the pending events and replace them with the new template's events.

NOT_SUPPORTED

For more information, refer to Setting Up Collection Process Templates.

Creation Date/Time defines the start date/time of the collection process. This is important as it affects the trigger dates of the process' collection events. This field becomes protected after the collection process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Collection Process Template** to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first collection event in the newly unprotected **Create Date/Time**.
- Change the **Collection Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

The collection process cancellation logic will cancel a collection process when its account's debt no longer exceeds a given amount (you define this threshold amount when you set up the system). Rather than compare an account's total debt to the threshold amount, the cancellation logic compares the account's debt that is older than X days to the threshold amount. X is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ___ Days In Arrears**.

NOT_SUPPORTED

For more information, refer to How Are Collection Processes (and their Events) Canceled?

Enter any **Comments** about the collection process.

The tree at the bottom of the page shows a variety of information about the **account** and the collection process including:

Copyright © 2026, Oracle and/or its affiliates.

- The service agreements linked to the collection process.
- The events linked to the collection process.
- The amount of debt currently associated with the account's debt classes.
- Whether a pay plan exists for the account, or existed at the time the collection process as created.
- The balance for any service agreements linked to the collection process in Inactive status but that have credit balances contributing to the overall debt class arrears for the account.
- A summary of other active collection activities (i.e., collection, severance and write-off processes) associated with the **account**.

Collection Process - SAs (Service Agreements)

The SA page contains the service agreements whose debt is being collected by the collection process. Open this page using **Menu**, and then **Credit & Collection**, and then **Collection Process**, and then **Search** and navigate to the **SAs** page.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

ID is the unique system-generated identifier for this collection process.

The **Collection SAs** scroll contains the service agreement being collected under this collection process. To modify a service agreement linked to the collection process, simply move to a field and change its value. To add a new service agreement, press the + button to insert a row, then fill in the information for each field. The following fields display:

Service Agreement ID identifies the service agreement being collected. Information about the service agreement is displayed beneath the SA ID.

Status defines the state of the service agreement in respect of the collection process. The following values may exist:

Active An Active service agreement is one currently being collected by the collection process. The service agreement will remain in this state until the last event linked to the process is complete OR the customer pays the collection amount on the service agreement OR you want to remove the service agreement from the collection process for whatever reason.

Inactive A service agreement becomes Inactive within the collection process when the last event linked to the process is completed OR the customer pays the collection amount on the

Copyright © 2026, Oracle and/or its affiliates.

service agreement OR you want to remove the service agreement from the collection process for whatever reason

The **Collection Amount** is the amount of debt that initially triggered the collection process. This will be zero for collection processes that were created manually.

The **Amount Still Owing** is the amount of debt older than the **Collection Amount Base Date** defined on the Main page.

The **Credit Amount** is displayed if the Amount Still Owing is less than zero.

The **SA Balances** area displays how much is owed for the service agreement. The **Days Old** and **Arrears Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

> **Note**
>
> Information about debt balances by age is not applicable to open-item accounts.

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the service agreement. This value is only displayed when it differs from the current balance.

> **NOT_SUPPORTED**
>
> Refer to Current Amount versus Payoff Amount for more information.

Collection Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Menu**, and then **Credit & Collection**, and then **Collection Process**, and then **Search** and navigate to the **Events** page.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

> **Note**
>
> We strongly recommend that you understand the information described in How Are Collection Events Completed? before using this page.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

The number and type of events associated with a collection process are defined by the process's collection process template (which is defined on the first page). Refer to Setting Up Collection Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

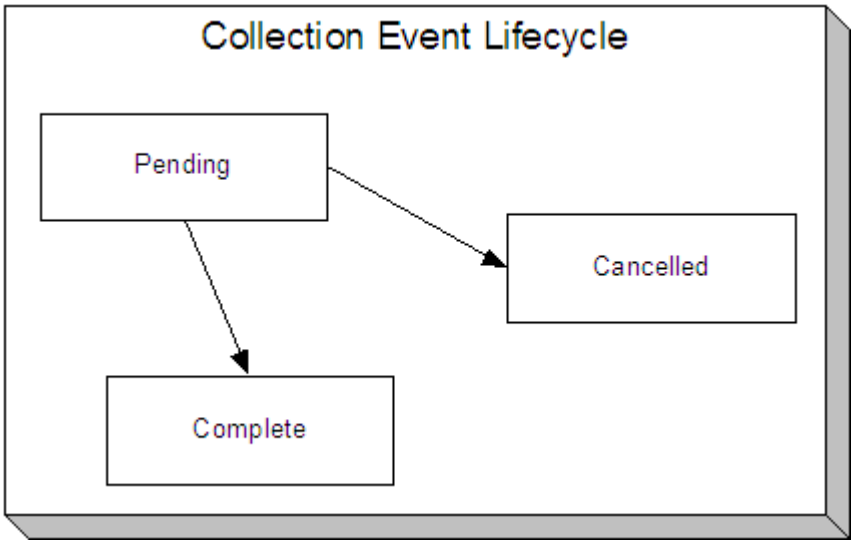
Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

ID is the unique system-generated identifier for this collection process.

The **Collection Events** scroll contains the collection events that comprise the collection process. Only one event is displayed at a time. The inner scroll controls the display of the customer contacts that were created by the system when a particular collection event was triggered.

Event Sequence is the unique identifier of the event.

Collection Event Status defines the state of the event. The following diagram shows the possible lifecycle of a collection event:



Collection events are initially created in the pending state.

When the system sees a pending event with a **Trigger Date** on or before the current date, the system executes the event's activity and completes the event.

You can cancel an individual collection event by setting the status to cancelled.

The **Collection Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to Setting Up Collection Process Templates.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

NOT_SUPPORTED

For more information about a collection event's trigger date, see Collection Event Trigger Date.

If the status of the event is complete, the **Completion Date** is displayed with the date on which the system completes the event. If the status of the event is cancelled, the **Canceled Date** is displayed with the date on which the event was system canceled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Collection Process Functions

The topics in this section describe how to perform common collection process maintenance functions. Refer to The Lifecycle Of A Collection Process for more information about collection processes. Refer to The Big Picture Of Collection Processing for high-level information about collection processing.

- How To Create A Collection Process
- How To Change Collection Events
- How To Cancel A Collection Process

How To Create A Collection Process

99.9% of all collection processes are created by the Account Debt Monitor and require no human intervention before they are executed (refer to How Are Collection Processes Created for information about how the account debt monitor creates collection processes). The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new collection process.
- After the account is populated on Control Central, choose the **Collection Process**, and then **Add** option on the account context menu to transfer to the collection process transaction in add mode.
- After the Collection Process - Main page appears, specify the appropriate **Collection Class Control**. **Collection Class Control** defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which service agreements can be linked to the collection process (every service agreement is associated with a specific debt class). It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process must be created for each

Copyright © 2026, Oracle and/or its affiliates.

debt class that you want to collect. Also be aware that a given debt class for an account may only have one ongoing collection process at any point in time.

- Choose a **Collection Process Template**. **Collection Process Template** defines the collection events in the process. You can override these events on the Events page. For more information, refer to Setting Up Collection Process Templates.
- Use **Collection Amount Base Date** to define the arrears cutoff date that the system uses when it "removes" a SA from a collection process. The system removes a SA from a collection process when its debt older than X is less than or equal to zero. X is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ___ Days In Arrears**.
- Navigate to the **SAs** page to define the specific service agreement's whose debt is covered by this collection process. You must define at least one service agreement.
- Save the collection process.

How To Change Collection Events

When a collection process is first created, it has one or more collection events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when a collection process is initiated are defined on the collection process's collection process template. The following points describe how to add / change / delete events on a collection process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the collection process whose events need to be changed.
  After the account is populated on Control Central, choose the **Collection Process**, and then **Search** option on the account context menu to transfer to the collection process transaction in update mode.

  To add a new event, transfer to the **Events** tab and press the + button in the **Collection Events** scroll to add a new event. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
- Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
- Choose a **Collection Event Status** of Pending.
- Choose the desired **Collection Event Type**.
- Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
  To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, press the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.

  To change an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, make the desired changes.

  After all desired changes have been made, save the collection process.

Copyright © 2026, Oracle and/or its affiliates.

How To Cancel A Collection Process

The Collection Process Monitor will automatically "remove" service agreements from a collection process when their debt has been reduced sufficiently. When all service agreements are "removed" from a collection process, the collection process is canceled. The following points describe how to manually cancel a collection process.

- Use **Control Central** to choose the account with the collection process to be cancelled.
- After the account is populated on Control Central, choose the **Collection Process**, and then **Search** option on the account context menu to transfer to the collection process transaction in update mode. A list of all collection processes associated with the account appears. If only one collection process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The collection process and its pending events are canceled.

Severance Process Maintenance

A severance process is a series of events (e.g., field orders, letters, To Do entries) meant to encourage an account to pay its delinquent debt. Linked to the severance process is the specific service agreement being severed. The topics in this section describe the pages on which severance process-related information is maintained.

NOT_SUPPORTED

For more information about severance processes, refer to The Lifecycle Of A Severance Process And Its Events.

- Severance Process - Main Information
- Severance Process - Severance Events
- Severance Process - Field Activity/Customer Contact
- How To Perform Common Severance Process Functions

Severance Process - Main Information

The Main page contains core severance process information. Open this page using **Menu**, and then **Credit & Collection**, and then **Severance Process**, and then **Search.**

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Severance Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the division and SA type, and severance process template code, the process status, the reason code for the process status and the creation date of the process.

Severance Process ID is the unique system-generated identifier for this severance process.

The remaining fields are maintained on the main severance process page.

Service Agreement ID is the system-generated identifier of the service agreement to which the severance process is linked. Adjacent to the ID number is summary information about the SA.

Severance Status defines the state of the severance process. The following values may exist:

Active The system creates a severance process in the Active state. A severance process remains in this state until the last event linked to the process is complete OR the customer pays the collection amount associated with the severance process OR you want to stop the process for whatever reason. The **Reason** field tells you why a severance process is still in the active state.

Inactive A severance process becomes Inactive when the last event linked to the process is completed OR the customer pays the collection amount on the severance process OR you want to stop the process for whatever reason. The **Reason** field tells you how a severance process became inactive.

NOT_SUPPORTED

To completely understand the significance of a severance process' status, refer to The Lifecycle Of A Severance Process And Its Events.

Reason defines how a severance process became Inactive or why a severance process is Active. The following values may exist:

Canceled by User This reason should be used when a user manually inactivates a severance process (for whatever reason).

Canceled by System This reason is specified by the system when it cancels a severance process. The system automatically cancels a severance process when sufficient credits have been posted to the process' service agreement since the process was started.

Events Pending This reason is specified by the system when a process has pending events.

Completed This reason is specified by the system when it completes the last event linked to the severance process.

Copyright © 2026, Oracle and/or its affiliates.

Severance Process Template defines the severance events in the process. You can override these events on the **Severance Events** page. If you change the template after events exist, the system will remove the existing events and replace them with the new template's events.

> **NOT_SUPPORTED**
>
> For more information, refer to Setting Up Severance Process Templates.

Create Date/Time defines the start date/time of the severance process. This is important as it affects the trigger dates of the process' severance events. This field becomes protected after the severance process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Severance Process Template** to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first severance event in the newly unprotected **Create Date/Time**.
- Change the **Severance Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

The **Collection Amount** is the amount of debt that initially triggered the severance process.

The **Amount Still Owing** is the amount of debt older than the **Severance Amount Base Date**.

The severance process cancellation logic will cancel a severance process when its SA's debt no longer exceeds a given amount (you define this threshold amount when you set up the system). Rather than compare a SA's total debt to the threshold amount, the cancellation logic compares the SA's debt that is older than X days to the threshold amount. X is the date specified in **Severance Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using __ Days In Arrears**.

> **NOT_SUPPORTED**
>
> For more information, refer to How Are Severance Processes (and their Events) Canceled?

Enter any **Comments** about the severance process.

The tree at the bottom of the page shows a variety of information about the **service agreement's** account and the severance process including:

- The events linked to the severance process.
- The amount of debt associated with the account's debt classes.
- A summary of other active severance activities (i.e., severance, severance and write-off processes) associated with the **service agreement's** account.

Copyright © 2026, Oracle and/or its affiliates.

Severance Process - Severance Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Menu**, and then **Credit & Collection**, and then **Severance Process**, and then **Search** and navigate to the **Events** page.

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Severance Process Functions for more instructions describing how to use this transaction.

Note

We strongly recommend that you understand the information described in How Are Severance Events Completed? before using this page.

NOT_SUPPORTED

The number and type of events associated with a severance process are defined by the process's severance process template (which is defined on the first page). Refer to Setting Up Severance Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the SA business unit and SA type, set ID and severance process template name, the process status, the reason code for the process status and the creation date of the process.

ID is the unique system-generated identifier for this severance process.

The **Severance Events** scroll contains the severance events that comprise the severance process. The grid at the bottom (when visible) shows the events that must be completed before the severance event is triggered. Only one severance event is displayed at a time.

Event Sequence is the unique identifier of the event.

Severance Event Status defines the state of the event.

NOT_SUPPORTED

For more information, refer to Severance Event Lifecycle.

Copyright © 2026, Oracle and/or its affiliates.

The **Severance Event Type** defines the event's activity (e.g., a field order is generated, a To Do entry is generated, a letter is sent).

> **NOT_SUPPORTED**
>
> For more information, refer to Setting Up Severance Event Templates.

If **Dependent on Other Events** is turned on, then the event's trigger date is only determined after the event(s) on which it depends are completed. When this switch is on, a grid appears at the bottom of the page in which you define the event(s) on which this event depends.

> **NOT_SUPPORTED**
>
> For more information about dependent events, see Severance Event Dependencies & Trigger Date.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event. This field is gray when the event is dependent on the completion of other events.

> **NOT_SUPPORTED**
>
> For more information about a severance event's trigger date, see Severance Event Dependencies & Trigger Date.

Days After Prev Response defines how a dependent event's trigger date is calculated. The trigger date is set equal to X days after the last dependent event is completed, where X is the number in this field. This field is gray when the event is not dependent on other events.

> **NOT_SUPPORTED**
>
> For more information about dependent events, see Severance Event Dependencies & Trigger Date.

If the status of the event is complete, the **Completion Date** is displayed with the date on which the system completes the event. If the status of the event is cancelled, the **Canceled Date** is displayed with the date on which the event was canceled.

The grid at the bottom of the page contains the events on which this event depends. This grid is only visible when the event is dependent on other events (as defined by the Dependent on Other Events switch). To add additional sequences press the + button (insert row), and define the sequence of the event on which this event depends.

Severance Process - Field Activity/Customer Contact

Copyright © 2026, Oracle and/or its affiliates.

The Field Activity/Customer Contact page contains information about the objects created when the system completes the event. Open this page using **Menu**, and then **Credit & Collection**, and then **Severance Process**, and then **Search** and navigate to the **FA/CC** page.

> **NOT_SUPPORTED**
>
> What happens when a severance event is completed (i.e., executed) is dictated by the severance event's template. Refer to Types Of Severance Events for more information.

Description of Page

The top portion of every page in this page contains the following information:

Severance Process is a concatenation of summary information about this severance process. It is composed of the name of the main customer on the account, the SA business unit and SA type, set ID and severance process template name, the process status, the reason code for the process status and the creation date of the process.

ID is the unique system-generated identifier for this severance process.

The **Severance Events** scroll contains the severance events that comprise the severance process. The grids show the objects created when the system completes each event.

The **Field Activity/SP** grid contains information about the field activities that were created when the event was activated. This information will only be populated with the event's event type causes a field activity to be generated.

The **Customer Contact** area contains information about the customer contacts that were created when the event was activated. This information will only be populated with the event's event type causes a customer contact to be created (note, letters are generated via the creation of a customer contact).

How To Perform Common Severance Process Functions

The topics in this section describe how to perform common severance process maintenance functions. Refer to The Lifecycle Of A Severance Process for more information about severance processes. Refer to The Big sPicture Of Severance Processing for high level information about severance processing.

- How To Create A Severance Process
- How To Change Severance Events
- How To Cancel A Severance Process

How To Create A Severance Process

99.9% of all severance processes are created when a "start severance" collection event is activated by the Collection Event Activator and require no human intervention before they are executed (refer to How Are Severance Processes Created for information about how the account debt monitor creates severance processes). The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

Copyright © 2026, Oracle and/or its affiliates.

- Use **Control Central** to choose the account that requires a new severance process.
- After the account is populated on Control Central, choose the **Severance Process**, and then **Add** option on the account context menu to transfer to the severance process transaction in add mode. If the account has multiple service agreements, you must select the service agreement to be severed.
- After the Severance Process - Main page appears, specify a **Severance Process Template**. **Severance Process Template** defines the severance events in the process. You can override these events on the Events page. For more information, refer to Setting Up Severance Process Templates.
- Use **Severance Amount Base Date** to define the arrears cutoff date that the system uses when it determines if a severance process can be cancelled. The system cancels a severance process when its SA's debt that is older than X is less than or equal to zero. X is the date specified in **Severance Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using ___ Days In Arrears**.
- Save the severance process.

How To Change Severance Events

When a severance process is first created, it has one or more severance events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when a severance process is initiated are defined on the severance process's severance process template. The following points describe how to add / change / delete events on a severance process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the severance process whose events need to be changed.
  After the account is populated on Control Central, choose the **Severance Process**, and then **Search** option on the account context menu to transfer to the severance process transaction in update mode.
- To add a new event, transfer to the **Events** tab and press the + button in the **Severance Events** scroll to add a new event. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
- Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
- Choose a **Severance Event Status** of Pending.
- Choose the desired **Severance Event Type**.
- If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then a) specify the sequences of the dependent events in **Event Dependencies** and b) specify the number of days after the completion of the last dependent event when the new event should be activated.
- If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
  To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired severance event. When the desired severance event appears, press the -

Copyright © 2026, Oracle and/or its affiliates.

button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.

To change an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired severance event. When the desired severance event appears, make the desired changes.

After all desired changes have been made, save the severance process.

How To Cancel A Severance Process

The system will automatically cancel a severance process when its service agreement's debt has been reduced sufficiently. Refer to How Are Severance Processes and Events Cancelled for more information.

The following points describe how to manually cancel a severance process.

- Use **Control Central** to choose the account with the severance process to be cancelled.
- After the account is populated on Control Central, choose the **Severance Process**, and then **Search** option on the account context menu to transfer to the severance process transaction in update mode. A list of all severance processes associated with the account appears. If only one severance process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The severance process and its events are canceled.

Write-Off Process Maintenance

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its unpaid, finaled debt. Linked to the write-off process are the specific service agreements that contributed to the delinquent debt. The topics in this section describe the pages on which write-off process-related information is maintained.

NOT_SUPPORTED

For more information about write-off processes, refer to The Lifecycle Of A Write Off Process And Its Events.

- Write Off Process - Main Information
- Write Off Process - SAs (Service Agreements)
- Write Off Process - Events
- How To Perform Common Write-Off Process Functions

Write Off Process - Main Information

Copyright © 2026, Oracle and/or its affiliates.

The Main page contains core write-off process information. Open this page using **Menu**, and then **Credit & Collection**, and then **Write Off Process**, and then **Search.**

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

Write Off Process ID is the unique system-generated identifier for this write-off process.

The remaining fields are maintained on the main write-off process page.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed adjacent.

Write-off Status defines the state of the write-off process. The following values may exist:

Active The system creates a write-off process in the Active state. A write-off process remains in this state until the last event linked to the process is complete OR the finaled debt is paid off OR you want to stop the process for whatever reason.

Inactive A write-off process becomes Inactive when the last event linked to the process is completed OR the finaled debt is paid off OR you want to stop the process for whatever reason. The **Reason** field tells you how a write-off process became inactive.

> **NOT_SUPPORTED**
>
> To completely understand the significance of a write-off process' status, refer to The Lifecycle Of A Write Off Process And Its Events.

Reason defines how a write-off process became Inactive or why a write-off process is Active. The following values may exist:

Canceled by User This reason should be used when a user manually inactivates a write-off process (for whatever reason).

Canceled by System This reason is populated by the system when it cancels a write-off process. The system automatically cancels a write-off process when all finaled debt associated with the process' service agreements is paid.

Events Pending This reason is populated by the system when a process has pending events.

Completed This reason is populated by the system when it completes the last event linked to the write-off process.

Copyright © 2026, Oracle and/or its affiliates.

Write Off Control defines the write-off debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which service agreements can be linked to the write-off process (every service agreement is associated with a specific write-off debt class). This field is gray when there are service agreements linked to the write-off process.

> **Note**
>
> **Multiple write-off processes may be linked to an account.** It's important to be aware that if an account's service agreements reference multiple write-off debt classes, a write-off process is created for each write-off debt class that you want to write-off.
>
> **Multiple write-off processes per write-off debt class.** A given write-off debt class for an account may have multiple ongoing write-off processes at any point in time. However, a given service agreement may only be on one active write-off process at any point in time.

Write Off Process Template defines the write-off events in the process. You can override these events on the **Events** page.

> **Note**
>
> **Changing the template.** If you change the template when all events are pending, the system will remove the pending events and replace them with the new template's events.

> **NOT_SUPPORTED**
>
> For more information, refer to Setting Up Write Off Process Templates.

Create Date/Time defines the start date/time of the write-off process. This is important as it affects the trigger dates of the process' write-off events. This field becomes protected after the write-off process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the Write-off Process Template to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first write-off event in the newly unprotected **Create Date/Time**.
- Change the **Write-off Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

Enter any **Comments** about the write-off process.

The tree at the bottom of the page shows a variety of information about the **account** and the write-off process including:

- The service agreements linked to the write-off process.
- The events linked to the write-off process.

Copyright © 2026, Oracle and/or its affiliates.

- The amount of debt associated with the account's debt classes.
- A summary of other active collection activities (i.e., collection, severance and write-off processes) associated with the **account**.

Write Off Process - SAs (Service Agreements)

The SA page contains the service agreements whose unpaid, finaled debt is being managed by the write-off process. Open this page using **Menu**, and then **Credit & Collections**, and then **Write Off Process**, and then **Search** and navigate to the **SAs** page.

> **NOT_SUPPORTED**
>
> The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off control name, and the creation date of the process.

ID is the unique system-generated identifier for this write-off process.

To modify a service agreement linked to the write-off process, simply move to a field and change its value. To add a new service agreement, press + to insert a row, then fill in the information for each field. The following fields display:

Service Agreement identifies the service agreement being collected. Information about the service agreement is displayed adjacent.

Write Off SA Status defines the state of the service agreement in respect of the write-off process. The following values may exist:

Active An Active service agreement is one currently being collected by the write-off process. The service agreement will remain in this state until the last event linked to the process is complete OR the service agreement is closed (due to payment in full) OR you want to remove the service agreement from the write-off process for whatever reason.

Inactive A service agreement becomes Inactive when the last event linked to the process is completed OR the service agreement is closed (due to payment in full) OR you want to remove the service agreement from the write-off process for whatever reason

The **Write Off Amount** is the amount of debt that initially triggered the write-off process.

The **SA Balances** area displays how much is owed for the service agreement. The **Days Old** and **Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the service agreement. This value is only displayed when it differs from the current balance.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Current Amount versus Payoff Amount for more information.

Write Off Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Menu**, and then **Credit & Collection**, and then **Write Off Process**, and then **Search** and navigate to the **Events** page.

NOT_SUPPORTED

The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Note

We strongly recommend that you understand the information described in How Are Write-off Events Completed? before using this page.

NOT_SUPPORTED

The number and type of events associated with a write-off process are defined by the process's write-off process template (which is defined on the first page). Refer to Setting Up Write Off Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

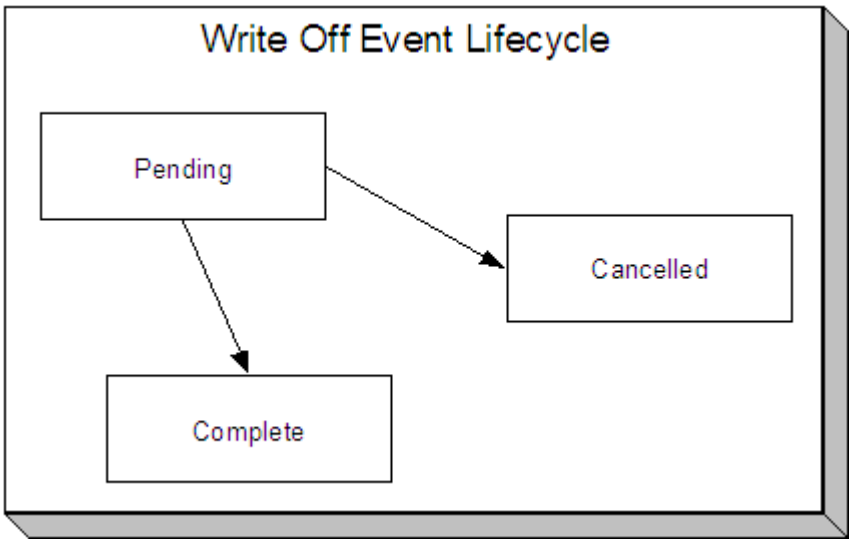
ID is the unique system-generated identifier for this write-off process.

The **Write Off Events** scroll contains individual write-off events that comprise the write-off process. Only one event is displayed at a time. The grid contains the customer contacts that were created by the system when a particular write-off event was triggered (if any).

Event Sequence is the unique identifier of the event.

Write Off Event Status defines the state of the event. The following diagram shows the possible lifecycle of a write-off event:

Copyright © 2026, Oracle and/or its affiliates.



Write-off events are initially created in the pending state.

When the system sees a pending event with a **Trigger Date** on or before the current date, the system executes the event's activity and completes the event.

You can cancel an individual write off event by setting the status to cancelled.

The **Write Off Event Type** defines the event's activity (e.g., the related debt is referred to a collection agency, a To Do entry is generated, a letter is sent).

NOT_SUPPORTED

For more information refer to Setting Up Write Off Process Templates.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

NOT_SUPPORTED

For more information about a write-off event's trigger date, see Write-off Event Trigger Date.

If the status of the event is complete, the **Completion Date** is displayed with the date on which the system completed the event. If the status of the event is cancelled, the **Canceled Date** is displayed with the date on which the event was cancelled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Write-Off Process Functions

The topics in this section describe how to perform common write-off process maintenance functions. Refer to The Lifecycle Of A Write-off Process for more information about write-off

Copyright © 2026, Oracle and/or its affiliates.

processes. Refer to The Big Picture Of Write-off Processing for high level information about write-off processing.

- How To Create A Write-Off Process
- How To Change Write Off Events
- How To Cancel A Write-Off Process

How To Create A Write-Off Process

99.9% of all write-off processes are created by the Write-Off Monitor and require no human intervention before they are executed (refer to How Are Write-off Processes Created for information about how the account debt monitor creates write-off processes). The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new write-off process.
- After the account is populated on Control Central, choose the **Write Off Process**, and then **Add** option on the **account context menu** to transfer to the write-off process transaction in add mode.
- After the Write-off Process - Main page appears, specify the appropriate **Write Off Control**. **Write Off Control** defines the debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which service agreements can be linked to the write-off process (every service agreement is associated with a specific debt class). It's important to be aware that if an account's service agreements reference multiple write-off debt classes, a write-off process must be created for each write-off debt class that you want to collect.
- Choose a **Write Off Process Template**. **Write Off Process Template** defines the write-off events in the process. You can override these events on the Events page. For more information, refer to Setting Up Write-off Process Templates.
- Navigate to the **SAs** page to define the specific service agreement's whose debt is covered by this write-off process. You must define at least one service agreement.
- Save the write-off process.

How To Change Write Off Events

When a write-off process is first created, it has one or more write-off events. The events are the activities that will be performed in the course of the write-off process.

The number and type of events that are created when a write-off process is initiated are defined on the write-off process's write-off process template. The following points describe how to add / change / delete events on a write-off process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the write-off process whose events need to be changed.
  After the account is populated on Control Central, choose the **Write Off Process**, and then **Search** option on the account context menu to transfer to the write-off process transaction in update mode.

  To add a new event, transfer to the **Events** tab and press the + button in the **Write Off Events** scroll to add a new event. At this point, the event has not been added to the

Copyright © 2026, Oracle and/or its affiliates.

database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:

- Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
- Choose a **Write Off Event Status** of Pending.
- Choose the desired **Write Off Type Code**.
- Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
  To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, press the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.

  To change an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, make the desired changes.

  After all desired changes have been made, save the write-off process.

How To Cancel A Write-Off Process

The system will automatically "remove" service agreements from a write-off process when the close (i.e., when their balance becomes zero). When all service agreements are "removed" from a write-off process, the write-off process is cancelled. Refer to How Are Write off Processes and Events Cancelled for more information.

The following points describe how to manually cancel a write-off process.

- Use **Control Central** to choose the account with the write-off process to be cancelled.
- After the account is populated on Control Central, choose the **Write Off Process**, and then **Search** option on the **account** context menu to transfer to the write-off process transaction in update mode. A list of all write-off processes associated with the account appears. If only one write-off process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The write off process and its pending events are canceled.

Writing Off Uncollectable Debt

The Write-off pages described in this section describe the transaction used to manually transfer bad debt to a write-off service agreement.

- Write Off - Main
- Write Off - Write Off SAs

Write Off - Main

Copyright © 2026, Oracle and/or its affiliates.

Most write-offs happen behind-the-scenes as part of the automated write-off processing. If you need to perform a partial write-off or cannot wait for the automated process, open **Menu**, and then **Credit & Collection**, and then **Write Off**, and then **Search.**

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Write Off Processing for background information about the write-off process.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account whose debt is being written off.

The **Candidate SAs** scroll contains one row for each of the account's service agreements that is eligible for write-off. The system displays the respective age and amount of each SA's debt. The accumulated total of all such debt appears in **Write Off Amount**. The checkbox adjacent to this field is used to confirm that you want to write-off the adjacent debt.

When you click the **Create** button, the system writes off the selected debt by executing the write-off algorithm that's plugged in on the account's customer class. Refer to The Ramifications of Write Offs in the General Ledger for an explanation of how the base-package algorithm (CCWO-DST) works.

- How To Manually Write-Off Debt

How To Manually Write-Off Debt

To manually write-off debt:

- Select the **Account** in question on Write Off - Main.
- Select the debt to be written off for each **Candidate SA**. You do this by turning on the checkbox adjacent to **Write Off Amount**. You can also vary the amount of the **Write Off Amount**.
- Click the **Create** button.

Write Off - Write Off SAs

The Write Off SAs page shows all write-off service agreements linked to the account along with their respective current and payoff balance. Use **Menu**, and then **Credit & Collection**, and then **Write Off**, and then **Search** and navigate to the **Write Off SAs** page.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account associated with the WO SAs displayed below.

The grid contains an entry for every write-off service agreement linked to the account. The following information is displayed for each write-off service agreement:

- The service agreement's **SA Information** is displayed.

Copyright © 2026, Oracle and/or its affiliates.

- **Current Balance** contains the WO SA's current balance (i.e., the amount written off).
- **Payoff Amount** contains the WO SA's payoff balance (i.e., the amount written off).

> **Note**
>
> **Write off SA.** There is a field on SA type called Special Role. Only those SA types with a role of Write Off are displayed.

Collection Referral

The Collection Referrals page contains information about an account's debt that has been referred to a collection agency.

> **NOT_SUPPORTED**
>
> Refer to How Do Collection Agency Referrals Work? for more information about how the system creates and maintains collection agency referrals as part of the automated write off processing. In theory, you need only access this page if you need to override the automated processing.

Open **Menu**, and then **Credit & Collection**, and then **Collection Agency Referral**, and then **Search** to maintain this information.

Description of Page

The **Agency Referrals** scroll contains one entry for every collection agency referral associated with the account. The following information is defined for each referral.

Use **Collection Agency** to define the agency to which the referral is being sent.

Start Date is the date on which the referral was initially created.

Referral Status defines if the referral is Active or Closed.

Use **Comments** to describe anything unusual about the referral.

The grid contains the history of interactions with the collection agency. Each time an account's debt is referred to a collection agency, the system creates a referral history record.

You can communicate changes about the referral by inserting a new row in the collection. For example,

- If you need to change the referral amount, insert a row and indicate the **Creation Date** and a **Referral History Reason** of Change Referral.
- If you need to cancel the referral, insert a row and indicate a **Referral History Reason** of Referral Cancellation.
- If the customer pays, insert a row and indicate a **Referral History Reason** of Referral Paid.
- If you need to add a new referral, insert a row and indicate a **Referral History Reason** of Initial Referral.

Copyright © 2026, Oracle and/or its affiliates.

Collection agency referral records are interfaced to the respective collection agency using the batch process defined on the collection agency control record (refer to Setting Up Collection Agencies). The **Batch Control** process and the respective **Batch Number** in which the records were interfaced to the collection agency are displayed adjacent.

The **System Generated** switch is on for those collection agency referrals created as part of the automated write-off processing. Refer to How Do Collection Agency Referrals Work? for more information.

- Downloading Collection Agency Referrals

Downloading Collection Agency Referrals

Collection referral records need to be periodically extracted and sent to collection agencies.

C1-CAGYX - Collection Agency Referral Extract

This batch was developed using the plug-in driven extract template. It contains a select record and process record algorithm. The records can be extended to include custom fields that your implementation needs. The process records algorithm type delivered with the system, C1-CAGYX-PR, contains instruction for extending the record.

- Collection Agency Extract Record

Collection Agency Extract Record

The topics in this section describe the information that appears on each collection agency extract record type.

- Collection Agency Referral Record
- End of File Footer Record
- Address Subrecord

Collection Agency Referral Record

This section describes the layout of the record created by C1-CAGYX.

Field	Format	Source/Value/Description
Account ID	A10	Collection Agency Referral's Account
Name	A254	Account's Main Person Primary Name
Person Type	A2	Person or Business Indicator (Main Person)
Person Type Description	A60	Description of Person Type
ID Type	A8	Primary ID Type (Main Person)
ID Value	A100	Primary ID Value (Main Person)
Primary Phone Type	A30	Person Contact Type (Main Person)
Primary Phone Number	A254	Primary Phone Person Contact Number (Main Person)
Primary Phone Number Extension	A6	Primary Phone Person Contact Extension (Main Person)
Email Address	A254	Primary Email Person Contact (Main Person)

Copyright © 2026, Oracle and/or its affiliates.

Mailing Address	A1241	Mailing Address (Main Person). See below for the fields. See Address Subrecord for the fields.
Service Address	A1241	Service Address of Premise linked to SA being written off. One chosen. Stopped SAs given priority. See below for the fields. See Address Subrecord for the fields.
Customer Class	A8	Account's Customer Class
Customer Class Description	A60	Customer Class Description
Currency	A3	Account's Currency
Referral Activity Date	D10	Collection Referral History Create Date
Referral Activity Reason	A4	Collection Referral History Reason
Referral Activity Reason Description	A60	Collection Referral History Reason Description
Referral Activity Amount	N13.2	Collection Referral History Referral Amount
Initial Referral Date	D10	Collection Agency Referral Initial Referral Date
Initial Referral Amount	N13.2	This is the amount on the record marked with a referral reason of initial. If more than one initial record is found the newest one is chosen by default. A parameter on the process record algorithm allows provides the option of choosing the oldest, newest or some of all initial records. Note: additional initial records can be created manually. The system only creates the first one.
Initial Referral Comments	A254	Collection Agency Referral Comments
Connect Date	D10	If the collection agency referral is linked to a write off process, this is the oldest start date for SAs associated with the write off process. If the collection agency referral is not linked to a write off process this is the oldest start date for SAs on the account, giving preference to stopped SAs.
Disconnect Date	D10	If the collection agency referral is linked to a write off process, this is the latest end date for SAs associated with the write off process. If the collection agency referral is not linked to a write off process this is the latest end date for SAs on the account, giving preference to stopped SAs.
Write Off Date	D10	If the collection agency referral is linked to a write off process, this is the creation date for the write off process. If the collection agency referral is not linked to a write off process, the initial referral date is noted here.
Payments Since Last Activity	N13.2	If the collection agency referral is linked to a write off process, this is the sum of pay segment transactions since the last agency referral history record up to the current record. If the collection agency referral is not linked to a write off process, the sum reflects all SAs.

Copyright © 2026, Oracle and/or its affiliates.

Canceled Payments Since Last Activity	N13.2	If the collection agency referral is linked to a write off process, this is the sum of canceled pay segment transactions since the last agency referral history record up to the current record. If the collection agency referral is not linked to a write off process, the sum reflects all SAs.
Adjustments Since Last Activity	N13.2	If the collection agency referral is linked to a write off process, this is the sum of adjustment and canceled adjustment transactions since the last agency referral history record up to the current record. If the collection agency referral is not linked to a write off process, the sum reflects all SAs.
Fin Resp - Name	A254	Primary Name of Financially Responsible Person on Account
Fin Resp - Person Type	A2	Person or Business Indicator (Financially Responsible Person)
Fin Resp - Person Type Description	A60	Description of Person or Business Indicator (Financially Responsible Person)
Fin Resp - ID Type	A8	Person's Primary ID Type (Financially Responsible Person)
Fin Resp - ID Value	A100	Person's Primary ID Value (Financially Responsible Person)
Fin Resp - Primary Phone Type	A30	Primary Phone Person Contact Type (Financially Responsible Person)
Fin Resp - Primary Phone Number	A254	Primary Phone Person Contact Number (Financially Responsible Person)
Fin Resp - Primary Phone Number Extension	A6	Primary Phone Person Contact Extension (Financially Responsible Person)
Fin Resp - Mailing Address	A1241	Mailing Address (Financially Responsible Person). See Address Subrecord for the fields.

End of File Footer Record

Field	Element Name	Source/Value/Description
Batch Code	A8	
Batch Number	N10	
Thread Number	N10	
Extract Date/Time	A26	
Records Processed	N10	

Address Subrecord

Used in mailing address and service address, total length = 1241 bytes

Field Name	Format	Source/Value/Description
COUNTY	A3	
ADDRESS1	A254	

Copyright © 2026, Oracle and/or its affiliates.

ADDRESS2	A254
ADDRESS3	A254
ADDRESS4	A254
CITY	A90
NUM1	A6
NUM2	A4
HOUSE_TYPE	A2
COUNTY	A90
STATE	A6
POSTAL	A12
GEO_CODE	A11
IN_CITY_LIMIT	A1

Payment Arrangements and Pay Plans

Payment arrangements and pay plans are very similar concepts in that they are both used to payoff debt. This section describes each method.

> **Note**
>
> To distinguish between payment arrangements and pay plans, it is helpful to understand some terminology: 1) an agreement with a customer to pay off their debt as part of their future bills (sometimes referred to as "current bill plus") is a **payment arrangement**, 2) an agreement with specifically scheduled payments is called a **pay plan**.

- Setting Up Payment Arrangement Requests
- Setting Up Payment Arrangements
- Setting Up Pay Plans

Setting Up Payment Arrangement Requests

A payment arrangement request is used to manage the step-by-step process of starting a payment arrangement. These steps can include: checking the account's eligibility for a payment arrangement, capturing the terms of the payment arrangement (i.e. amount of the outstanding debt to include in the payment arrangement, down payment amount, installment details, etc.), performing required approvals, monitoring the payment of an applicable down payment, and the actual creation of the payment arrangement service agreement. Refer to The Big Picture Of Payment Arrangement Requests for more information

The topics in this section describe how to set up a payment arrangement request.

- Starting A Payment Arrangement Request
- Maintaining Payment Arrangement Requests

Copyright © 2026, Oracle and/or its affiliates.

- Payment Arrangement Request Process Flow
- Viewing Payment Arrangement Request History

Starting A Payment Arrangement Request

There are multiple ways start a payment arrangement request:

- From Control Central:
  - Select the Account.
  - Navigate to the **Payment Agreements** tab page
  - On the **Payment Arrangement Requests** zone, click on the **New Payment Arrangement** button.
- From the Main Menu: Navigate using **Menu**, and then **Credit & Collection**, and then **Payment Arrangement Request**, and then **Add**.
- From the **Payment Arrangement Request Search** portal, click on the **Add** button on the page header.
- From a page/zone that displays the Account: On the Account context menu, navigate using **Go To Payment Arrangement Request**, and then **Add**.

All navigation options launch the Payment Arrangement Request Process Flow, which will take you through each of the steps needed to set up a payment arrangement request.

Maintaining Payment Arrangement Requests

A payment arrangement request is used to start a payment arrangement. Aside from capturing the agreed upon terms of the payment arrangement (such as the service agreements and debts to include, number of installments and installment amount), it can also manage any applicable prerequisites for starting the payment arrangement, e.g. down payments, approvals, etc.

Refer toThe Big Picture Of Payment Arrangement Requests for an overview of payment arrangement request functionality.

This section describes how to view and maintain payment arrangement requests.

Navigate using **Menu**, and then **Credit & Collection**, and then **Payment Arrangement Request**, and then **Search**. You are brought to a query portal with options for searching for payment arrangement requests.

Once a payment arrangement request record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Payment Arrangement Request** zone shows the details of the payment arrangement request. Refer to the zone's embedded help for more information.

The **Notes** zone allows users to capture additional notes on the payment arrangement request.

Payment Arrangement Request Process Flow

The base product provides a **Payment Arrangement Request** process flow type that includes three steps:

Copyright © 2026, Oracle and/or its affiliates.

- Execution of one or more eligibility criteria, to determine whether or not the account is eligible for a payment arrangement. The results of each eligibility criteria and the overall eligibility result are displayed on the first panel of the process flow
- Setting up the terms of the payment arrangement:
  - The service agreements and debt to include in the payment arrangement
  - If a down payment must be paid prior to starting the payment arrangement, the down payment amount can be a percentage or a fixed-amount portion of the total payment arrangement amount
  - The remaining payment arrangement amount (after down payment, if applicable)
  - The number of installments
  - The installment amount
- Review of the eligibility results and the agreed upon payment arrangement terms before submitting the payment arrangement request

Refer to the **Payment Arrangement Request** process flow panels for more details. Also refer to Understanding Process Flows for information about how process flows work, in general.

Viewing Payment Arrangement Request History

An account's payment arrangement request history can be viewed on the Payment Arrangement Requests zone of the Control Central – Payment Agreements page. Refer to this zone's embedded help for more details.

Setting Up Payment Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount. Refer to The Big Picture Of Payment Arrangements for information about the financial ramifications of payment arrangements and for a description of how the system monitors payment arrangements.

The topics in this section describe how to set up a payment arrangement and how to break a payment arrangement (if a customer doesn't fulfill their obligations).

- Pay Arrangement - Main
- How To Set Up A Payment Arrangement
- How To Add Additional Debt To A Payment Arrangement
- How To Break A Payment Arrangement
- How To Cancel A Payment Arrangement
- Pay Arrangement - History

Pay Arrangement - Main

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This page is used to create the payment arrangement service agreement. If you require eligibility checking and other prerequisites for starting a payment arrangement, set up a Payment Arrangement Request instead.

This page allows you to set up a payment arrangement. When you create a payment arrangement, the system performs the following functions:

- It creates a payment arrangement service agreement (PA SA). It sets the installment amount on this service agreement equal to the installment amount specified on this page.
- It transfers delinquent debt from each delinquent service agreement(s) to the new PA SA.
- As it transfers the debt to the PA SA, it creates synchronizing adjustments to reduce the PA SA's current balance by the amount transferred. Why? Because when delinquent debt is transferred to the PA SA, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to payoff the debt in installments (and therefore they don't currently owe anything on the PA SA).

> **Note**
>
> **Payment Arrangement Algorithm.** The system only performs the above steps if the new PA SA's SA Type does not reference a Payment Arrangement algorithm. If your implementation uses a Payment Arrangement algorithm, then the logic performed when a payment arrangement SA is created will depend on this algorithm.

You could do the above functions by adding a new service agreement (using the service agreement page) and creating transfer adjustments (using the adjustment page). However, this is tedious. Rather, open **Menu**, and then **Credit & Collection**, and then **Payment Arrangement**, and then **Add** to set up a payment arrangement.

Description of Page

SA (service agreement) **Info** and **SA ID** are displayed on every page. These values only appear after the payment arrangement service agreement (PA SA) exists on the database. The **ID** is a system assigned random number that stays with the payment arrangement service agreement for life. The **SA Info** is a concatenation of important details about the payment arrangement service agreement and its account.

The **Current Balance** that appears beneath **SA Info** contains the PA SA's current balance. When you initially set up a PA SA this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **SA Info** contains the PA SA's payoff balance. The payoff balance is the total amount to be paid off over the PA SA's life. It is only non-zero after delinquent funds have been transferred to the PA SA. This balance is only displayed when it differs from the **Current Balance**.

> **NOT_SUPPORTED**
>
> Refer to Current Amount versus Payoff Amount for more information.

Copyright © 2026, Oracle and/or its affiliates.

Account contains the name of the main customer on the account. **Account ID** is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** below **Account** are the account's respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Candidate SAs** scroll contains one row for each of the account's service agreements with aged debt. The system displays the respective age and amount of each SA's debt. All debt that is checked will be transferred to the PA SA when you click **Create**.

> **Warning**
>
> The system automatically indicates that all debt that isn't "new" (i.e., debt that the customer has seen on their bills) should be transferred to the payment arrangement. It does this by checking each parcel of aged debt.

Total is the sum of ALL current balances (not just the checked amounts) from all candidate SAs.

The next two fields are used to populate the arrangement amount (i.e., the amount the customer will be billed on each bill).

- The **Installments** field exists to calculate the **Arrange Amount** (the next field) when a time period has been stipulated. Simply enter the number of installments and tab out. When you tab out of the field, the system sets the **Arrange Amount** equal to **New Payoff Balance** / **Installments**.

> **Note**
>
> **Arrange Amount may not add up.** The **Arrange Amount** is rounded up so that if you multiply the number of installments by the calculated installment amount, the result will be greater than the PA SA's payoff amount. Don't worry - only the actual dollar amount of the PA SA's total debt will be billed by the system. The last installment will be slightly lower than the other installments.

- Use the **Arrange Amount** field to define the payment arrangement's installment amount if a specific dollar amount has been stipulated. For example, if the customer agrees to pay off their delinquent debt by paying an additional $50 on each bill, you'd enter "50" in this field. Note - If you enter a dollar amount in this field and tab out, the system will calculate the number of installments and display the number in the **Installments** field.

New Payoff Balance contains the PA SA's payoff balance. This is the total amount of debt that will be paid off over the lifetime of the PA SA. This value equals the current payoff balance on the PA SA plus all selected debt from the **Candidate SAs** scroll.

Define the **CIS Division** and **SA Type** of the payment arrangement service agreement.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Payment arrangement SA types.** There is a field on SA type called Special Role. Only those SA types with a role of Payment Arrangement may be selected.

Clicking **Create** (or **Change**)causes the system to perform the following:

- It adds / updates a payment arrangement service agreement (PA SA). It sets the installment amount on this service agreement equal to the installment amount specified on the page.
- It transfers delinquent debt from the delinquent service agreement(s) to the PA SA. The adjustment type used to transfer these funds is defined on the PA SA's SA type.
- It reduces the current balance on the PA SA by the amount of debt transferred to the PA SA. Why? Because when delinquent debt is transferred to the PA SA, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to pay off the debt in installments (and therefore they don't currently owe anything). The adjustment type used to transfer these funds is defined on the PA SA's SA type.

> **Note**
>
> **Payment Arrangement Algorithm.** The system only performs the above steps if the new PA SA's SA Type does not reference a Payment Arrangement algorithm. If your implementation uses a Payment Arrangement algorithm, then the logic performed when a payment arrangement SA is created will depend on this algorithm.

You can transfer to the **History** page to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA SA.

Clicking **Break** causes the PA SA to become broken. Please refer to the note below for what transpires when a PA is broken.

> **Note**
>
> **Break logic is in a plug-in.** Please be aware that the logic that is executed when a payment arrangement is broken exists in a plug-in algorithm (plugged-in on the PA SA's SA type). An implementation can use either a Break Payment Arrangement algorithm which the system calls when a payment arrangement is broken, or the Payment Arrangement algorithm which the system calls when a payment arrangement is created, canceled or broken. These two algorithms are mutually exclusive on the SA Type.
>
> **Break Payment Arrangement Algorithm.** The base package algorithm (SAT BREAK PA) expires the PA SA and cancels all related frozen adjustments, returning the debt to the original service agreements. Because this logic is in a plug-in, you can develop alternative logic and plug it in if the base package logic is not satisfactory.

Copyright © 2026, Oracle and/or its affiliates.

How To Set Up A Payment Arrangement

> **Note**
>
> The following steps only apply to setting up payments using the Payment Arrangement – Main page.

To set up a new payment arrangement:

- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkbox adjacent to the **Candidate SAs** amounts).
- Enter the number of **Installments** and the total **Arrange Amount**.
- Define the **CIS Division** and **SA Type** of the new payment arrangement service agreement.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag is set to *Restricted*, the CIS Division is set to the account's CIS Division and cannot be changed

- Click the **Create** button.

How To Add Additional Debt To A Payment Arrangement

To transfer additional debt to an existing payment arrangement:

- Select the **SA ID** of the existing payment arrangement.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkbox adjacent to the **Candidate SAs** amounts).
- Enter the number of **Installments** and the total **Arrange Amount**.
- Click the **Create** button.

> **Note**
>
> **After transfer.** After transferring the funds to the payment arrangement service agreement, confirm the account's payoff and current balances make sense (the payoff balance is the total amount of debt that the account will eventually have to payoff, the current balance is how much they currently owe, i.e., the amount that credit and collections monitors). The account's payoff and current balances are displayed in the page's second section. If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

Copyright © 2026, Oracle and/or its affiliates.

How To Break A Payment Arrangement

You would break a payment arrangement when a customer doesn't make the agreed payments. Typically, your credit and collections processes will break a payment arrangement behind-the-scenes. Refer to The Big Picture Of Payment Arrangements for information about how the system monitors payment arrangements.

If you want to manually break a payment arrangement simply select the **SA ID** of the existing payment arrangement and click the **Break** button.

How To Cancel A Payment Arrangement

You would cancel a payment arrangement if you made a mistake and never intended to set up the payment arrangement in the first place. To cancel a payment arrangement

- Select the **SA ID** of the existing payment arrangement and navigate to the **History** tab.
- Click the **Cancel** button to cancel the transfer adjustments that were used to transfer debt to the payment arrangement from the customer's regular service agreements.
- Next, navigate to the payment arrangement SA's (PA SA) financial history page (using the service agreement context menu). You will see that there are a few non-canceled adjustments (these were used to change the PA SA's current balance after debt was transferred to the PA SA). Cancel these adjustments by drilling down on these adjustments and then click the **Cancel** button on the adjustment maintenance transaction.
- Finally, display the PA SA on the service agreement maintenance transaction and cancel the payment arrangement.

> **Note**
>
> **Payment Arrangement Algorithm.** If the payment arrangement SA's SA Type references a Payment Arrangement algorithm, then the logic performed when the PA SA is canceled will depend on this algorithm.

Pay Arrangement - History

Open **Menu**, and then **Credit & Collection**, and then **Payment Arrangement**, and then **Search** and navigate to the **History** page to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA SA.

Description of Page

SA (service agreement) **Info** and **ID** are displayed on every page in this page. These values only appear after the payment arrangement service agreement (PA SA) exists on the database. The ID is a system-assigned random number that stays with the service agreement for life. The SA Info is a concatenation of important details about the service agreement and its account.

Copyright © 2026, Oracle and/or its affiliates.

Current Balance beneath **SA Info** contains the PA SA's current balance. When you initially set up a PA SA this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **SA Info** contains the PA SA's payoff balance. The payoff balance is the total amount to be paid off over the PA SA's life. It is only non-zero after delinquent funds have been transferred to the PA SA. This balance is only displayed when it differs from the Current Balance.

> **NOT_SUPPORTED**
>
> Refer to Current Amount versus Payoff Amount for more information.

Account contains the name of the main customer on the account. **Account ID** is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** below **Account** are the account's respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Adjustment Details** area contains one row for each transfer adjustment used to transfer delinquent debt to the PA SA. Click the Go To button to transfer to the adjustment page to view the details of an adjustment (or to cancel the adjustment if you want to reinstate the debt on the originating service agreement).

Total Frozen Adj is the total amount of debt transferred to the payment arrangement that has not been canceled.

Clicking **Cancel** causes the system to cancel all related frozen adjustments, returning the debt to the original service agreements. You must specify the cancel reason by selecting it from the drop down list when Payment Arrangement Cancellation window is displayed.

Total Canceled Adj is the total amount of debt transferred to the payment arrangement that has been canceled.

Setting Up Pay Plans

A pay plan (PP) is an agreement with a customer to make specific payments on specific dates (as opposed to payment arrangements where the customer makes fixed payments as part of their regular bills).

Consider the following situations that would require a pay plan:

- A customer makes a request to payoff severely overdue debt in one or more installments. The existence of the pay plan insulates the portion of the debt covered by the plan from C&C processing (specifically, from the account debt monitor).
- A commercial or industrial customer cannot or will not post a deposit. This type of pay plan is not specifically paying off past debt; rather, it can be viewed as an installment plan to prepay a regular service agreement.

The topics in the following section describe how to set up a pay plan.

- Pay Plan - Main

Copyright © 2026, Oracle and/or its affiliates.

- How To Set Up A Pay Plan
- Reviewing Pay Plans

Pay Plan - Main

The pay plan transaction allows you to set up a pay plan. When you create a pay plan, the system performs the following functions:

- It creates an active pay plan using the pay plan type entered.
- It defaults the customer as the payor if no third party is defined.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Pay Plans for information about the financial ramifications of pay plans and for a description of how the system monitors pay plans.

Open **Menu**, and then **Credit & Collection**, and then **Pay Plan**, and then **Search** to maintain a customer's pay plan.

Description of Page

Pay Plan and **Pay Plan ID** only appear after the pay plan is added to the database. **Pay Plan ID** is a system-assigned random number that stays with the pay plan for life. **Pay Plan** is a concatenation of important details about the pay plan.

Account ID identifies the account of the customer for which the pay plan exists. This field is protected once the pay plan is Active.

The **Status** displays the status of the pay plan. Refer to The Lifecycle Of A Pay Plan for more information about a pay plan's status.

Created by displays the id of the user who created the pay plan and the date and time on which it was created.

Clicking the **Cancel** button causes the pay plan's status to become cancelled.

Last Updated by displays the id of the user who last modified the pay plan and the date and time on which it was changed.

Select a Pay Plan **Type**. The Pay Plan Type defines the **Debt Class** whose debt is insulated by the pay plan. The account's **Current Balance** in this debt class appears beneath as well as the portion that is considered **Delinquent Debt**.

If a 3rd party is responsible for the pay plan's payment, **Third Party Payor** must be checked, and the associated 3rd party selected.

The **Payor Account Id** contains the account ID and name of the person responsible for making the pay plan's payments. If a **Third Party Payor** is responsible, the payor's account appears; otherwise the account whose debt is insulated by the pay plan appears.

The **Start Date** defines the first day on which the pay plan is tracked; it defaults to the current date.

Pay Method defines the manner in which the customer intends to fulfill payments under this pay plan.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Significance of payment method.** A pay method can have a grace period associated with it. This period controls when a scheduled payment is considered to be missed. For example, a pay method of "postal" might have a grace period of 3 days. This would give the customer 3 extra days to make the scheduled payments before the pay plan is broken.
>
> In addition to grace period, pay method also control if the system will automatically create an automatic payment on the scheduled payment date. If the pay method so indicates AND the account has been set up for automatic payment, the PPAPAY background process will create Automatic Payments on the scheduled payment date.

The **Scheduled Payments** grid contains the pay plan's scheduled payments. The **Scheduled Date** and **Scheduled Amount** must be entered for each scheduled payment. The **Total Amount** of the scheduled payments is displayed at the bottom.

Free format **Comments** can be entered to describe any special notes about the pay plan.

How To Set Up A Pay Plan

To set up a new pay plan:

- Select the **Account** in question on Pay Plan - Main.
- Select the **Pay Plan Type** for the type.
- Identify a **Third Party Payor** if there is one.
- Select a **Pay Method**.
- Enter one or more **Scheduled Payments**. The **Total Amount** of the scheduled payments should cover the customer's **Delinquent Debt** if you want the pay plan to protect the customer from additional credit and collection activity.
- Click the **Save** button.

Reviewing Pay Plans

In addition to using this maintenance page for reviewing pay plans, Control Central has a pay plan tree that can be used to view pay plan history, including current status. Refer to Pay Plan Tree for more information.

Financial Transactions

In this section, we describe the financial transactions generated as a result of your bills, payments and adjustments.

- The Big Picture Of Financial Transactions
- Financial Transaction

Copyright © 2026, Oracle and/or its affiliates.

- Account Financial History
- Account Bill / Payment History
- SA Financial History
- SA Non-Accrual Accounting Balance
- Balance Control
- Match Event

The Big Picture Of Financial Transactions

The topics in this section provide background information about a variety of financial transaction issues.

> **NOT_SUPPORTED**
>
> We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate the system's financial architecture.

- Bill Segment Financial Transactions
- Payment Segment Financial Transactions
- Adjustment Financial Transactions
- Financial Transactions And Aged Debt
- Current Balance versus Payoff Balance
- The Source Of GL Accounts On Financial Transactions
- Obscure Things That Can Happen
- The Big Picture of Balance Control
- The GL Interface

Bill Segment Financial Transactions

A bill segment has a related financial transaction. The financial transaction contains the financial effects of the bill segment on the service agreement's current and payoff balances and on the general ledger.

If a bill segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a bill correction.

> **NOT_SUPPORTED**
>
> For more information about bill segment financial transactions, refer to Bill Details.

Copyright © 2026, Oracle and/or its affiliates.

Payment Segment Financial Transactions

A payment segment has a related financial transaction. The financial transaction contains the financial effects of the payment segment on the service agreement's current and payoff balances and on the general ledger.

If a payment segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a negative payment.

NOT_SUPPORTED

For more information about payment segment financial transactions, refer to Payment Details.

Adjustment Financial Transactions

An adjustment has a related financial transaction. The financial transaction contains the financial effects of the adjustment on the service agreement's debt and on the general ledger.

If the adjustment is eventually cancelled, another financial transaction will be linked to the adjustment to reverse its financial effect. The cancellation financial transaction appears on the next bill produced for the account as an adjustment.

NOT_SUPPORTED

For more information about adjustment financial transactions, refer to Adjustment Details.

Financial Transactions And Aged Debt

When a financial transaction's related bill segment / payment segment / adjustment is frozen, the financial transaction (FT) is also frozen. When an FT is frozen, its service agreement's debt is impacted. It is important to stress the following in respect of this impact:

- The FT's GL details will be interfaced to the GL when the GL interface next executes.
- If the FT decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the FT appears on a bill.
- If the FT increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the FT. There is a switch on an FT called New Charge that controls the arrears behavior. If this switch is on, the customer's aged debt will not reflect the FT amount until the FT is swept onto a bill. The moment the FT is swept onto the customer's bill, the debt starts aging. If this switch is off, the date on which the FT starts aging must be defined in the Arrears Date field.

Copyright © 2026, Oracle and/or its affiliates.

- The amount a customer owes in total is immediately affected by the FT regardless of whether the FT appears on a bill. This means that the amount of aged debt may not be in sync with the total amount owed. This seems odd, but is useful from a credit and collections perspective. You see, you probably don't want to start aging a FT until the customer has actually seen it.

NOT_SUPPORTED

For more information, refer to Financial Transactions Created Between Bills.

Current Balance versus Payoff Balance

NOT_SUPPORTED

For information about current and payoff balance in general, refer to Current Amount versus Payoff Amount.

The topics in this section describe when payoff amount differs from current amount for the various types of financial transactions.

- Adjustments - Current Balance versus Payoff Balance
- Billing - Current Balance versus Payoff Balance
- Payment - Current Balance versus Payoff Balance

Adjustments - Current Balance versus Payoff Balance

NOT_SUPPORTED

For information about current and payoff balance for adjustments, refer to Adjustments - Current Balance versus Payoff Balance.

Billing - Current Balance versus Payoff Balance

NOT_SUPPORTED

For information about current and payoff balance for bill segments, refer to Billing - Current Balance versus Payoff Balance.

Copyright © 2026, Oracle and/or its affiliates.

Payment - Current Balance versus Payoff Balance

NOT_SUPPORTED

For information about current and payoff balance for payment segments, refer to Payment - Current Balance versus Payoff Balance.

The Source Of GL Accounts On Financial Transactions

NOT_SUPPORTED

For information about the source of the distribution codes used to generate the GL accounts on the financial transactions, refer to The Source Of GL Accounts On Financial Transactions.

Obscure Things That Can Happen

The topics in this section provide information about obscure things that can happen when a financial transaction (FT) is frozen.

- A Stopped Service Agreement May Be Closed
- A Closed Service Agreement May Be Reactivated
- A Write-Off Process May Be Deactivated If The SA Is Closed
- A Reactivated Service Agreement May Be Closed
- A Collection Process May Be Canceled When A Credit FT Is Frozen
- A Severance Process May Be Canceled When A Credit FT Is Frozen
- One Or More Algorithms May Be Executed

A Stopped Service Agreement May Be Closed

If an FT causes a Stopped SA's current and payoff balances to become zero, the system closes the service agreement (i.e., the service agreement's status becomes Closed).

A Closed Service Agreement May Be Reactivated

Copyright © 2026, Oracle and/or its affiliates.

After a service agreement is closed (i.e., after it's stopped and paid-in-full), it's possible for some types of financial transactions to be linked to the service agreement. These financial transactions could cause the current and/or payoff balances to become non-zero. If this happens, the system reactivates the service agreement (i.e., the service agreement's status becomes reactivated).

When a service agreement becomes reactivated, it becomes eligible for review by the write-off process monitor (when it next runs). The write-off process monitor will, in all likelihood, start a write-off process for the reactivated service agreement.

Financial events that can be linked to a closed service agreement are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The cancellation of a payment segment
- The freezing of an adjustment
- The cancellation of an adjustment

A Write-Off Process May Be Deactivated If The SA Is Closed

When a service agreement becomes Closed, it will be removed from any ongoing write-off processes. This removal could cause the write-off process to become Inactive (if this is the last service agreement linked to the write-off process). The inactivation of the write-off process, could, in turn, cause collection agency referrals associated with the write-off process to be cancelled.

A Reactivated Service Agreement May Be Closed

After a service agreement has been reactivated (refer to the previous section for how this happens), a financial transaction (or transactions) may be linked to it that will cause the current and payoff balance to return to zero. If this happens, the system closes the service agreement (i.e., the service agreement's status becomes Closed).

All types of financial transactions can be posted to a Reactivated service agreement.

A Collection Process May Be Canceled When A Credit FT Is Frozen

Periodically, the system determines if an account's debt violates your collection criteria. If so, a collection process is created using the violated criteria's collection process template.

The Collection Process Monitor reviews an active collection process whenever one of its service agreements' debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The freezing of an adjustment that credits a service agreement

At review time, the Collection Process Monitor removes any service agreement from a collection process when the service agreement has no debt older than the arrears date on the

Copyright © 2026, Oracle and/or its affiliates.

collection process. When all service agreements are removed, the collection process is cancelled.

NOT_SUPPORTED

Refer to The C&C Monitors for more information about debt monitoring.

A Severance Process May Be Canceled When A Credit FT Is Frozen

If the customer does not respond to a collection process's prodding, the system creates a severance process for each delinquent service agreement associated with the collection process.

The system reviews an active severance process whenever its service agreements' debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The freezing of an adjustment that credits a service agreement

At review time, the system determines if the service agreement has no debt older than the arrears date on the severance process. If so, the system looks at the severance process' severance process template:

- If the severance process template indicates that the process can be **Used to sever service**, the system cancels all pending severance events and deactivates the severance process. Note: the only severance processes that are not used to sever service are those that request reconnection of a cut service.
- If the canceled process has field activities,
- If the field activities are not linked to a field order and are still in the pending status, the system cancels the field activities.
- Otherwise, the system generates worklist entries to alert an operator that field activities have been dispatched. The operator must determine how to handle these activities because they could already be scheduled for dispatch (or completed!).

Note

Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no equivalent of the collection process monitor for severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The C&C Monitors for more information about debt monitoring.

One Or More Algorithms May Be Executed

If the FT is linked to a service agreement whose SA type has Freeze Algorithms, these algorithms will be executed when the FT is frozen. Because you can code whatever you please in an algorithm, anything could happen. Examples of such algorithm include:

- An algorithm that routes consumption to third party service providers when a bill is frozen (refer to STG SND CONS for more information). This type of algorithm would be plugged in on SA types whose consumption can be interfaced to other service providers.
- A bill message can be added to the account when the system detects that its deposit has been paid in full (refer to DEP PIF MSG for more information). This type of algorithm would be plugged in on deposit SA types.

In addition, FT freeze algorithms can also be defined on an account's customer class. Because you can code whatever you please in an algorithm, anything could happen. An example of such an algorithm is one that cancels match events when an FT is cancelled (refer to CNCL FT MEVT for more information). This type of algorithm would be plugged in on customer classes associated with open item customers.

The Big Picture of Balance Control

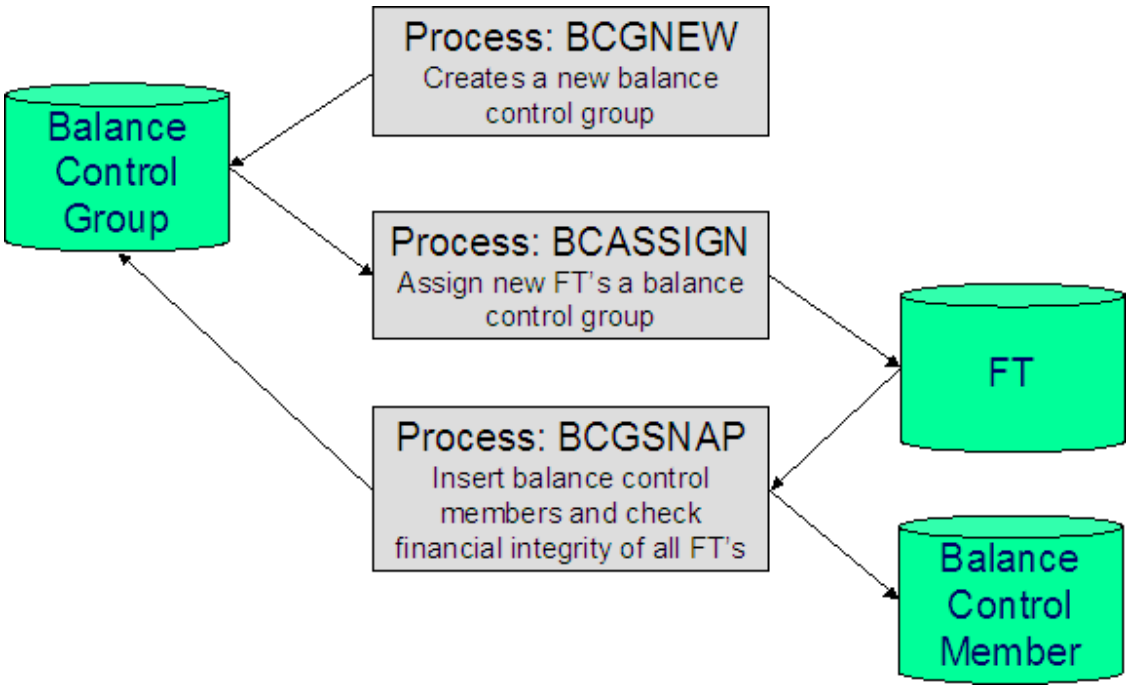
The balance control processes are used to check the financial integrity of your system. The contents of this section describe how these processes work.

- The Balance Control Background Processes
- Balance Control Information Is Available Online

The Balance Control Background Processes

The following diagram illustrates the balance control background processes:

Copyright © 2026, Oracle and/or its affiliates.



- BCGNEW - Create A New Balance Control Group
- BCASSIGN - Assign New Financial Transactions A Balance Control Group
- BCGSNAP - Insert BC Members And Check Financial Integrity Of All FTs
- Consider A Backup At This Point

BCGNEW - Create A New Balance Control Group

This process creates a Pending balance control group if one doesn't already exist. You may wonder why an entire process is dedicated to such a trivial task. The reason is because the next process, BCASSIGN, is a multi-threaded (i.e., parallel) process and we only need one Pending balance control group regardless of the number of threads used to assign balance control ID's to financial transactions.

BCASSIGN - Assign New Financial Transactions A Balance Control Group

This multi-threaded (i.e., parallel) process assigns the Pending balance control group to new FTs whose freeze date/time is before the create date/time of the balance control record.

BCGSNAP - Insert BC Members And Check Financial Integrity Of All FTs

This process performs the following two functions:

- It summarizes new financial transactions under the current Pending balance control group as follows:
- It creates a balance control member for every combination of **Division**, **SA Type** and **FT Type** referenced on the financial transactions belonging to the balance control group.
- It updates each balance control member with the following information:

Copyright © 2026, Oracle and/or its affiliates.

- The number of financial transactions (FTs)
- The sum of the total amounts on the FTs in this balance control group.
- The sum of the current amounts on the FTs in this balance control group.
- The sum of the total amounts from all FTs (in this and all other balance control groups).
- The sum of the current amounts from all FTs (in this and all other balance control groups).
- It sets the status of the balance control group to Complete.
- It checks the integrity of the financial transactions in each historical Balance Control Group. It does this by summarizing EVERY financial transaction throughout time and determining if the sums are in sync with the values maintained on the balance control members. If integrity problems are detected, a detailed error message is displayed on the run control associated with the process.

If you opt to run the balance control processes on a nightly basis, you will find that the verification processing will take longer every night (because there are more financial transactions over time). In order to deal with this issue, the BCGSNAP process has a parameter that allows you to control which of the above functions is implemented (it's call VERIFY-ONLY-SW). In order to speed nightly processing, run this process with the switch set to "G" (this causes new financial transactions to be summarized under a new balance control). Then, once a week (or month), run this process with the switch set to "Y" (this checks the financial integrity of all financial transactions in the system).

If you run the balance control processes less frequently, you can set the VERIFY-ONLY-SW to "N" (this causes both of the above functions to execute).

Consider A Backup At This Point

We recommend you backup your database AFTER you run the above processes. Why? So that if a financial integrity problem is spotted in the future, you can compare the current database against the backup to see what changed.

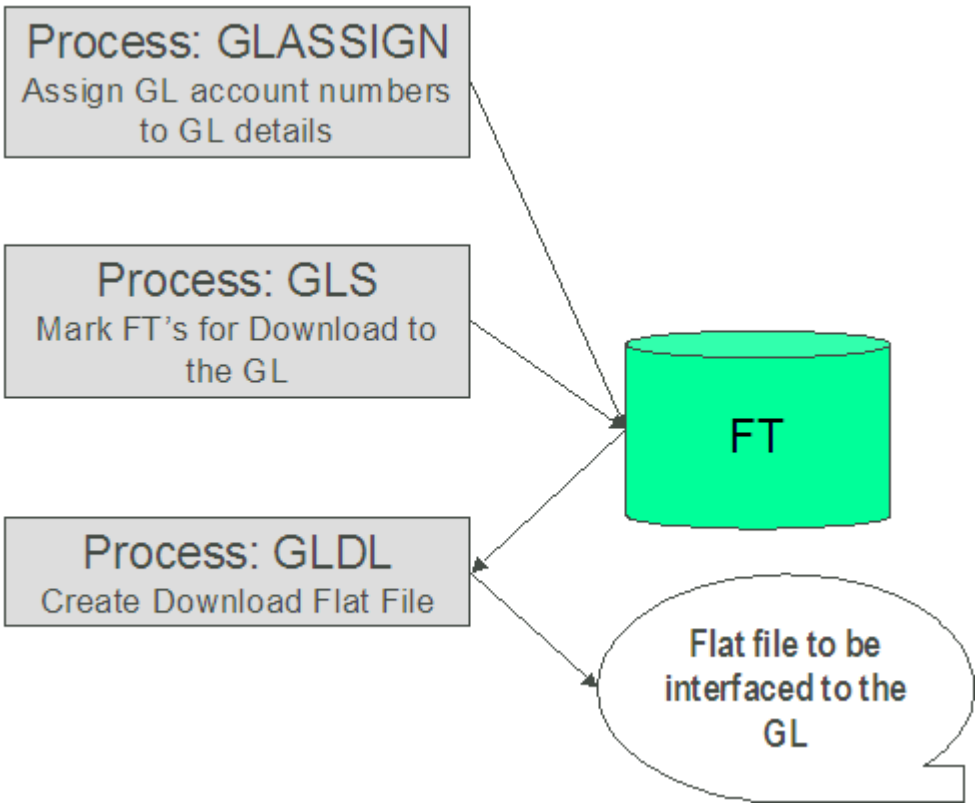
Balance Control Information Is Available Online

Your internal auditors may be interested in the total number and amount of financial transactions that have been posted since a given point in time. You can use the Balance Control page to see this information.

The GL Interface

The following diagram illustrates the GL Interface.

Copyright © 2026, Oracle and/or its affiliates.



- GLASSIGN - Assign GL Account Numbers To GL Details
- GLS - Prepare FTs for Download
- GLDL - Create General Ledger Download Flat File

GLASSIGN - Assign GL Account Numbers To GL Details

The GLASSIGN process assigns GL account numbers to the GL details associated with financial transactions. GL account numbers are assigned as follows:

- Every GL detail references a distribution code.
- Every distribution code references a GL assignment algorithm. The base package algorithm simply uses the default GL account associated with the distribution code. However, you can construct your own algorithm(s) to assemble your GL account numbers in your desired fashion. Refer to Setting Up Distribution Codes for more information about the GL format algorithm.
- The GLASSIGN process simply calls each GL's details distribution code's GL assignment algorithm and updates the GL detail with the result (i.e., the GL account number). This GL account number is then used by the GLDL process when it creates the consolidated journal entry that's interfaced to the GL.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If incorrect GL account numbers get assigned to GL details...** If you do not plug in the correct algorithm or your algorithm is wrong, you can correct the GL account numbers that are assigned to the GL details. How? Write a simple program that resets the GL account numbers on the erroneous GL details. Then run GLASSIGN. GLASSIGN will re-execute the distribution code GL assignment algorithm and refresh the account number accordingly.

GLS - Prepare FTs for Download

The GLS process creates FT download staging records for all FTs that are ready to be posted to the GL (the FT download staging records are stored on the FT/Process table). Each FT download staging record is marked with a batch control ID and run number.

- The batch control ID is the process responsible for creating the flat file that contains the consolidated journal entry that is interfaced to your general ledger. The batch control ID is defined on Installation Options - Financial. The system comes with a single GL download program (GLDL - Create General Ledger Download Flat File). The GL account numbers are provided by the GL account algorithms that are specified on the distribution codes. Refer to Setting Up Distribution Codes for more information about the GL account algorithm.
- The run number is the batch control ID's current run number.

This process also changes the status of the FT to distributed. You may not change the FT's GL details after this time.

GLDL - Create General Ledger Download Flat File

The GLDL process creates the flat file that contains the consolidated journal entry that is interfaced to your general ledger. One header and multiple detail records are created as described below. At the conclusion of the batch process, a validation is performed to compare debits against credits. If they are not equal, the process terminates with an error.

- Header Record Layout
- Detail Record Layout for GLDL

Header Record Layout

Field Name	Format	Source/Value/Description
REC_TYPE	A1	1 (1 is used for header records)
BATCH_CD	A8	The code for the batch process that created the file. (This value will always be 'GLDL'.)

Copyright © 2026, Oracle and/or its affiliates.

BATCH_NBR	N10	The batch process will be run many times. This field indicates which of those runs produced a particular output file. This may be thought of as an instance identifier for the batch process.
BATCH_RERUN_NBR	N10	Any given instance of the GLDL batch process may be re-run at any time. If this instance has been re-run, this field will be populated with a number indicating how often.
EXTRACT_DTTM	A26	The date and time of the run that created a particular output file.
DETAIL_REC_CNT	N12	The number of details records
DETAIL_REC_TOTAL_DR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records that were greater than zero (debits).
DETAIL_REC_TOTAL_CR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records that were less than zero (credits).

Detail Record Layout for GLDL

> **Note**
>
> A record is created for each unique occurrence of REC_TYPE, GL_DIVISION, CURRENCY_CD, GL_ACCOUNT and ACCT_PERIOD. If a given GL account has debit and credit FINANCIAL_AMOUNTs, two records will be created - one shows the debit amount, the other shows the credit amount.

Field Name	Format	Source/Value/Description
REC_TYPE	A1	2 (2 is used for detail records)
GL_DIVISION	A5	This is the GL Division from the CI_FT record. It determines a) which Accounting Calendar will be used to calculate the Accounting Period (below) and b) the default currency (i.e., if currency code is not sourced directly from the FT).

Copyright © 2026, Oracle and/or its affiliates.

CURRENCY_CD	A3	The currency in which the Amount is denominated. **Note** The Use FT's Currency Code (Y/N) batch parameter controls whether the Currency Code is retrieved directly from the FT or from the GL Division (default).
GL_ACCOUNT	A254	Contains the GL account number as supplied by the distribution code's GL account algorithm.
ACCT_PERIOD	A6	An accounting period in the format YYYYPP where YYYY is the fiscal year and PP is the accounting period. This is derived from the GL detail's FT's accounting date and GL division.

Copyright © 2026, Oracle and/or its affiliates.

FINANCIAL_AMOUNT	N13.2	This is the debit or credit amount. Debits are represented by positive numbers; credits are represented by negative numbers. If a given GL account has debit and credit entries, two records will be created - one shows the debit amount, the other shows the credit amount.
STAT_CODE	A8	This is the GL distribution code's statistic code (if any)
STAT_AMOUNT	N13.6	This is the statistical amount from the GL detail lines

> **Note**
>
> The GLASSIGN process updates an FT's GL details with the appropriate GL account number.

Financial Transaction

Payment segments, adjustments and bill segment have a corresponding financial transaction. The financial transaction is created by the system when any of the source transactions is created. It contains the financial effects of its corresponding adjustment / bill segment / payment segment.

> **NOT_SUPPORTED**
>
> **For background information**, refer to The Big Picture Of Financial Transactions.

The topics in this section describe the financial transaction page.

- Financial Transaction - Main
- Financial Transaction - FT Process

Financial Transaction - Main

Use **Menu**, and then **Financial**, and then **Financial Transaction** to update and maintain financial transaction information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Navigating from source transactions.** On the bill, payment, and adjustment maintenance pages, you can click the financial transaction go to button to transfer to this page.

Description of Page

Most of the attributes on this page are display-only. The following points describe the conditions under which certain fields may be modified:

- **Accounting Date** may be modified until the financial transaction (FT) is interfaced to the GL (i.e., until the **GL Distribution Status** is Distributed).
- **New Charge** may be modified until the FT is swept onto a bill (when the next bill is completed for the FT's account). If **New Charge** is modifiable and turned off, **Arrears Date** must be specified.
- **Show on Bill** and **Correction** can be modified until the FT is frozen.

The remainder of this section defines each of the fields on the page.

FT Type is the type of financial transaction. Values are: Adjustment, Adjustment Cancellation, Bill Segment, Bill Segment Cancellation, Pay Segment, and Pay Segment Cancellation. Click the go to button to view the originating transaction on the adjustment, bill, or payment page.

FT ID is the system-assigned, unique identifier of the financial transaction.

SA ID is the system-assigned, unique identifier of the service agreement to which the financial transaction is linked.

Bill ID is the system-assigned, unique identifier of the bill to which the financial transaction is linked. This field is only populated after the FT is swept onto a bill (and this happens when the next bill is completed for the FT's account).

Sibling ID is the system-assigned, unique identifier of the source transaction associated with the FT.

- If this FT is associated with a bill segment, **Sibling ID** contains the unique identifier of the bill segment.
- If this FT is associated with a payment segment, **Sibling ID** contains the unique identifier of the payment segment.
- If this FT is associated with an adjustment, **Sibling ID** contains the unique identifier of the adjustment.

Parent ID contains the following:

- If this FT is associated with a bill segment, **Parent ID** contains the unique identifier of the bill with which the bill segment is associated.
- If this FT is associated with a payment segment, **Parent ID** contains the unique identifier of the payment with which the payment segment is associated.
- If this FT is associated with an adjustment, **Parent ID** contains the adjustment type.

Create Date/Time are the date and time the FT was created.

Accounting Date is the accounting date that will be used by the general ledger to define the accounting period(s) into which the FT will be booked.

Copyright © 2026, Oracle and/or its affiliates.

CIS Division is the CIS division associated with the FT. This comes from the CIS division linked to the FT's service agreement's SA type.

GL Division is the GL division associated with the FT. This comes from the GL division linked to the FT's service agreement's SA type.

Show on Bill indicates if the FT will be shown on the customer's bill. You should only turn this off for erroneous FTs that should not be shown to the customer. For example, if you cancel / rebill a bill segment and you want to suppress the resultant FTs on the printed bill, turn this switch off.

New Charge indicates if the FT's charge only starts aging when the FT is swept onto the next bill produced for the account. If you want to manually enter an **Arrears Date** you must turn this switch off. The system leaves this switch set when assigning an arrears date during the bill completion process. This is done so the system knows which FTs were assigned an arrears date during bill completion. If a completed bill is reopened, the arrears date is reassigned when the bill is completed.

Not In Arrears indicates if the FT's financial impact has been canceled and therefore should not be considered for arrears purposes. This switch is turned on by the system when a FT's source transaction in canceled (both the original FT and the cancellation FT are marked as **Not In Arrears**).

Match Event ID is the system-assigned, unique identifier of the match event to which the financial transaction is linked. This field is only enabled for Open Item Accounts. Be aware that changing a financial transaction's match event can result in the balancing / unbalancing of the prior and newly referenced match events.

Arrears Date is the date the FT starts aging. This field is typically blank until the FT is swept onto the account's bill. If you want to start aging an FT on an historical date (for whatever reason), enter the appropriate **Arrears Date**. (Note - you must uncheck the "New Charge" box first in order to do this.) If the arrears date is assigned manually it is not updated during bill completion nor is it adjusted when a bill is reopened and completed again.

Correction causes the FT to be summarized in the correction area of the bill-at-a-glance. This is set by the system automatically when a bill segment is cancelled / rebilled after its parent bill is completed (i.e., sent to the customer).

Redundant indicates if the FT's financial impact is considered irrelevant. This only happens after an FT has reached an age that is no longer relevant for aging purposes (e.g., when the FT is older than 120 days - or whatever age has been set as the Oldest Bucket Age on the Installation Options) and when its balance is exactly equal to zero with respect to other redundant FTs.

Transferred Out indicates if the FT has been transferred to a service provider. This could only happen if the FT is associated with a SA that is associated with a "they bill for us" service provider. Refer to They Bill For Us for more information.

The **Frozen** switch is turned on when the FT has been frozen (i.e., posted to the service agreement's payoff and/or current balances). If this switch is on, both **Freeze Date/Time** and **Frozen By** are populated.

Current Amount contains the FT's impact on the service agreement's current amount.

Payoff Amount contains the FT's impact on the service agreement's payoff amount.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to Current Balance versus Payoff Balance.

Currency Code is the currency code associated with the FT's account.

GL Distribution Status defines the status of the FT in respect of its interface to the general ledger. When an FT is first created, its status is Pending . When an FT is frozen, this value is set to Generated. If you modify the GL details, the status becomes Modified. After it has been marked for interface to the general ledger by the GLS process, the status becomes Distributed.

GL Extract Dates display the **Scheduled** date that the GL details will be marked for interface to the general ledger. Note that this date is typically only set to a future date for FTs associated with automatic payments. **Actual** displays the date that the GL details were actually posted to the general ledger.

The grid at the bottom shows the FT's debits and credits (i.e., the detail journal lines). Debits are shown as positive amounts, credits are negative amounts. The following points describe rules governing if and how the information in the grid can be modified:

- The FT GL details may not be modified if the FT is Pending or Distributed.
- All **Amounts** must sum to zero.
- Multiple FT GL details may be set to affect the **Total Amount** (although they are normally generated with one GL checked to affect total amount).
- All FT GL's checked to affect **Total Amount** must sum to the **Payoff Amount** of the FT.

The following information is displayed:

- **Seq**uence number is the system-assigned, unique identifier of the journal line.
- **Total Amount** defines if the journal line contains the total of the other journal lines. For example, on a bill segment for utility service, this would be turned on for the receivable account because its amount equals the sum of the other payable and revenue GL accounts.
- **Distribution Code** is the CIS distribution code from which the GL account constituents are derived.
- If the **GL Account** has been populated, the GL account number is displayed.

NOT_SUPPORTED

Refer to GLASSIGN - Assign GL Account Numbers To GL Details for more information about how the GL account is populated.

- **Amount** defines the journal line's amount.
- **Statistic Amount** defines the statistical amount that will be posted to the GL. This value is only populated on distribution lines created for calculation rules designated as affecting GL statistical quantity.
- **Characteristic Type** and **Characteristic Value** describe the characteristic value that was used when the line's amount was calculated. This information is only displayed if the journal line was derived from bill calculation lines that were calculated using a bill factor

Copyright © 2026, Oracle and/or its affiliates.

(because only bill factor's use characteristic values). Refer to An Illustration Of A Bill Factor And Its Characteristics for more information.

> **Note**
>
> **Tax reporting.** A journal line's characteristic value is NOT interfaced to the GL. We have included this attribute on the journal line so that tax reporting can be performed from this system (tax reporting typically necessitates showing each taxing authority - the characteristic value - that participated in a given tax payable GL account).

- If you have configured your installation options to indicate that fund accounting is practiced, the description of the **Fund** associated with this distribution code is displayed.

Financial Transaction - FT Process

Use **Menu**, and then **Financial**, and then **Financial Transaction** and navigate to the **FT Process** page to view those ancillary processes that may be triggered as a result of this financial transaction.

> **Note**
>
> **Algorithms cause financial transactions to be linked to batch processes.** Financial transactions get associated with a given process / batch number when certain algorithms are executed. For example, when a bill segment is frozen, the system executes the freeze algorithms associated with the bill segment's service agreement's SA type. One of these algorithms may check if there are service providers who are interested in the bill segment's consumption, and if so, schedule the consumption to be downloaded by linking the bill segment's financial transaction to the service provider's consumption download process.

Description of Page

This page shows the **Batch Processes** associated with a given **Financial Transaction**. Information on this page may not be modified. This information appears purely for audit purposes.

Account Financial History

This portal shows how an account's current and payoff balance have changed over time. You can access the portal from **Menu**, and then **Financial Query**, and then **Account Financial History**.

Refer the portal’s embedded help for more details.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For information about current and payoff balance, refer to Current Amount versus Payoff Amount.

Account Bill / Payment History

This portal shows an account's bills and payments. You can access the portal from **Menu**, and then **Financial Query**, and then **Account Bill / Payment History** .

For balance-forward accounts, bill rows contain the balance presented on the respective bill, and payment rows contain the amount of the respective payment. However, for open-item accounts, this query behaves differently. Refer to the portal's embedded help for more details.

Note

Credit Notes or Correction Notes. The Bill Correction option on the Installation record controls whether Credit Notes or Correction Notes are allowed. The default transaction type filter displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value CRNT on the customizable lookup field TXN_FLTR_TYPE_FLG .

SA Financial History

This portal shows how a service agreement's current and payoff balance have changed over time. You can access the portal from **Menu**, and then **Financial Query**, and then **SA Financial History** .

Refer to the portal's embedded help for more details.

NOT_SUPPORTED

For information about current and payoff balance, refer to Current Amount versus Payoff Amount.

SA Non-Accrual Accounting Balance

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Only relevant if you practice non-accrual accounting.** This transaction is only relevant if you practice non-accrual accounting (e.g., you only pay the taxing authorities when the customer pays you, you defer revenue recognition till a later date, or a combination of the two). Refer to Payables Cash Accounting and Deferred Accrual Accounting for more information about non-accrual accounting.

This page displays a grid that shows a service agreement's non-accrual accounting balance for each non-accrual holding distribution code that's been booked to. Use **Menu**, and then **Financial Query**, and then **SA Non-Accrual Accounting Balance** to open this page.

Description of Page

The following columns are displayed in the grid:

Holding Distribution Code This is a general ledger distribution code used as the holding account for non-accrual accounting. Each holding account that's been used by this service agreement will appear here.

Payable Distribution Code This is the general ledger distribution code to which the payable is (or will be) transferred when the cash event occurs or when the designated revenue recognition date elapses. The system supports revenue recognition on the bill due date.

> **NOT_SUPPORTED**
>
> **For more information** on how the distribution codes are set up, refer to Setting Up Distribution Codes.

Payoff Balance This column shows the service agreement's balance for each non-accrual holding account. If this amount is non-zero, it indicates that for this service agreement we are still holding some payables back until payment is received by the customer, at which time some or all of this amount will be transferred to the payable distribution code.

Accounting Method This column shows which accounting method is used for the distribution.

Accounting Priority This column shows the priority level associated with the distribution code. If more than one distribution code is to receive a payable then the priority level, along with debt age, will determine the order in which the transfers are made.

> **NOT_SUPPORTED**
>
> **For information** about non-accrual accounting, refer to Payables Cash Accounting and Deferred Accrual Accounting.

Balance Control

Copyright © 2026, Oracle and/or its affiliates.

This page is used to summarize the financial transactions that belong to a particular balance control group.

NOT_SUPPORTED

Refer to The Big Picture of Balance Control **for more information**.

Note

Balance Control information is current only as of the Create Date/Time listed. The information displayed on the page is captured when the balance control background process is run. Any financial activity subsequent to the create date and time will not be shown.

Use **Menu**, and then **Financial Query**, and then **Balance Control** to open this page.

Description of Page

This page displays summarized financial information for a particular balance control group. It also details financial information for all SA Types that belong to the group.

The following fields are displayed on the page:

Group ID This is a sequential value that uniquely identifies a particular balance control run. The Group ID is assigned and incremented automatically every time the Balance Control background process is run.

Status This is the status of the balance control record: Pending, and Complete . The status of the balance control record will only be pending while the balance control background process is being executed. Balance control information is only reliable if the status is complete.

Create Date/Time This is the date and time that the balance control record was created. Financial information displayed on this page will only contain information for financial transactions frozen before this date and time.

Current Amount This is the sum of the current amounts of all FTs that belong to the balance control group.

Payoff Amount This is the sum of the payoff amounts of all FTs that belong to the balance control group.

Current Balance This is the sum of the current amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous current balance of the entire CIS as of the create date and time.

Payoff Balance This is the sum of the payoff amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous payoff balance of the entire CIS as of the create date and time.

The number of FT information details the number of bill segments, pay segments, adjustments and their cancellations that belong to the balance control group.

Adjustment The total number of adjustments that belong to this balance control group.

Adjustment Cancellation The total number of adjustment cancellations that belong to this balance control group.

Copyright © 2026, Oracle and/or its affiliates.

Bill Segment The total number of bill segments that belong to this balance control group.

Bill Segment Cancellation The total number of bill cancellations that belong to this balance control group.

Pay Segment The total number of pay segments that belong to this balance control group.

Pay Segment Cancellation The total number of pay cancellations that belong to this balance control group.

The **SA Type** scroll summarizes the FTs that belong to a particular SA Type within the balance control group - otherwise all fields are defined as above. You can scroll through all SA Types that belong to the balance control group.

SA Type The SA Type being summarized.

Match Event

> **Warning**
>
> Match Events are only used if you practice Open Item Accounting.

Match events are used to match debit and credit financial transactions together. When financial transactions are linked to a balanced match event, they no longer affect the customer's arrears.

You can use the match event page to do the following:

- Add, change, delete, and cancel match events.
- Add and remove financial transactions from / to a match event.
- Designate financial transactions as being "in dispute" (disputed financial transactions do not affect aged debt until they are resolved).
- View all unmatched financial transactions linked to an account.

> **NOT_SUPPORTED**
>
> Refer to Match Events for a full description of the lifecycle of a match event and a description of how the system automatically creates match events.

- Match Event - Main
- Match Event - FT Details
- Match Event - Subtotals
- How To Perform Common Match Event Functions

Copyright © 2026, Oracle and/or its affiliates.

Match Event - Main

A typical match event matches a bill's financial transactions with a payment's financial transactions. The **Main** page allows you to define the bill(s) and payment(s) whose financial transactions are matched together. Use **Menu**, and then **Financial**, and then **Match Event**, and then **Search** to open this page.

> **Note**
>
> **Debit must equal credits.** Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every service agreement referenced on the match event. Until that time, the financial transactions on a match event continue to affect arrearage. The only exception is the case of disputes.
>
> **You can match any debit and credit.** While the above describes the matching of bills and payments, it's important to remember that a match event matches any type of debit with any type of credit. This means that a match event could match a bill with a credit or correction note, or a payment with a payment cancellation.
>
> **You can match any number of payments under a match event.** While most match events deal with a single bill and a single payment, there's no limitation to the number of payments on a match event. The only restriction is that the debits and credits must net to zero for all service agreements.
>
> **It is better not to mix multiple bills on a single match event.** For purposes of bill balance information, it is strongly recommended that you compose your match events with financial transactions limited to a single bill. If you mix financial transactions from multiple bills on a single match event you will not be able to determine the unpaid balance of a partially paid bill.
>
> **You can match specific financial transactions.** While most match events deal with every financial transaction on a bill and payment, a match event can deal with individual financial transactions. For example, a match event could match a bill segment with a combination of a payment segment and a write-off adjustment. If you need to add or remove a specific financial transaction (i.e., a bill segment, payment segment, or adjustment), navigate to the **FT Details** tab. Perhaps a better way to differentiate between this page and the **FT Details** tab is to consider the example of a bill with 100 bill segments. When you link this bill to a match event, you are actually linking its 100 financial transactions. If you wanted to add only a subset of this bill's financial transactions to the match event, you'd use the **FT Details** tab.

> **NOT_SUPPORTED**
>
> Refer to How To for instructions describing how to perform common match event maintenance functions.

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Copyright © 2026, Oracle and/or its affiliates.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with the match event for life. **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

Account ID defines the account whose financial transactions are matched under this match event. This field is gray after the match event is added to the database.

Match Event Status defines the state of the match event.

- Match events are initially created in the open state. Please note that you may delete an open match event.
- The system automatically changes an open event's status to balanced when the sum of the **Debit(s)** equals the sum of the **Credit(s)**for each service agreement on the match event. It's worth stressing that a match event may contain financial transactions from many service agreements and each service agreement's financial transactions must sum to zero before the match event can become balanced.
- You may re open a balanced event (by adding / removing items so that the match event becomes unbalanced).
- You can cancel a balanced or open match event by changing the **Match Event Status** to cancelled. You must also define a **Cancel Reason** if you cancel a match event. Refer to How Are Match Events Cancelled? for more information about cancellation.

Turn on **Dispute** if this match event exists to designate certain financial transactions as disputed. In addition,

- Describe the reason for the dispute in **Remarks**.
- Link the disputed financial transactions to the match event by selecting the bill(s) below. If only a subset of a bill is disputed:
- Click on the respective hyperlink in the **Unmatched Debits** column grid (this will transfer you to the next tab with these financial transactions displayed).
- Select the bill segments that you want to designate as disputed.
- Press the **Link / Unlink** button to link the selected bill segments to the match event.

> **Note**
>
> **Disputing items on a balanced event.** The **Dispute** switch will be protected when the match event is Balanced or Cancelled. If the items on a balanced event are being disputed, you must add / remove items so that the match event becomes Open before you can turn on the **Dispute** switch.

The remainder of the page is dynamic depending on the **Match Event Status**:

- If the status is Balanced or Open, a grid appears containing a summary of the bills, payments, credit or correction notes, and payment cancellations that contribute financial transactions to the match event (note, we refer to these collectively as "contributing objects" in the following discussion). The following columns appear in this grid:
- **Transaction Type** defines whether the contributing object is a Bill, Payment, Payment Cancellation, or Credit or Correction Note. In the rare situation when unbilled financial

Copyright © 2026, Oracle and/or its affiliates.

transactions are linked to the match event, a transaction type of Not Billed Yet appears. If the contributing object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.

- **Matched Activity Information** contains summary information about the contributing object. This column is blank if the transaction type is Not Billed Yet.
- The remaining columns contain the count and amount of each contributing object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
- **Matched Debits** summarizes the contributing object's debit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be removed from the match event.
- **Matched Credits** summarizes the contributing object's credit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these credits will be removed from the match event.
- **Other Debits** summarizes the contributing object's debit FT's that are NOT linked to this match event.
- **Other Credits** summarizes the contributing object's credit FT's that are NOT linked to this match event.

> **Note**
>
> **Adding more financial transactions to a balanced match event.** When a match event is Balanced, the grid showing unmatched FT's is suppressed (and therefore you can't add additional FT's to the match event). To expose this grid, simply change the status of the match event to Open.

- If the status is Open, a second grid appears containing a summary of the bills, payments, credit notes, correction notes and payment cancellations with at least one unmatched financial transaction (we refer to these collectively as "unmatched objects" in the following discussion). The following columns appear in this grid:
- **Transaction Type** defines whether the unmatched object is a Bill, Payment, Payment Cancellation, Credit, or Correction Note. A transaction type of Not Billed Yet is used for unbilled financial transactions (e.g., adjustments generated between bills). If the unmatched object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.
- **Unmatched Activity Information** contains summary information about the unmatched object. This column is blank if the transaction type is Not Billed Yet.
- The remaining columns contain the count and amount of each unmatched object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
- **Unmatched Debits** summarizes the unmatched object's debit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be added to the match event.
- **Unmatched Credits** summarizes the unmatched object's credit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the

Copyright © 2026, Oracle and/or its affiliates.

checkbox and press the **Link / Unlink** button, these credits will be added to the match event.

- **Matched Debits** summarizes the unmatched object's debit FT's that are linked to any match event.
- **Matched Credits** summarizes the unmatched object's credit FT's that are linked to any match event.

> **Note**
>
> **Credit Notes or Correction Notes.** The Bill Correction option on the Installation record controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value CRNT on the customizable lookup field TXN_FLTR_TYPE_FLG.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the above grids. If debits and credits do not sum to zero, the **Difference** is also shown. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated when the database is changed).

The **Link / Unlink** button becomes enabled when you select an object in the grids. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - FT Details

On the **Main** tab, you can add and remove bills, payments, credit notes, correction notes and payment cancellations to / from a match event. When you do this, you are actually adding and removing all of the financial transactions linked to these objects. For example, when you link a bill with 100 bill segments to a match event, you are actually linking its 100 financial transactions.

You need only use the **FT Details** tab when you need to add or remove specific financial transactions. For example, you would use the **FT Details** tab if you need to remove 1 of a bill's 100 financial transactions from a match event.

Use **Menu**, and then **Financial**, and then **Match Event**, and then **Search** and navigate to the **FT Details** page (note, you can also open this page using many of the hyperlinks on the Main and SA Subtotals tabs).

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

Copyright © 2026, Oracle and/or its affiliates.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the financial transactions that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use **Transaction Type Filter** to restrict the type of transactions that appear in the grid. The following options are available:
- All. Use this option if you do not wish to restrict financial transactions based on their transaction type.
- Bill. This option allows you to view a specific bill's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID**.
- Credit Note. This option allows you to view a specific credit note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every credit note has a unique bill ID).
- Correction Note. This option allows you to view a specific correction note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every correction note has a unique bill ID).
- Not Billed Yet. This option allows you to view financial transactions that haven't appeared on a bill yet (e.g., adjustments and corrections).
- Payment. This option allows you to view a specific payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.
- Payment Cancellations. This option allows you to view a specific canceled payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.

> **Note**
>
> **Credit Notes or Correction Notes.** The Bill Correction option on the Installation record controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value CRNT on the customizable lookup field TXN_FLTR_TYPE_FLG.

- Use **Linkage Filter** to restrict the transactions based on the match event to which they are linked. The following options are available:
- All. This option shows all financial transactions regardless of their match event.
- Linked to another Match Event. This option shows financial transactions linked to a different match event.
- Linked to any Match Event. This option shows financial transactions linked to any match event.
- Linked to this Match Event. This option shows financial transactions linked to this match event.
- Unmatched. This option shows financial transactions that are not linked to a match event. This filter is protected and set to Linked to this Match Event if the **Transaction Type Filter** is All.

Copyright © 2026, Oracle and/or its affiliates.

- Use **Debit / Credit Filter** to restrict the transactions based on whether they are debits or credits. The following options are available:
- All. This option shows all debit and credit financial transactions.
- Credit. This option shows all credit financial transactions.
- Debit. This option shows all debit financial transactions.
- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
- Address. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you do not wish to restrict financial transactions based on service agreement attributes.
- Geographic Type. Use this option to restrict financial transactions to those whose service agreements are linked to service points associated with a given **Geo Type** and **Value**.
- SA ID. Use this option to restrict financial transactions to those linked to a specific **Service Agreement**.
- SA Type. Use this option to restrict financial transactions to those whose service agreements are linked to a given **CIS Division** and **SA Type**.

> **Warning**
>
> Don't forget to click the search button after changing the filters.

The **Select All** / **Clear All** buttons are used to select financial transactions to add to / remove from the match event (note, after selecting the desired financial transactions, you must also press the **Link / Unlink** button at the bottom of the page).

> **Note**
>
> **50 financial transactions at a time.** Clicking **Select All** selects the first 50 bill segments in the grid. If more than 50 financial transactions exist, you must select them in batches.

The grid that follows contains the financial transactions (FT) that match your search criteria. The following information appears in the grid:

- Select box. Select the FT if you want to Link / Unlink it. You implicitly link FT's that are NOT already linked and you implicitly unlink FT's that are already linked. This field is protected if the FT is linked to another match event.
- **FT Amount.** This column contains the amount of the financial transaction.
- **FT Type.** This column indicates the type of financial transaction: Bill Segment, Bill Segment Cancellation, Pay Segment , Pay Segment Cancellation, Adjustment, and Adjustment Cancellation .
- **Arrears Date.** This column contains the financial transaction's arrears date.

Copyright © 2026, Oracle and/or its affiliates.

- **SA Information.** This column contains a summary of the financial transaction's service agreement.
- **Remarks.** This column highlights the financial transaction's match status: Linked to this match event, Not linked to a match event, Linked to another match event - match event status.
- **Premise Information.** This column contains a summary of the premise (if any) associated with the financial transaction's service agreement.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the grid. If debits and credits do not sum to zero, the **Difference** is calculated. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated with the database is changed).

The **Link / Unlink** button becomes enabled when you select a row in the grid. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - Subtotals

Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every service agreement referenced on the match event. This page shows the sum of the debits and credits for every service agreement that contributes at least one financial transaction to the match event. Use **Menu**, and then **Financial**, and then **Match Event**, and then **Search** and navigate to the **Subtotals** page

Description of Page

This page shows the sum of the debits and credits for every service agreement that contributes at least one financial transaction to the match event.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the service agreements that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use the **SA Filter** to define the types of service agreements whose financial transactions appear in the grid. The following options are available:
- Address. Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- All. Use this option if you do not wish to restrict service agreements.
- Geographic Type. Use this option to restrict service agreements to those linked to service points associated with a given **Geo Type** and **Value**.
- SA ID. Use this option to see a specific **Service Agreement**.

Copyright © 2026, Oracle and/or its affiliates.

- SA Type. Use this option to restrict service agreements to those with a given **CIS Division** and **SA Type**.
- Use **Status Filter** to restrict the service agreements based on whether the sum of the debits and credits they contribute to the match event. The following options are available:
- All. Use this option if you do not wish to restrict service agreements based on this status.
- Balanced. This option shows only service agreements where the sum of debit and credits nets to zero on this match event.
- Unbalanced. This option shows only service agreements where the sum of debit and credits do not net to zero on this match event.

> **Warning**
>
> Don't forget to click the search button after changing the filters.

The grid that follows contains the service agreements that match your search criteria. The following information appears in the grid:

- **SA Information.** This column contains a summary of important information about the service agreement.
- **Difference.** This column contains the difference between the **Matched Debits** and **Match Credits**. If this is non-zero, the value appears in red.
- **Matched Debits.** This column contains the sum of debit financial transactions that this service agreement contributes to this match event.
- **Matched Credits.** This column contains the sum of credit financial transactions that this service agreement contributes to this match event.
- **Premise Information.** This column contains a summary of the premise (if any) associated with the financial transaction's service agreement.

How To Perform Common Match Event Functions

- How To Find The Match Event Associated With A Financial Transaction
- How To Dispute An Item
- How To Match A Small Mismatch

How To Find The Match Event Associated With A Financial Transaction

If you need to find the match event on which a financial transaction (FT) was matched, display the FT in question on Financial Transaction - Main and then use the "go to" button adjacent to the **Match Event** to drill to the match event.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The easiest way to display a financial transaction is to find its corresponding bill, payment or adjustment and then drill down on the desired transaction.

How To Dispute An Item

Refer to Disputing Items for background information about disputes.

If a customer wants to dispute an item:

- Turn the match event's dispute switch on.
- Link the disputed financial transaction to the match event.
- Describe in the match event's comments the reason for the dispute.

How To Match A Small Mismatch

Assume the following scenario arises:

- A bill is produced for $2000
- The customer pays $1993
- An unbalanced match event will result because the customer didn't pay exactly what is owed

If you want to match this payment to the bill (and leave $7 for the next bill), do the following:

- Create a transfer adjustment of $7 where the transfer from / to service agreement is the same. This results in a debit financial transaction (FT) of $7 and a credit FT of $7.
- Create a match event (or update the unbalanced match event) where the matched FTs are:
- The $1993 payment
- The credit side of the transfer adjustment ($7)
- And the $2000 bill
- Then, if the customer pays their next bill in full, the $7 debit (associated with the transfer adjustment) will be swept onto it.

> **Note**
>
> **Automating small mismatches.** The algorithm responsible for matching a payment to a specific bill can have a tolerance amount defined on it. If the payment is within the tolerance limit, this algorithm will do the above for you. In other words, you don't have to manually do the above if you populate the tolerance limit appropriately on this algorithm. Refer to DSOV BILL-ID for more information about this algorithm.

Copyright © 2026, Oracle and/or its affiliates.

Deposits

In this section, we describe how to manage your deposits.

- The Big Picture Of Deposits
- Deposit Review

The Big Picture Of Deposits

The topics in this section provide background information about a variety of deposit issues.

> **NOT_SUPPORTED**
>
> We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate how deposits fit into the system's financial architecture.

- Cash Deposits
- Refunding Deposits
- Partial Refunds
- Total Amount To Bill
- Current Balance versus Payoff Balance
- Multiple Deposits Linked To A Single Account
- Deposits and Credit & Collections
- Deposit Seizures When Other SAs Are Stopped
- Non-Cash Deposits
- Third Party Deposits
- Deposit Class Controls Everything
- Deposit Background Processes

Cash Deposits

Before you can bill a customer for a deposit, there must be a deposit service agreement (SA). Deposit SAs will be created by a CSR using Start/Stop Service (the same page used to create all other types of service agreements).

Deposit SAs behave just like any other service agreement in that:

- A deposit SA must reference an SA type. Refer to Service Agreement Type Controls Everything for information about how SA type controls a service agreement's behavior.

Copyright © 2026, Oracle and/or its affiliates.

- Bill segments will be produced to bill the customer for a deposit. Note, bill segments associated with deposit SAs will appear on the same bill as other bill segments related to its account (i.e., a bill could contain a combination of utility, deposit, and non-utility bill segments).
- When a customer makes a payment, it will be distributed amongst its account's SAs based on each SA's payment distribution priority. This means a single payment could relieve receivable accounts (associated with utility service agreements) and increase a payable account (associated with the deposit service agreement). Refer to Distributing A Payment Amongst An Account's Service Agreements for more information.
- You can use a transfer adjustment to relieve debt on some other type of service agreement (by transferring all or part of the deposit's credit balance to another service agreement).
- You can use an A/P adjustment to cause a check to be created if you need to manually refund a deposit.
- When you no longer require the deposit, a CSR will stop the deposit SA using Start/Stop Service (the same page used to stop all other types of service agreements). Refer to Refunding Deposits for information about how the refund actually takes place.

NOT_SUPPORTED

Refer to Deposit Class Controls Everything **for information** about deposit interest, the recommended deposit amount algorithm, and how deposits are refunded.

Note

An account can have many deposits. You can view all deposits (both cash and non-cash) linked to an account on Account - Deposits .

Refunding Deposits

To refund a deposit, you simply need to change the deposit SA's state to pending stop. The system does everything else (i.e., calculates interest and refunds the deposit to the customer).

There are three ways in which a deposit SA's state can become pending stop:

- The Deposit Refund background process (DEPRFND) make a deposit service agreement pending stop when it detects that the customer meets the automatic refund criteria. Refer to Defines The Conditions That Cause The System To Automatically Refund A Deposit for a description of how to define the conditions that control automatic refund.
- An operator, at their discretion, may refund a deposit by stopping the deposit SA using Start/Stop Service (the same page used to stop all other types of service agreements). This page simply changes the deposit SA's state to pending stop.
- The system changes the state of a deposit service agreement to pending stop when it recognizes that all other SAs of its deposit class have also been stopped. This may be necessary, for example, when a customer's utility service agreements have been severed for non-payment. Refer to Deposit Seizures When Other SAs Are Stopped for more information.

Copyright © 2026, Oracle and/or its affiliates.

The system changes the state of a pending stop SA to stopped when the process responsible for Finalizing Pending Stops next runs.

When a deposit SA becomes stopped, the system calculates interest through the stop date. Refer to Controls Interest Calculation for a description of how interest is calculated.

> **Note**
>
> If you can't wait for this process to run, simply display the deposit SA on Service Agreement - Main Information and press the stop button.

When the stopped deposit SA is final billed, the system produces a final bill segment that shows the deposit amount being returned to the customer. When the bill segment's bill is completed, the deposit is refunded to the customer as per the deposit refund algorithm on the deposit's deposit class. Refer to Controls How Deposits Are Refunded To A Customer for a description of how a deposit is refunded.

> **Warning**
>
> None of the base package deposit refund algorithms create an A/P adjustment when the deposit service agreement is final billed. Rather, they depend on the write-off preprocessing to actually refund the deposit to the customer.

Partial Refunds

You can perform a partial refund by creating an adjustment for the deposit SA. If you want to refund the adjustment with a check, this adjustment should be an A/P adjustment (i.e., one that is interfaced to A/P). If you want to refund the adjustment by offsetting some other service agreement's debt, create a transfer adjustment (transferring from the deposit SA to the SA to be relieved).

After creating the adjustment, you'll find that the deposit SA has a positive current amount due (because you debited the deposit SA to refund the credit amount). You should create another adjustment to return the current amount of the deposit to zero. It's important to use an adjustment type that only affects current amount when you do this.

After getting the deposit's current amount back to an expected amount, you will have to change the deposit SA's Total Amount To Bill to reflect the amount of deposit you want to hold for the customer. So, for example, if you were holding a $500 deposit and you want to refund $100, you should change the deposit SA's Total Amount To Bill to be $400.

Total Amount To Bill

Most deposit service agreements contain a "Total Amount To Bill" field. The label for this field is defined on the SA type (on the Billing tab). For deposit SAs, it should be labeled something liked **Cash Deposit Amount**. This field should reflect the deposit amount you WANT to hold on this service agreement. Please be aware of the following in respect of this field:

Copyright © 2026, Oracle and/or its affiliates.

- If the payoff balance (i.e., the current deposit held) is less than the Total Amount To Bill, the system will generate a bill segment to charge the customer for the deposit.
- If the deposit service agreement uses a recurring charge (i.e., you bill the total amount in installments), the amount billed to the customer will not exceed the recurring charge amount (unless the bill segment is prorated).
- If the deposit service agreement does not use a recurring charge, the amount billed will equal the difference between Total Amount To Bill and the service agreement's payoff balance.
- If you need to bill an incremental deposit on an existing deposit service agreement, simply change the deposit service agreement's Total Amount To Bill.
- If the payoff balance (i.e., the current deposit held) is greater than the Total Amount To Bill, the system will NOT refund the excess deposit.

> **Note**
>
> **SA Type and Total Amount to Bill.** A deposit service agreement's SA type controls if the deposit service agreement uses Total Amount to Bill. Typically, the only SA types that indicate this field can be used are loan and deposit service agreements.

Current Balance versus Payoff Balance

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

The current balance on a deposit SA contains the amount of the deposit that has been billed, but not paid. The payoff balance on a deposit SA contains the amount of deposit being held. The payoff balance is a negative number because your company owes this money to the customer.

The financial ramifications of a deposit SA are predictable (if you're an accountant). The following table outlines the different financial events and their impact on the general ledger, arrearage history, and the amounts due (both current and payoff).

> **Note**
>
> It's important to be aware that everything that is shown in this table is controlled by how you set up the deposit SA type's bill segment type, payment segment type, and adjustment types. Refer to Service Agreement Type Controls Everything for how to do this.

Event	GL Accounting	Arrearage Rule	Effect On Payoff Amt	Effect On Current Amt	Payoff Balance	Current Balance

Copyright © 2026, Oracle and/or its affiliates.

Deposit billed	N/A - the GL is not affected when a deposit is billed	$100 starts aging	0	+100	0	100
Payment received	Cash 100 Deposit Payable <100>	$100 relieved accordingly	-100	-100	-100	0
Interest calculated	Interest Exp 5 Deposit Payable <5>	N/A	-5	0	-105	0

The following points describe the events in the above table:

- **Deposit billed.** In this example, the customer is billed for $100 deposit.
- The customer really thinks they owe the billed amount, $100. Therefore, current amount is affected. However, if the customer was to cash out, they wouldn't owe your organization anything, therefore payoff amount is not affected.
- Notice that the GL is not affected when the deposit is billed. This is because most organizations do not show a receivable for billed deposits (as it's not a true receivable).
- Because current amount changed by $100, arrearage history is affected accordingly.
- **Payment received.** With any luck, the client will pay the $100 that was billed.
- The payment has a normal affect on the GL (debit cash, credit deposit payable). The deposit payable GL account is defined as the deposit SA type's Distribution Code.
- The amount the customer thinks they owe decreases by $100, therefore current amount is affected by the payment amount. And, if the customer was to cash out, your organization would owe the customer $100, therefore payoff amount is affected by the payment amount.
- Because current amount changed by $100, arrearage history is affected accordingly.
- **Interest calculated.** In this example, the system calculates interest of $5.
- The interest is posted to the GL (the interest expense distribution code is defined on the respective adjustment type).
- The interest amount didn't affect how much the customer thinks is due them. Therefore current amount is unaffected. However, if the customer was to cash out, your organization would owe them $100 + $5 (the interest) therefore payoff amount is affected by $5.
- Because current amount is not changed, arrearage history is not affected.

Multiple Deposits Linked To A Single Account

The system has been designed to allow your customers to have multiple deposits. This is necessary when a deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one deposit for regulated debt and another for unregulated debt.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Deposit class.** A deposit's deposit class controls the service agreements that are covered by a given deposit. A deposit's deposit class is defined on its SA type. Refer to Designing Your Deposit Classes for more information.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend and refund deposits for each individual deposit class. For example, if you have both regulated and unregulated deposit classes, the system will recommend separate deposits for each class. And at refund time, a deposit will be distributed to only those service agreements associated with its deposit class (i.e., a regulated deposit will not be used to satisfy unregulated debt).

Deposits and Credit & Collections

The account debt monitor monitors deposit service agreements just as it monitors every other service agreement for overdue debt. If the amount of debt on the deposit SA violates your collection criteria, a collection process will start.

> **NOT_SUPPORTED**
>
> Refer to The Lifecycle Of A Collection Process And Its Events **for more information**.

It's important to be aware that deposit debt can be treated differently from other types of debt linked to an account by creating a specific debt class for the deposit SA type.

> **NOT_SUPPORTED**
>
> Refer to Designing Your Collection Procedures for more information about how to have different collection criteria for different debt classes.

The type of severance process associated with a deposit SA is probably rather simple - you probably will want a To Do entry generated to advise an operator that a customer hasn't paid their deposit.

> **NOT_SUPPORTED**
>
> Refer to The Lifecycle Of A Severance Process And Its Events **for more information**.

Please see Deposit Seizures When Other SAs Are Stopped for a description of how the system seizes an active deposit service agreement when normal (i.e., non-cash deposit) service agreements are stopped.

After a deposit SA is stopped, it will be "final billed" the next time the account is billed. Refer to Refunding Deposits for more information about how deposit refunds are "final billed". If a credit

Copyright © 2026, Oracle and/or its affiliates.

balance remains on the deposit SA after it is final billed, the Write-Off Monitor will process the deposit SA just like it processes all unpaid and final billed service agreements. If you set up the deposit SA's write-off controls properly (i.e., you plug in the appropriate refund and write-down algorithms), the write-off monitor will refund / write-down the credit balance (thus causing the deposit SA to close).

Deposit Seizures When Other SAs Are Stopped

Consider the situation when a severance process cuts a service due to non-payment. At some point, if the customer doesn't pay, you'll want to seize the deposit and apply it to the overdue debt. To implement this, the system simply stops the deposit SA when it stops the last SA in its deposit class (refer to Finalizing Pending Stops for more information about stopping service agreements). Then, when the account is next billed, the standard deposit refund process will offset outstanding debt before it is refunded to a customer.

Refer to Refunding Deposits for a complete description of how the system refunds a stopped deposit SA's balance to other service agreements in its deposit class.

> **Note**
>
> **Deposit class refund method is the key.** It's important to stress that the only way the system will use a deposit to offset overdue debt is if the deposit's deposit class has a refund method that does this. If you've set up a deposit class' refund method to cut a check for the entire deposit amount, the system will not offset outstanding debt. Refer to Deposit Class - Refund Method for more information.

Non-Cash Deposits

When an account is required to post a deposit, they can remit cash or some other type of surety (e.g., letters of credit, surety bonds, 3 rd party deposits). When cash is remitted, a deposit service agreement is used. When some other type of surety is used, a *non-cash deposit* must be created.

Non-cash deposits are held in respect of an account and an account may have an unlimited number of non-cash deposits (note: cash deposits are held in respect of a deposit service agreement that is linked to remitting account). Refer to Account - Deposits for more information.

Each non-cash deposit must reference a non-cash deposit type. Besides defining the type of surety, the non-cash deposit type also controls:

- Whether such types of non-cash deposits can be highlighted when they are due to expire via the creation of a To Do entry (refer to TD-NCDEX for more information about this To Do entry).
- Whether a reference to a cash-deposit (i.e., service agreement) must be specified. Refer to 3rd Party Deposits for more information about how a cash deposit can be referenced by another party.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Setting Up Non-Cash Deposit Types for more information.

In addition to non-cash deposit type, each non-cash deposit must reference a deposit class. Why? Because the system amalgamates cash and non-cash deposits when it determines if an account is holding an adequate deposit.

NOT_SUPPORTED

Refer to What Do Deposit Classes Do? for more information.

If you need to use a non-cash deposit to satisfy an account's debt, you must "cash in" the non-cash deposit. When the cash arrives, create a payment and apply it to the customer's outstanding service agreements.

Third Party Deposits

A third party deposit arises when a third party remits cash to cover the deposit needs of one or more accounts. Both cash and non-cash deposits must be created to record a third party deposit. The following example will explain how to do this.

Assume the Salvation Army remits $5,000 on behalf of 50 customers (where each customer's account is allocated $100). In this situation, you'd create the following information in the system:

- The Salvation Army must have an account with a deposit service agreement.
- When the Salvation Army remits the funds:

Warning

If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- Create an adjustment to "bill" the deposit service agreement (causing the deposit SA's current balance to be $5,000 and the payoff balance to be 0).
- Add a payment for the $5,000 against the Salvation Army's account. This payment will cause the deposit SA's current balance to be 0 and the payoff balance to be -$5,000. Note, if you don't mind the Salvation Army's current balance to be -$5,000 after the payment is made, you wouldn't have to create the adjustment to "bill" the deposit.
- Create a non-cash deposit for each of the 50 accounts being covered by the cash deposit. On each non-cash deposit, define the appropriate non-cash deposit type (e.g., 3rd party deposit) and amount $100 each.
- Interest will be applied to the Salvation Army's deposit service agreement as per the interest algorithm on the deposit SA's deposit class.

Copyright © 2026, Oracle and/or its affiliates.

- If you need to use the Salvation Army's payment to payoff overdue debt, you will use a transfer adjustment(s) to transfer from the deposit service agreement to the respective overdue service agreement(s).

> **Important**
>
> Be aware that the system will allow the sum of third party deposits to exceed the amount of the cash deposit (in our previous example, the system would allow you to create $6,000 worth of third party non-cash deposits even though only $5,000 of cash was remitted).

Deposit Class Controls Everything

A deposit service agreement's SA type references a deposit class. The topics in this section describe how deposit class controls the behavior of a deposit SA.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Deposit Classes for a complete description of the business rules governed by a deposit class.

- Controls Interest Calculation
- Controls The Recommended Deposit Amount
- Defines The Conditions That Cause The System To Recommend A New or Additional Deposit
- Defines Automatic Refund Conditions
- Controls How Deposits Are Refunded To A Customer

Controls Interest Calculation

Interest is applied to a deposit SA when:

- A deposit SA is stopped. Refer to Refunding Deposits for a description of how a deposit SA can be stopped.
- The Apply Interest background process (DEPINTRF) determined that sufficient time has passed since interest was last calculated. The amount of time between interest calculations is defined on the deposit class.
  A deposit SA's deposit class controls the following interest calculation functions:
  - The frequency of interest calculation.
  - The algorithm used to calculate the amount of interest (note, the interest rate is defined using a bill factor plugged in on this algorithm).
  - The method used to refund the interest.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Adjustments are used to apply interest.** Interest is applied to a deposit SA using an adjustment (the adjustment type is also defined on the deposit class).

Controls The Recommended Deposit Amount

The following events cause the calculation of the recommended deposit amount:

- The Deposit Review background process (DEPRVW) compares an account's existing deposit (if any) to the recommended amount.
- A user requests a deposit calculation for a new deposit service agreement by pressing the  button on the Start Confirmation.

A deposit service agreement's deposit class controls the algorithm used to derive the suggested deposit amount.

Defines The Conditions That Cause The System To Recommend A New or Additional Deposit

The Deposit Review background process (DEPRVW) will recommend an additional deposit be billed if the customer doesn't satisfy the definition of a *good customer*.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend deposits for each individual deposit class used by an account. For example, if an account has both regulated and unregulated deposit classes, the system will recommend separate deposits for each class.

When the Deposit Review process analyzes an account's debt, it uses the following deposit class attributes:

- The definition of a good customer (the system only recommends deposits for bad customers).
- The tolerance percent that must be exceeded before the system will recommend an additional deposit for a customer. This tolerance percent prevents the recommendation of small amounts.
- The method used to derive the recommended deposit amount.

Assuming the C1-CR-UP-DRR review method algorithm is specified on the deposit class, the system does not automatically produce bills when an additional deposit is needed. Rather, the system's recommendations appear on the Deposit Review page. To implement a recommendation, an operator should change the deposit SA's Total Amount To Bill or add a new deposit SA.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Non-cash deposits are included.** It's important to be aware that the system compares the recommended deposit amount against the sum of deposits on hand for a given deposit class. When amalgamating the total deposit on hand, the system includes both cash (i.e., deposit SA's payoff balance) and non-cash deposits.

Defines Automatic Refund Conditions

The Deposit Refund background process (DEPRFND) will refund a deposit to a customer when conditions defined on the deposit SA's deposit class are met. For example, a deposit's deposit class may indicate the system should automatically refund a deposit after the deposit has been held for 6 months and the customer is a good customer (the definition of a "good customer" is also on the deposit class).

Controls How Deposits Are Refunded To A Customer

The method used to refund a deposit to a customer is defined on the deposit SA's deposit class. For example, a deposit's deposit class may indicate the system will first apply the deposit to outstanding debt first and, if funds remain, cut a check.

It's important to be aware that if your company refunds deposits by first offsetting outstanding debt, the system will only offset debt within an individual deposit class. For example, if a customer has two deposits - one for regulated debt, the other for unregulated debt - the regulated deposit will only be used to offset regulated debt.

Deposit Background Processes

The topics in this section describe the background processes that automate deposit processing.

- Deposit Interest
- Deposit Refund
- Review Deposits

Deposit Interest

The Deposit Interest background process (DEPINTRF) examines all cash deposit service agreements (i.e., service agreements with a SA type with a special role of Cash Deposit). If enough time has passed since interest was last calculated, interest will be calculated using the service agreement's deposit class' Interest Refund Algorithm. Note: interest will be automatically calculated every X months where X is defined in the deposit class' Months Between Interest Refund.

Copyright © 2026, Oracle and/or its affiliates.

Deposit Refund

The Deposit Refund background process (DEPRFND) examines all cash deposit service agreements (i.e., service agreements with a SA type with a special role of Cash Deposit). If the account meets the service agreement's deposit class' Refund Criteria Algorithm, the system changes the service agreement's state to Pending Stop. Refer to Refunding Deposits for a description of what happens next.

Review Deposits

The Review Deposits background process (DEPRVW) examines all accounts with service agreements that are governed by a deposit class (i.e., service agreements with an SA type that references a deposit class). If the account fails the deposit class' Good Customer Algorithm, the system calculates the recommended deposit amount (using the deposit class' Deposit Recommendation Algorithm). If the recommended amount exceeds the amount of deposit currently requested, the system will request an additional deposit. The deposit class' **Review Method Algorithm** is used to determine what action to take if the system requests an additional deposit. The system's recommendations appear on the Deposit Review page.

You may optionally provide a Deposit Class as input to restrict the review to account whose service agreements reference that deposit class.

> **Note**
>
> **Preventing small recommendations.** The system uses the deposit class's Review Tolerance Percentage to prevent the recommendation of small deposits by the Deposit Review background process. For example, if this field contains 10(%), the system would only recommend an additional deposit if the existing requested deposit amount is less than 90% of the recommended amount.
>
> **Multiple deposits could be recommended for an account.** Because an account can have service agreements that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Deposit Review

The Deposit Review page shows all accounts for which an additional deposit is recommended.

> **Note**
>
> The Review Deposits background process inserts rows on this query.
>
> **Multiple deposits could be recommended for an account.** Because an account can have service agreements that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Copyright © 2026, Oracle and/or its affiliates.

Open this page using **Menu**, and then **Financial Query**, and then **Deposit Review.**

Description of Page

Use **Account ID** and / or **Deposit Class** to filter the rows. Each row displays and account / deposit class that holds an inadequate deposit.

> **Note**
>
> **Rows are also displayed for errors.** Please be aware that if the Recommend Additional Deposit background process encounters an error when it attempts to calculate an account's recommended deposit, these errors also appear as rows in this query.

Press the  button to add a new deposit SA for the customer. Pressing this button causes Start/Stop Service to open. When this page opens, you should create a new deposit service agreement for the customer (by referencing the appropriate division / SA Type).

If the customer has one or more deposit service agreements, the drill button will be enabled. Pressing it will cause the Service Agreement - Main Information page to open. When this page opens, you can change Total Amount To Bill to reflect the total amount of deposit you want to hold. When billing next runs, it will bill the customer for the marginal difference between the current deposit held and the deposit amount to bill.

> **NOT_SUPPORTED**
>
> Refer to Defines The Conditions That Cause The System To Recommend A New or Additional Deposit for how rows are added to this query.

Statements

In this section, we describe how to manage your customer's statements.

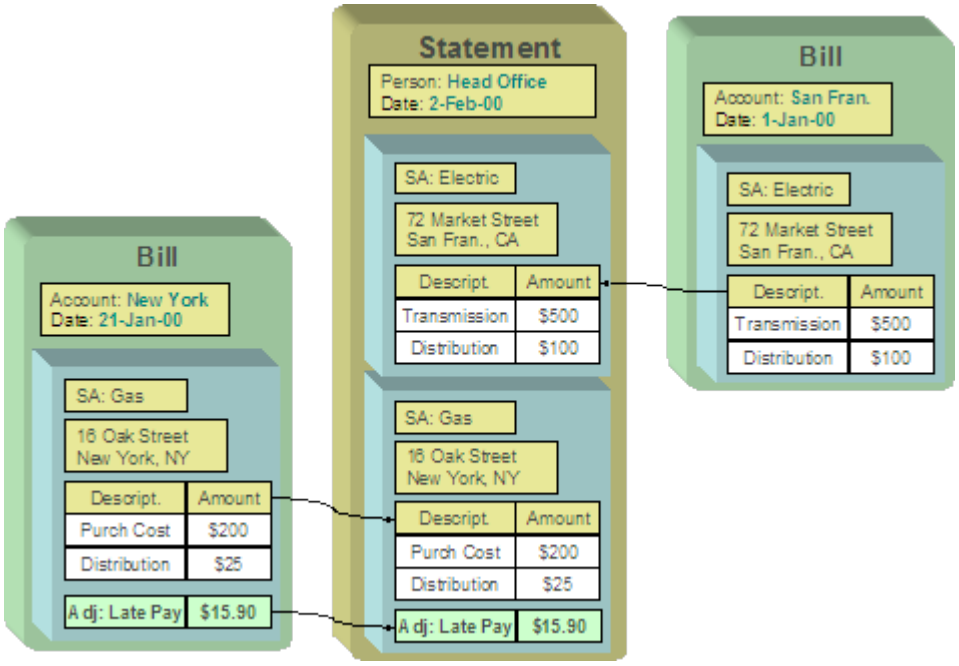
- The Big Picture of Complex Statements
- Statement Construct Maintenance
- Statement Maintenance

The Big Picture of Complex Statements

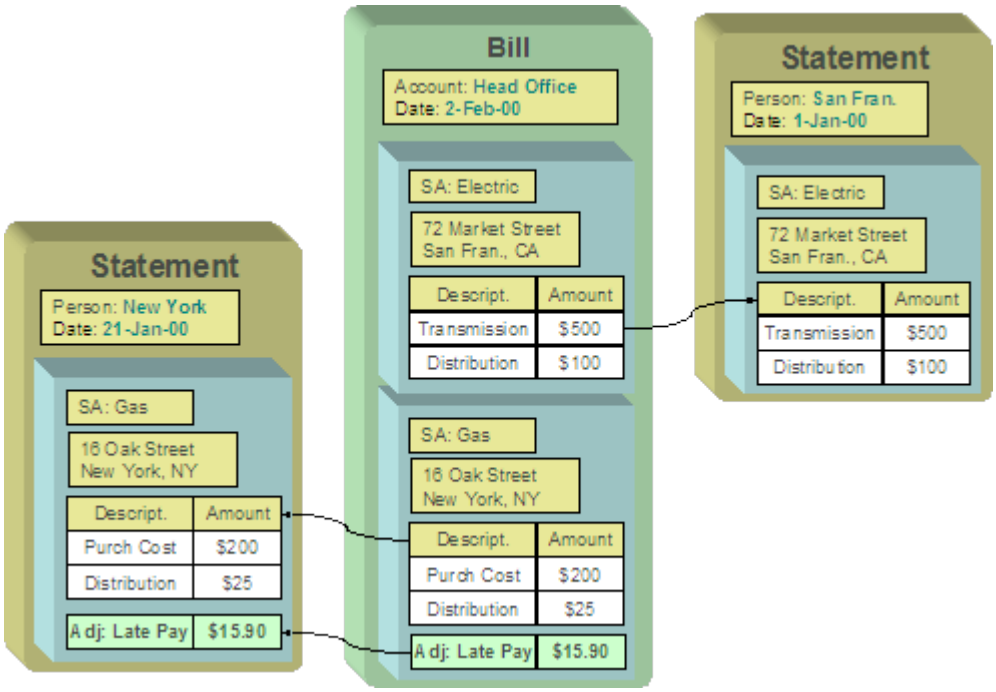
Statements allow you to set up a person to receive a consolidated report of the financial activity for one or more accounts and/or service agreements. This allows you to model complex account scenarios, including the following examples.

Copyright © 2026, Oracle and/or its affiliates.

- Satellite offices for a company are each responsible for paying their own bill, but the head office would like a consolidated report of the bills for all the satellite offices once a month. In this case, each satellite office is set up as an account covering the services for their office. The head office does not want to receive individual copies of each bill, but rather a consolidated report of the financial activity for all the offices.

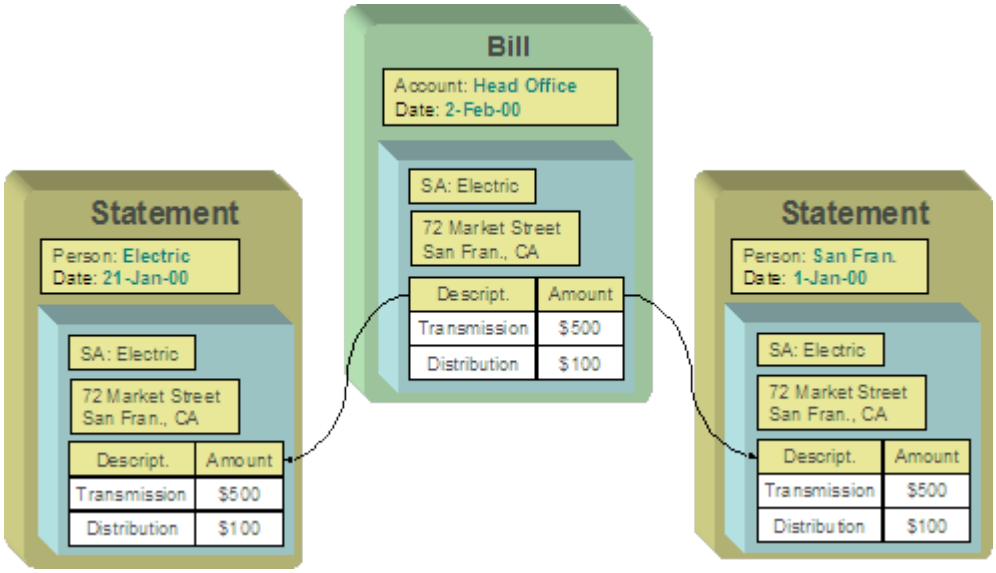


- The head office is responsible for paying the bill. The satellite offices should receive a copy of only their portion of the bill. In this case, a service agreement is set up for each satellite office under the head office's account. Each satellite office should receive a copy of the bill segment information for their office's service agreement.



Copyright © 2026, Oracle and/or its affiliates.

- The head office is responsible for paying the bill. Multiple departments need to see copies of the electric portion of the bill. As illustrated by this scenario, a service agreement or account may appear on many statements.



The topics in this section provide additional information about how to set up statements.

- Constructing Statements
- Producing Statements
- Printing Statements

Constructing Statements

Statements are produced for a person. Each statement contains the financial transactions for one or more service agreements. The service agreements may belong to any number of accounts. Over time, a person may receive many statements.

When setting up a person to receive statements, it's important to note that a person can receive different statements for different groupings of service agreements. For example, the head office may want a statement of the financial details for the western division offices separately from the eastern division offices.

For each separate statement a person wishes to receive, you will set up a statement construct. On the statement construct, you define the service agreements and/or accounts whose financial information should appear on the statement. If you specify an account on a statement construct, billing information for all the service agreements linked to the account will be included on the statement. The superset of service agreements and accounts defined on a statement construct are called the "construct details".

NOT_SUPPORTED

Refer to Statement Construct Maintenance for more information.

Copyright © 2026, Oracle and/or its affiliates.

Producing Statements

- Statement Cycles
- Create Statements Background Process
- Statement Routing
- Where Are Statements Sent
- On-Line Statement Production
- How to Regenerate a Specific Statement

Statement Cycles

The accounts and service agreements linked to a statement construct may have different bill cycles and as a result, their bills are produced on different dates. So when should a statement containing the consolidation of all this financial information be produced? To answer this question, there is a statement cycle.

Just as with a bill cycle, the statement cycle has a schedule. A statement cycle's schedule controls when a statement should be produced. Each statement construct will point to its own statement cycle.

NOT_SUPPORTED

Refer to Statement Construct Maintenance and Statement Cycle for more information.

Create Statements Background Process

This statement construction background process (STMPRD) is responsible for periodically creating statements. It works as follows:

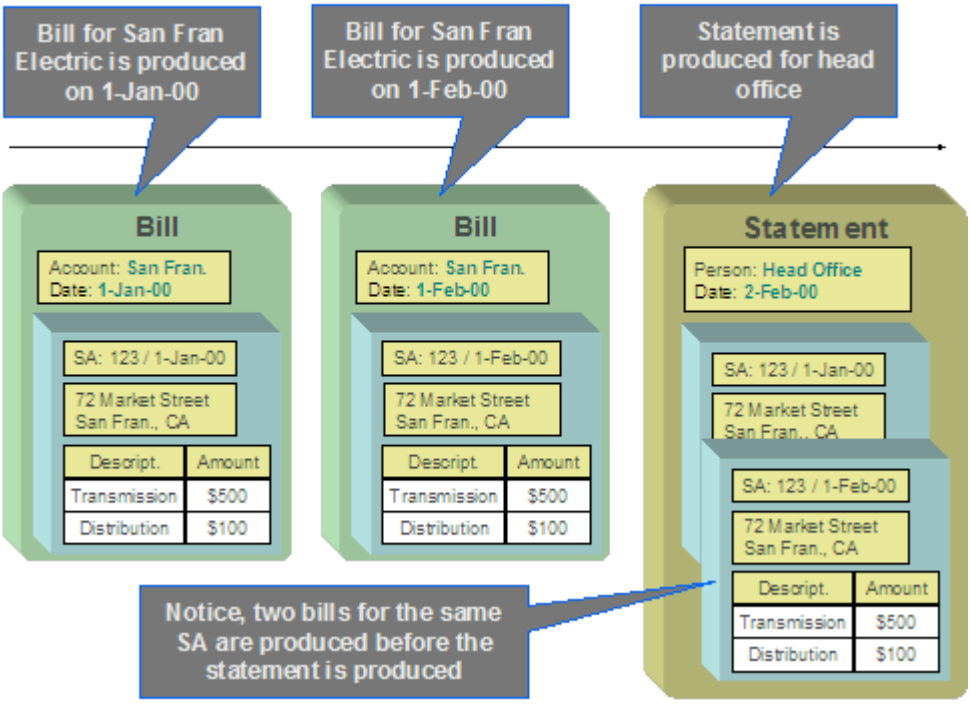
- All statement cycles are selected with pending schedules whose production date is on or before the business date.
- For each such cycle, all of its statement constructs are selected. For each statement construct:
  - If a statement already exists for the cycle schedule and construct, it is deleted (this allows for reproduction after a mess up). Note well, this is true even if the statement is printed as you may have only spotted the problem after viewing the printed statement.
  - A statement is created.
  - All SAs associated with the construct details effective on the business date are extracted.
  - All "SA snapshots" associated with bills whose BILL DATE is within the construct detail's start and end dates that are not already linked to one of the construct's

Copyright © 2026, Oracle and/or its affiliates.

statements are linked to the new statement. Note, an "SA snapshot" is created for every service agreement linked to an account when a bill is completed for an account.

– If no "SA snapshots" are found, the statement is removed.

> **Note**
>
> The following diagram illustrates how the above logic could result in multiple bills for a single service agreement being linked to the same statement.



> **NOT_SUPPORTED**
>
> Refer to Statement Maintenance for more information about a statement and its details.

> **Note**
>
> **This process is rerunnable.** If a problem occurred during the statement creation, simply run the background process again. It will clean up all existing statements for each construct in the current cycle schedule and start again.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Printing Statements for more information for how a separate background process uses the information snapshot on a statement to construct the extract file used to print the statements.

Statement Routing

Just as with bills, statements require routing information in order for the system to understand how and where to send the statements. For example, the system needs to know the method of routing, the number of copies and the format of the statement. This information is captured on a statement construct.

Where Are Statements Sent

The system extracts a statement's address as per its statement construct's address source. Note, if the address source is Person and the person has a seasonal address effective on the business date, the seasonal address will be used.

If you need to reprint a statement because it was addressed incorrectly, correct the person's address, and then reprint the statement (by pressing the **Reprint** button on the statement). When you press this button, the statement will be marked for downloading. When your download process next runs, it simply extracts the statement construct's current address.

On-Line Statement Production

Statements may be produced on-line as well as in batch. To produce a statement on-line, simply go to the Statement page, select the correct Construct ID and choose the Generate button.

This statement will include the financial information related to bills for the accounts and service agreements linked to this construct, with the following criteria:

- The bills do not already appear on a statement
- The bill dates fall within the "effective period" for the statement construct. In other words, the bill falls within the start and end period for the statement construct details.

NOT_SUPPORTED

Refer to Statement Maintenance for more information.

How to Regenerate a Specific Statement

Copyright © 2026, Oracle and/or its affiliates.

There may be times when a specific statement needs to be regenerated. For example, perhaps a cancel / rebill has occurred since the statement was produced, and you wish to recreate the statement to reflect this.

If you need to regenerate a statement using up-to-date information, simply delete the statement and generate a new statement. This may be done on-line.

> **Note**
>
> Statements may be deleted, even if they have already been printed. This enables you to delete and regenerate statements.

> **NOT_SUPPORTED**
>
> Refer to Statement Maintenance for more information.

Printing Statements

The contents of this section describe the technical implementation of printing statements.

- Statement Route Types Control The Information Merged Onto Statements
- Technical Implementation Of Online Statement Display
- Technical Implementation Of Printing Statements In Batch
- Reproducing The Statement Print Flat File
- How To Reprint A Specific Statement
- The Statement Extract Records

Statement Route Types Control The Information Merged Onto Statements

Every statement references a statement route type. The route type controls the following statement print functions:

- It contains an algorithm that is responsible for extracting the information merged onto your statements. Specifically, algorithms of this type create the "flat file records" that are passed to your statement print software. Algorithms of this type are called under the following scenarios:
  - The background process that builds the flat file that's passed to your statement print software calls these algorithms to construct the "flat file records" for each statement.
  - If your statement print software has the ability to construct a real-time image of a statement (in a PDF), you can plug-in an Online Statement Image algorithm on the Installation Record. This type of algorithm will call the statement's dispatch group extract algorithm to extract the information that is merged onto the statement. Refer to Technical Implementation Of Online Statement Display for the details.
- Refer to The Statement Extract Records for examples of the record layouts passed to the document composition application.

Copyright © 2026, Oracle and/or its affiliates.

- It contains the ID of the background process that builds the flat file that's passed to your statement print software. The base package example of this process (STMDWLD) simply calls each statement's construct's route type's extract algorithm to format the information placed onto the flat file.

Technical Implementation Of Online Statement Display

Users can view an image of any statement if you setup the following:

- Plug-in an Online Statement Display algorithm on the Installation Record. Refer to ONSD-ST for an example of such an algorithm. Note, if your statement print software is not capable of producing a PDF containing an image of a statement, users will not be able to view images of statements.
- Plug-in the appropriate extract algorithm on each statement route type. Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to STEX-TX for an example of such an algorithm.

When you plug-in these algorithms, a button appears on Statement - Main. When a user presses this button, the following takes place:

- The installation record's Online Statement Display algorithm is executed.
- This algorithm calls the statement's route type's extract algorithm. This algorithm constructs the information that's merged onto the statement and returns it to the Online Statement Display algorithm. This algorithm, in turn, passes it to your statement print software.
- Your statement print software renders an image of the statement in a PDF and returns it to the Online Statement Display algorithm.
- And finally, the Online Statement Display algorithm displays the PDF in a separate Adobe session.

Technical Implementation Of Printing Statements In Batch

The background process that extracts statement information (STMDWLD) reads all statements in a given run number that are marked with its batch control ID. For each statement, it creates numerous records on a flat file. These records contain the information that is merged onto your statements.

The base package example of this process (STMDWLD) simply calls the extract algorithm on the statement route type to format the information placed onto the flat file. Refer to Statement Route Types Control The Information On Statements for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If your software doesn't support online statement display.** The algorithm that formats statement extract records that's plugged in on statement route type serves two purposes: 1) it formats the records used to construct online images of a statement, and 2) it formats the records downloaded to your statement print software in batch. If your statement print software does not support the rendering of statement images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that both formats the extract records and downloads them (and then specify this batch process on your statement route type).

Reproducing The Statement Print Flat File

You can reproduce the flat file at any time. Simply request the STMDWLD process and specify the run number associated with the historic run.

How To Reprint A Specific Statement

If you need to reprint a specific statement, navigate to Statement - Main and press the **Reprint** button.

If your implementation has enabled the online creation of statement images, you can also press the **Display Statement** button on this page and then print the resultant PDF on your local printer.

The Statement Extract Records

Numerous different types of records are interfaced to the document composition application. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the document composition application statement template.
- The next 186 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed statement appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted statements are printed.
- The remaining bytes contain the **Statement Information** that appears on the printed statement. The type of information differs for each type of record.

The topics in this section describe each component.

- Printer Keys (Record Types)
- Sort Key
- Mailing Key
- Statement Information

Copyright © 2026, Oracle and/or its affiliates.

Printer Keys (Record Types)

The following table defines each printer key (i.e., record type) on the statement print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
0100	Statement record	Required. 1 per statement.
0120	Total by currency record	Required. At least one per statement.
0500	Construct detail record	Required. At least one per construct detail linked to the statement construct.
0900	Service agreement record	1 per SA
1100	Bill record	1 per bill
1300	Financial Transaction (FT) record	1 per FT linked to the SA.
3600	End of record type 1100	1 for each bill
3700	End of record type 0900	1 for each SA
5000	End of record type 0500	1 for each statement construct detail
9999	End of record type 0100	1 for each Statement

Sort Key

The following table defines the structure of the sort key that appears on each statement print record. Please note that different components are filled in for each printer key.

Field Name	Format	Record Type	Source/Value/ Description
Statement ID	A12	All	Prime key of the statement
Sequence Number	N2	All	Set to "1"
Copy Number	N2	All	Defaulted to "1" on the first copy, set to "1" when a statement is requested on-line
Statement record group	A2	0010	Blank
		0100	10
		0120	20
		0500, 0900, 1100, 1300, 3600, 3700, 5000	30
		9999	60
Construct detail sort sequence 1	N3	0010, 0100, 0120, 9999	blank

Copyright © 2026, Oracle and/or its affiliates.

		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's PRT_SEQ
Construct detail sort sequence 2	A10	0010, 0100, 0120, 9999	blank
		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's STM_CNST_DTL_ID
Premise Indicator	A1	0010, 0100, 0120, 0500, 9999	Blank
		5000	'Z'
		0900, 1100, 1300, 3600, 3700	For a non-premise, this will be 'N' otherwise 'Y'
Premise State	A6	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's state (if this is for a non-premise, this will be blank)
Premise City	A90	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700,	Premise's city (if this is for a non-premise, this will be blank)
Premise Address 1	A58	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's address line 1(if this is for a non-premise, this will be blank)
Premise Id	A10	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	Premise's ID (if this is for a non-premise, this will be blank)
SA Print Priority	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	The SA's SA type's print priority
SA Id	A10	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	The SA's SA ID
SA record group	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900	10
		1100, 1300, 3600	20
		3700	30
Bill Date	A10	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill date in the format YYYYMMDD.

Copyright © 2026, Oracle and/or its affiliates.

Bill Id	A12	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill id
Bill record group	A2	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100	10
		1300	20
		3600	30
Arrears Date	A10	0010, 0100, 0120, 0500, 0900, 1100, 3600, 3700, 5000, 9999	Blank
		1300	Financial transaction arrears date
FT Type	A2	0010, 0100, 0120, 0500, 0900, 1100, 3600, 3700, 5000, 9999	Blank
		1300	Financial transaction type

Mailing Key

The following table defines the structure of the mailing key that appears on each statement print record.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on address associated with the statement.

Statement Information

The topics in this section describe the information that appears on each on each statement print record. The following links are provided so that you can move easily to specific topic within this section:

- Global Extract Information Record (0010)
- Statement Record (0100)
- Total By Currency Record (0120)
- Statement Construct Detail Record (0500)
- Service Agreement Record (0900)
- Bill Record (1100)
- Financial Transaction (FT) Record (1300)
- End Bill Record (3600)
- End Service Agreement Record (3700)

Copyright © 2026, Oracle and/or its affiliates.

- End Statement Construct Detail Record (5000)
- End Statement Record (9999)
- Address Sub-record

Global Extract Information Record (0010)

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
STATEMENT_CNT	N10	Count of statements on this extract. Multiple copies of a statement are counted multiple as well.
EXTRACT_DTTM	A31	System time of extraction. Formatted according to user profile.
STMT_ID_START	A12	This is only used when a statement is produced as a result of an online request (from Statement - Main). It contains the Statement Id to extract.
STMT_ID_END	A12	This is only used when a statement is produced as a result of an online request (from Statement - Main). It contains the Statement Id to extract.
THREAD_NBR	N10	This field is only used if statements are extracted in batch. It contains the thread number in which the statement was extracted.
THREAD_CNT	N10	This field is only used if statements are extracted in batch. It contains the total number of threads that were extracted.
STATEMENT_IND	A1	Defaulted to "Y"

Top of Page

Statement Record (0100)

Field Name	Format	Source/Value/Description
STMT_ID	A12	CI_STM
PER_ID	A10	CI_STM_CNST

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
STMT_DT	A31	STM_DT from CI_STM. Formatted according to user profile.
COPY_NBR	N1	"1"
NBR_STMT_COPIES	N1	"1"
ENTITY_NAME	A64	Main name of CI_PER (from CI_PER_NAME).
ADDRESS_SBR	A361	Address where the statement should be sent (the address is defined on the statement's statement construct).
ENTITY_NAME1	A64	CI_PER
ENTITY_NAME2	A64	CI_PER
ENTITY_NAME3	A64	CI_PER
DESCR50	A50	CI_STM_CNST

Top of Page

Total By Currency Record (0120)

One total record will be created for each currency code associated with the FT's associated with the SA snapshots linked to the statement.

Field Name	Format	Source/Value/Description
CURRENCY_CD	A3	This is the currency code associated with the summary information.
CUR_BAL	A30	This is the sum of CUR_AMT snapshot on the SA snapshots (CI_BILL_SA) linked to the statement. SA snapshots (CI_BILL_SAs) are referenced on CI_STM_DTL.
BILL_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "corrections" and "current charges" on bill's associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CUR_ADJ_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "adjustments" on bills associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.
CUR_PAY_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "payments" on bills associated with the statement. Please note, only those FT's associated with SAs linked to the statement will be amalgamated. This is formatted according to user's display profile.

Top of Page

Statement Construct Detail Record (0500)

Field Name	Format	Source/Value/Description
STM_CNST_ID	A12	CI_STM_DTL
SA / Account Indicator	A1	A switch to indicate if the statement construct is for a service agreement or an account. 'S' for service agreement, 'A' for account.
Unique ID	A10	The unique identifier of the account or service agreement
SA / Account Info	A160	Contains the standard information about the account or service agreement
STMT_PRT_DESCR	A50	CI_STM_CNST

Top of Page

Service Agreement Record (0900)

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_SA
Address Information	A120	This is the standard format of a premise address that is displayed throughout the system.
SA Information	A120	This is the standard SA information that is displayed throughout the system

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
SVC_TYPE_CD	A30	CI_SA_TYPE

Top of Page

Bill Record (1100)

Field Name	Format	Source/Value/Description
BILL_ID	A12	
Completion Date	A31	Date formatted according to user profile.
Due Date	A31	Date formatted according to user profile.
SA Ending Balance	A30	
FT Exist Indicator	A1	

Top of Page

Financial Transaction (FT) Record (1300)

Field Name	Format	Source/Value/Description
FT_ID	A12	CI_FT
ARS_DT	A31	CI_FT (The date when the transaction starts aging). Date formatted according to user profile.
AMOUNT	A30	CUR_AMT from CI_FT formatted according to user display profile.
FT_DESCR	A120	Formatted FT information.

Top of Page

End Bill Record (3600)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End Service Agreement Record (3700)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End Statement Construct Detail Record (5000)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End Statement Record (9999)

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Address Sub-record

The address sub-record that in statement record (0100) is composed of the following:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Statement Construct Maintenance

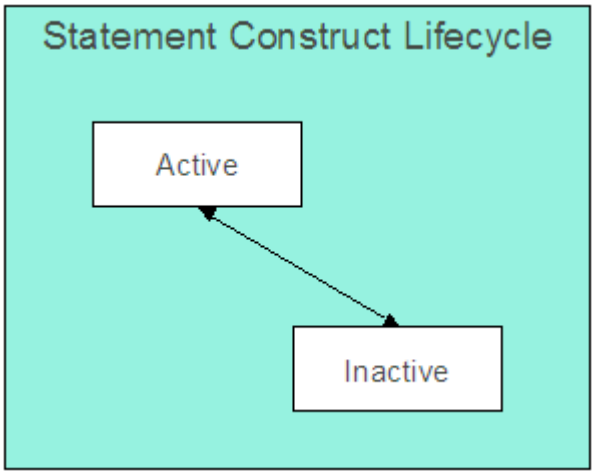
The statement construct enables you to indicate the collection of accounts and/or service agreements whose financial details should be consolidated for a statement to a given person. The statement construct also contains routing information needed by the system to send the statement to the person in the preferred manner.

- Lifecycle of a Statement Construct
- Statement Construct - Main
- Statement Construct - Details

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle of a Statement Construct

The following diagram illustrates the lifecycle of a statement construct.



Active: The statement construct will be in this status while a person is actively receiving statements for the accounts and service agreements linked to this construct. A statement construct may transition from this status to inactive.

Inactive: The statement construct will be in this status when a person no longer wishes to receive statements for the accounts and service agreements linked to this construct. A statement construct may transition from this status back to active.

Statement Construct - Main

Open this page using **Menu**, and then **Customer Information**, and then **Statement Construct**, and then **Search.**

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

Person ID is the person who receives the future statements.

> **Note**
>
> A person may be linked to many statement construct records.

Status indicates whether or not this person is currently receiving statements. The values are Active and Inactive.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Lifecycle of a Statement Construct for more information.

Address Source indicates where the source of the statement's address. The values are Person and Mailing Premise. If the value is person, the person's mailing address is used. If the value is mailing premise, indicate the **Mailing Premise** to use.

Statement Cycle controls when statements are produced for this statement construct. Refer to Statement Cycle for a description of how the production schedule is maintained.

Statement Route Type controls how the statement is routed to the **Person** (e.g., via email, postal service, EDI, etc.). Refer to Statement Route Type **for more information** on setting up statement route types.

Number of Copies indicates how many copies of the statement the person wishes to receive.

Statement Format indicates if the person receives a Detailed or a Summary statement.

Note

The values for this field are customizable using the Lookup table. The values need to match the formats supported by your statement print software. This field name is STM_FORMAT_FLG.

Print Description is a brief description that can be printed on the statement.

The tree shows the statement construct's details. The nodes expand to show account, premise and service agreement information for the construct details.

Note

Large statement constructs. The tree is hard to use if the statement construct has many details. In this situation, transfer to the adjacent tab page and use the filter to restrict the details that are shown.

Statement Construct - Details

This page is used to maintain the accounts and service agreements where financial transactions should appear on statements produced using this statement construct. Open this page using **Menu**, and then **Customer Information**, and then **Statement Construct**, and then **Search** and navigate to the **Details** page.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

Copyright © 2026, Oracle and/or its affiliates.

If a statement construct has a large number of details, you can use the **Details Filter** to limit the details that appear in the grid. The following options are available:

- All. Use this option to view all details linked to the statement construct.
- Person Name. Use this option to restrict details linked to accounts whose main customer has a primary name that matches an input **Name**.
- SAs at an Address. Use this option to only show details linked to service agreements that are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- Statement Print Description. Use this option to restrict details to those with a given **Description**.

Don't forget to click the search button after changing the filter.

The grid contains the Accounts and/or Service Agreements whose financial information contributes to this statement construct. Each record in the grid contains the following information:

- **Construct Detail Type** defines if the construct detail is for an entire Account or a single Service Agreement. This field is protected if a statement detail was produced using this construct.
- **Service Agreement/Account** identifies the account (for Account construct details) or the service agreement (for Service Agreement) construct details. This field is protected if a statement detail was produced using this construct.
- **Statement Print Description** is available for use by the extract program to include on the printed statements to help the recipient identify the details.
- **Print Order** control where the financial transactions associated with the account / service agreement appear on the printed statement.
- **Start Date** is the date that this account / service agreement's financial transactions should begin appearing on statements.
- **End Date** is the date that this account / service agreement's financial transactions should stop appearing on statements.

> **Note**
>
> Only financial transactions linked to bills whose BILL DATE is within the construct detail's **Start Date** and **End Dates** will appear on statements.

- **Construct Detail ID** is a system generated unique identifier of the construct detail.

Statement Maintenance

Statements allow you to set up a Person to receive a consolidated report of the financial activity for one or more accounts and/or service agreements. The topics in this section describe how to maintain statements.

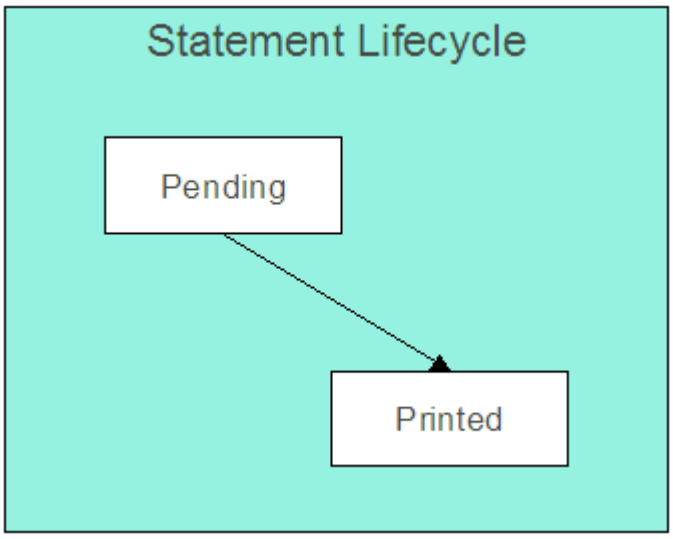
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The system creates most statements behind-the-scenes.** Most statements are created by the system when it processes the statement cycle schedule. You should only have to access the statement pages to regenerate a statement or to add a statement real-time. For information about how the system creates statements, refer to The Statement Creation Background Process.

- Lifecycle of a Statement
- Statement - Main
- Statement - Details

Lifecycle of a Statement

The following diagram illustrates the lifecycle of a statement.



Pending: A statement is created in this state and will remain in this state until the statement details are extracted. When that occurs, it will transition to printed.

Printed: A statement transitions to this state after the details of a statement have been extracted.

Statement - Main

Open this page using **Menu**, and then **Financial**, and then **Statement**, and then **Search**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Correcting erroneous statements.** It's important to be aware that there are very few fields that are directly modifiable by a user. This is because a statement is a summary of financial transactions. If you need to change information on a statement, you must change the source information (e.g., change the statement's construct or complete additional bills) and then regenerate the statement. Refer to How To Regenerate A Statement for more information.

Description of Page

Statement is a concatenation of summary information about this record. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Statement ID is the unique system-generated identifier for this statement.

Indicate the **Construct ID** for the statement construct whose construct details should be included in this statement.

Address is a display-only field that contains the address to which the statement will be sent.

The **Statement Status** indicates the current status of the statement. The values are Pending, Printed.

> **NOT_SUPPORTED**
>
> Refer to Lifecycle of a Statement for more information.

The **Create Date** indicates the date that this statement was created.

The **Batch Control** and **Batch Number** are the batch process and run in which the statement was (or will be) sent to the recipient.

> **NOT_SUPPORTED**
>
> Refer to Printing Statements and to Statement Route Type for more information.

The **Statement Action** area contains buttons that you use to generate, reprint or delete a statement.

- The **Generate** button enables you to create a statement on-line. Refer to On-line Statement Production for more information.
- The **Reprint** button will stamp the latest run number for the statement's batch control. This will enable the statement to be reprinted the next time the batch control process executes. Refer to How to Reprint A Specific Statement for more information.
- The **Delete** button will remove the statement and its details from the database. Refer to How to Regenerate a Specific Statement for more information.
- The **Display** button will display an online image of the statement when pressed. Refer to Online Statement Display for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can only use the **Display** button if your system has been configured to display an on-line image; otherwise, a message indicating that the service is not available will appear. This option can only be configured by your technical staff. Refer to Technical Implementation Of Online Statement Display for more information.

The bottom portion of this page is dedicated to a tree that shows the statement details. The nodes expand to show every service agreement that will appear on the statement and the financial details that will be swept onto the statement for each service agreement. You may transfer to the details of any of the nodes by selecting that node. Context menus are also available for various nodes to facilitate easy navigation to other pages.

> **Note**
>
> **Large statements.** The tree will be hard to use if the statement has many service agreements that contributed financial transactions to the statement. In this situation, transfer to the adjacent tab page and use the filter to restrict the service agreements that are shown.

Statement - Details

This page is used to view the service agreements that contributed financial transactions to the statement. Open this page using **Menu**, and then **Financial**, and then **Statement**, and then **Search** and navigate to the **Details** page.

Description of Page

Statement is a concatenation of summary information about the statement. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Statement ID is this statement's unique system-generated identifier.

If a statement has a large number of service agreements that contributed financial transactions to the statement, you can use the **SA Filter** to limit the service agreements that appear in the grid. The following options are available:

- All. Use this option to view all service agreements that contributed financial transactions to the statement.
- Person Name. Use this option to only show service agreement linked to accounts whose main customer has a primary name that matches **Person Name**.
- SAs at an Address. Use this option to only show service agreements that are linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
- Statement Print Description. Use this option to only show service agreements associated with a construct detail with a given **Statement Print Description**.

Don't forget to click the search button after changing the filter.

Copyright © 2026, Oracle and/or its affiliates.

The grid contains service agreements that contributed financial transactions to the statement. If a service agreement's financial transactions appeared on multiple bills, a separate line is displayed for each bill. You can press the go to button adjacent to **Bill Due Date** to be transferred to Financial Transactions On A Bill where the individual financial transactions associated with the associated service agreement and bill can be viewed. The following information is displayed in the grid:

- **Statement Print Description** is the description of the construct detail associated with the service agreement.
- **Bill's Due Date** is the date of the bill on which the service agreement's financial transactions appear. Press the adjacent go to button to be transferred to Financial Transactions On A Bill where the individual financial transactions can be viewed.
- **Premise Information** describes the characteristic premise, if any, associated with the service agreement.
- **SA Information** describes the service agreement that contributed financial transactions to the statement.
- **Account Information** describes the account associated with the service agreement whose details appear on the statement.
- **Construct Detail ID** is the unique identifier of the construct detail associated with the service agreement.
- **Statement Detail ID** is the unique identifier of the statement detail on which the service agreement's financial transactions appear.

Sales & Marketing

The sales and marketing functionality satisfies many diverse requirements. For example, you can use this functionality to:

- Enroll new customers using a single transaction (i.e., you don't have to use the person, premise, service point, and start/stop transactions to enroll a new customer who resides at a new premise).
- Sell new products to existing customers.
- Update person, account and premise information using a single transaction.
- Market your services to prospects from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer, premise and related service agreements.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Setup proposals for prospective services (and then send a quotation to the customer for these services).
- And more...

The topics in this section provide more information about the sales and marketing functions.

- The Big Picture of Campaigns, Packages, and Orders
- The Big Picture Of Package Eligibility Rules
- The Big Picture Of Campaign Eligibility Rules

Copyright © 2026, Oracle and/or its affiliates.

- Designing Campaigns and Packages
- Other Useful Information
- Maintaining Orders
- Maintaining Campaigns
- Maintaining Packages
- The Big Picture of Program Management

The Big Picture of Campaigns, Packages, and Orders

Three objects are at the heart of the sales and marketing functionality: **Campaigns**, **Packages**, and **Orders**.

- A **package** defines a basket of goods and services that can be offered to a customer or a prospect.
- A **campaign** is a structured effort to offer a given set of packages to existing customers or new prospects.
- An **order** is created for each customer / prospect to which a campaign is targeted.

The topics in this section provide background information about these objects.

- An Overview of Campaigns, Packages and Orders
- Supported Business Processes
- Examples of Campaigns and Packages

An Overview of Campaigns, Packages and Orders

An order is used to define:

- Demographic information about a customer / prospect.
- Geographic information about the service address.
- The customer's response to eligibility-oriented questions. For example, you can pose questions like: Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth?

After the above information is defined, the order transaction presents packages that may be offered to the customer. A package controls the various types of service agreements that will be created if the customer selects the package.

> **Note**
>
> **The customer must be eligible for a package.** When you setup a package, you define its eligibility criteria. For example, you can setup a package that is only applicable to commercial customers in Toronto who pay automatically and who commit to a one-year service contract.

Copyright © 2026, Oracle and/or its affiliates.

If the customer elects to take a package, the order transaction sets up / updates the "V" (along with all of the ancillary things that happen when service is initiated, e.g., field activities are created). Please note that in addition to setting up the "V", it's also possible to populate / update other information when a package is selected. For example, you could have the system setup the customer's automatic payment options.

> **Note**
>
> **An order can be completed without creating service agreements.** It is possible to use the order transaction to simply create / update persons and accounts. Refer to Marketing Surveys and Setting up a New Customer Prior To Using Start/Stop for more information.

All orders must reference a campaign. An order's campaign defines:

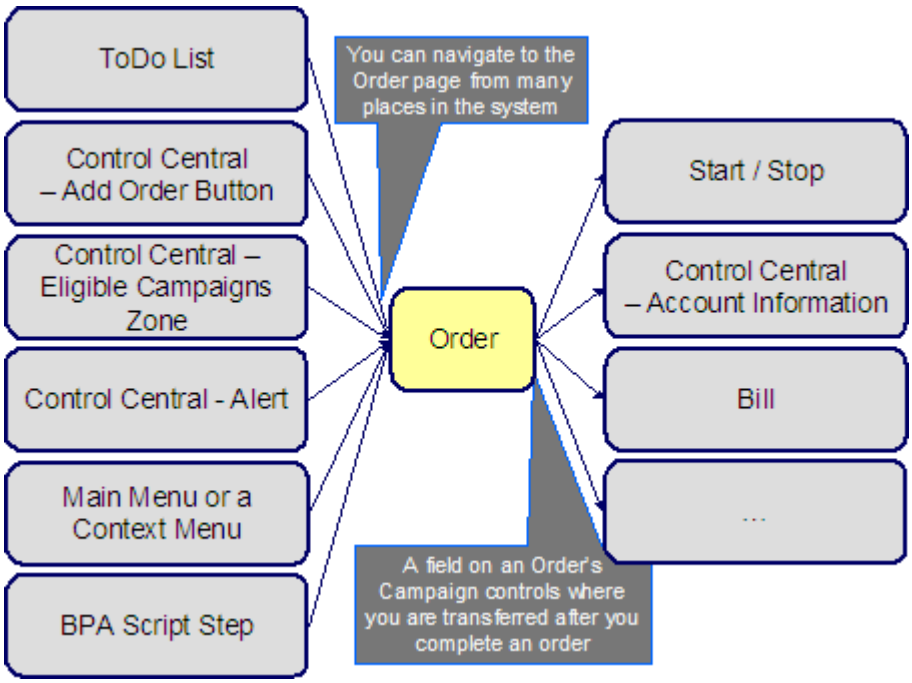
- How the campaign's orders use accounts and premises. For example, you can indicate that a given campaign is only targeted at existing premises (thus preventing the creation of a new premise when an order is completed).
- The type of information defaulted onto an order. For example, you can setup a campaign to default a given account management group on all orders linked to the campaign. This account management group will subsequently default onto the new account when the order is completed.
- The validation rules that control how its orders use accounts and premises. Specifically, on a campaign you define if an account / premise is required / optional / not allowed on its orders. In addition, if an account / premise is required or optional, you can control whether new accounts / premises can be created when an order is completed (the alternative is to force each order to use an existing account / premise). These controls prevent the unwanted proliferation of new accounts and premises for campaigns that are targeted at existing accounts and premises.
- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above - Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth? - are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level eligibility rules and business process flows.

> **Note**
>
> **Bottom line. Campaigns** allow you to define a group of **packages** that a customer is eligible to purchase. Every time a campaign is targeted at a customer, an **Order** is created. The information entered on the order qualifies the customer for one of the campaign's packages. If a customer elects to take a package, the "V" is setup / updated along with all of the ancillary things that happen when service is initiated, e.g., field activities are created, workflow processes are initiated, etc.

Copyright © 2026, Oracle and/or its affiliates.

Supported Business Processes

Before providing examples of specific business processes, we'd like to highlight that you have control over some aspects of the user-interface flow (i.e., the screen navigation is not hard-coded). The following illustration highlights the potential user-interface flow:



> **Note**
>
> **Bottom line.** You can navigate to the order transaction from many places in the system. After you complete an order, you are transferred to an appropriate transaction. The specific transaction is controlled by you when you setup the order's campaign.

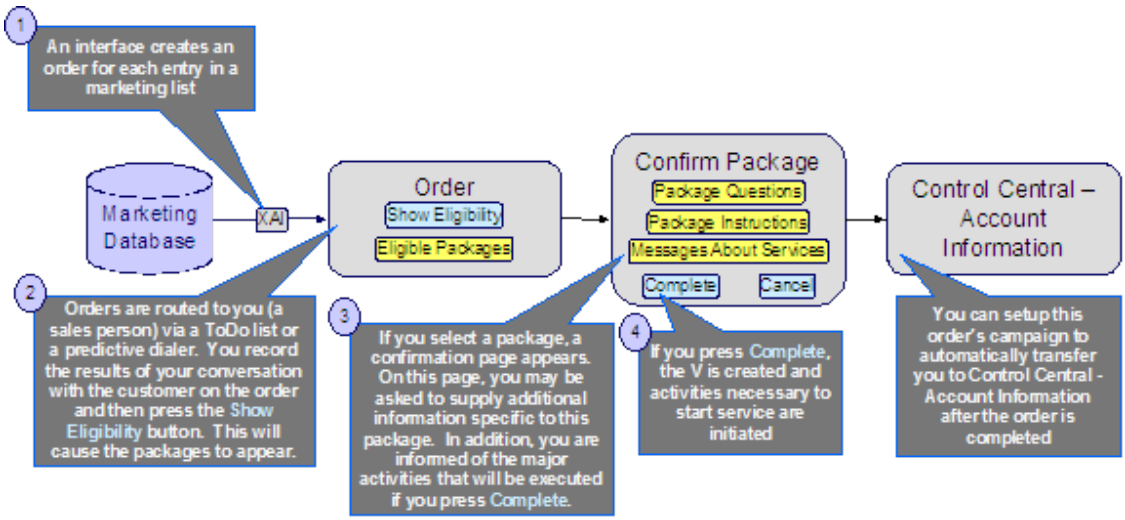
The topics that follow illustrate specific business process flows that are possible in your implementation of the sales and marketing functionality.

- Marketing to Prospects from a Marketing List
- Marketing to Pre-Selected Customers
- Real-Time Marketing of Additional Services to a Customer
- Real-Time Marketing of Services to a Prospect
- Setting up a New Customer Prior To Using Start/Stop

Marketing to Prospects from a Marketing List

The following business process flow illustrates how the sales and marketing functionality would be used to market to prospects uploaded from a marketing database.

Copyright © 2026, Oracle and/or its affiliates.



The following points describe the steps in this business process:

- An upload interface creates an order for each entry in a marketing database.

> **Note**
>
> The system is not supplied with a dedicated interface to upload a marketing list.

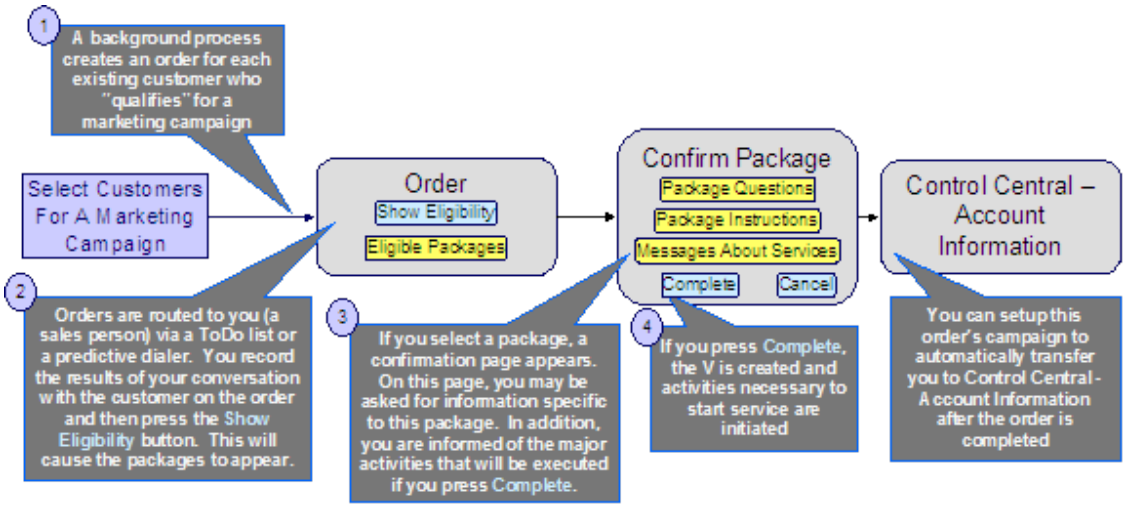
- Each order is routed to a salesperson via a To Do entry (or via a predictive dialer if your company supports this type of technology).
- The first step in the salesperson's interaction with the prospect will be to confirm the information uploaded from the marketing list (e.g., is your Bob Smith?, is your address 15 Main St?, etc.). Any changes are made to the information saved on the order.
- The salesperson's next step involves posing eligibility-oriented questions related to the campaign. Examples of questions include:
  - **What is your date of birth?** This type of question might be asked if you have packages that are only marketed to senior citizens. Potential answers would be a valid date that's in the past.
  - **Would you be interested in our full-service option (electricity, digital cable and gas)?** Potential options are: yes and no.
  - **Who is your current electric service provider?** Potential answers are: Utility.Com, Reliant Resources, Other.
  - **Would you like to pay automatically?** Potential answers are: Yes, No.
  - **Would you like to pay using a levelized payment plan?** Potential answers are: Yes, No.
  - Etc.
- The gathered information is saved on the order and then the system displays the packages that can be offered to the customer. The available packages may be restricted based on information on the order. For example, some packages are only applicable to senior citizens, while others are only applicable to customers whose current service provider is Utility.com.

Copyright © 2026, Oracle and/or its affiliates.

- If the customer is interested in a package, the salesperson selects it. Prior to the order being completed, the package confirmation page is shown. On this page are additional instructions about the package and a description of how the system will be updated if the package is selected (e.g., the types of service agreements will be displayed). In addition, the package may require additional information before it can be chosen. For example, the package may only be selectable if the customer agrees to pay automatically. In this situation, the package confirmation page will require the salesperson to enter the customer's bank and bank account.
- After entering all package-specific information, the order is completed. When the order is completed, the "V" is setup and all work necessary to satisfy the order is initiated (e.g., field activities are created, workflow processes are initiated, etc.).
- And finally, the user is returned to Control Central - Account Info where an overview of the customer is displayed. From this page, the user can drill down to any service agreement (or to start/stop) to perform any fine-tuning.
- Please be aware that the page to which you are transferred after order completion is controlled by a field on the order's campaign. In our example, it made sense to transfer the user to Control Central as it provides a nice confirmation to the user (it shows a great deal of information about an account). However, you could setup the campaign to take the user to virtually any page in the system. You'll see good examples later in this discussion.

Marketing to Pre-Selected Customers

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers selected by a background process.



The following points describe the steps in this business process:

- A background process creates an order for each customer to be targeted under a marketing campaign.

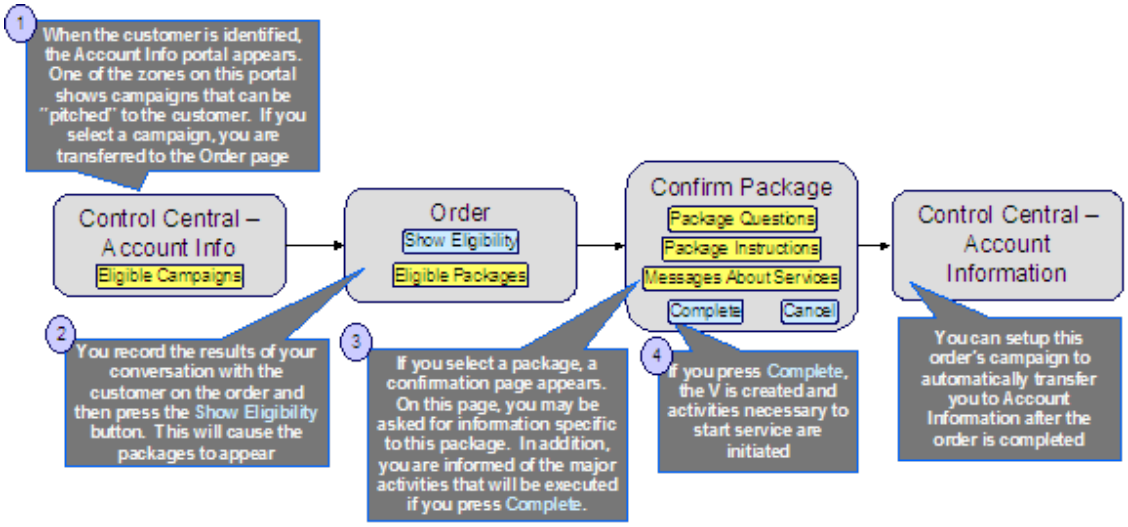
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The base package does not have an example of such a background process. Please speak to your support team if you require assistance in developing this type of process.

- Sales people would then process the order as described above.

Real-Time Marketing of Additional Services to a Customer

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers when they call in.



The following points describe the steps in this business process:

- When you find the customer's record on Control Central - Search, you are automatically transferred to the Account Information portal. One of the zones on this page contains the campaigns that can be offered to the customer.

> **Note**
>
> The "potential campaigns" zone only appears if the CSR has modified their preferences to display this zone.

- If you select a campaign from this zone, you are transferred to the Order transaction.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> An order's campaign controls whether the Main or Questions & Misc Fields tab is initially displayed when the order transaction is invoked from this zone. For example, you might want to have the **Main** tab displayed if you want the order-taker to confirm the customer's demographic information before posing questions. Alternatively, you may want to skip this step and immediately present the order-taker with the campaign's questions. Keep in mind that the user can always display the other tab regardless of what is initially displayed.

- You then process the order as described above.

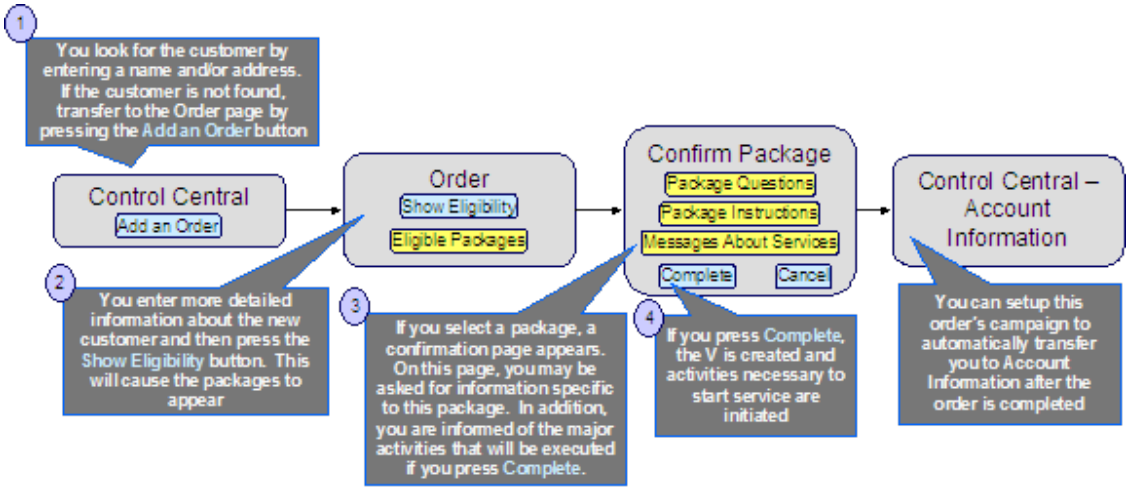
> **Note**
>
> **An easy way to create one-time charges.** The above business process illustrates how the selection of a package will create one or more new service agreements for a customer. If your organization levies one-time charges (e.g., tree trimming charges, damage assessments, etc.), you can also use the order transaction to create both a "one-time charge" service agreement and create the related billable charge when the order is completed. To do this, setup a campaign with a separate package for each possible one-time charge. If an order is created for this type of campaign and the relevant package is selected, the system will setup the new billable charge service agreement and link the billable charge to it. Refer to An Easier Way To Create One Time Charges for an example of such a campaign.

Real-Time Marketing of Services to a Prospect

The following business process flow illustrates how the sales and marketing functionality would be used to market services when a new customer calls.

> **Warning**
>
> We do not recommend using this business process if your organization's service territory is predefined (i.e., if you've set up the premises and service points for your entire service territory). Why? Because it can result in the creation of new premises and / or service points and you probably don't want this to happen if you've already set everything up. While it is possible to disable the creation of new service points and premises by developing new plug-ins, we'd recommend using the business process described under How To Add A New Customer From Control Central if a new customer calls in and you have predefined premises and service points.

Copyright © 2026, Oracle and/or its affiliates.



The following points describe the steps in this business process:

- You look for the customer using Control Central - Search. Because the customer is new, you push the Add Order button on Control Central to transfer to the Order transaction.
- You then process the order as described above.

In order to take advantage of the above business process, it's important that you understand the following points:

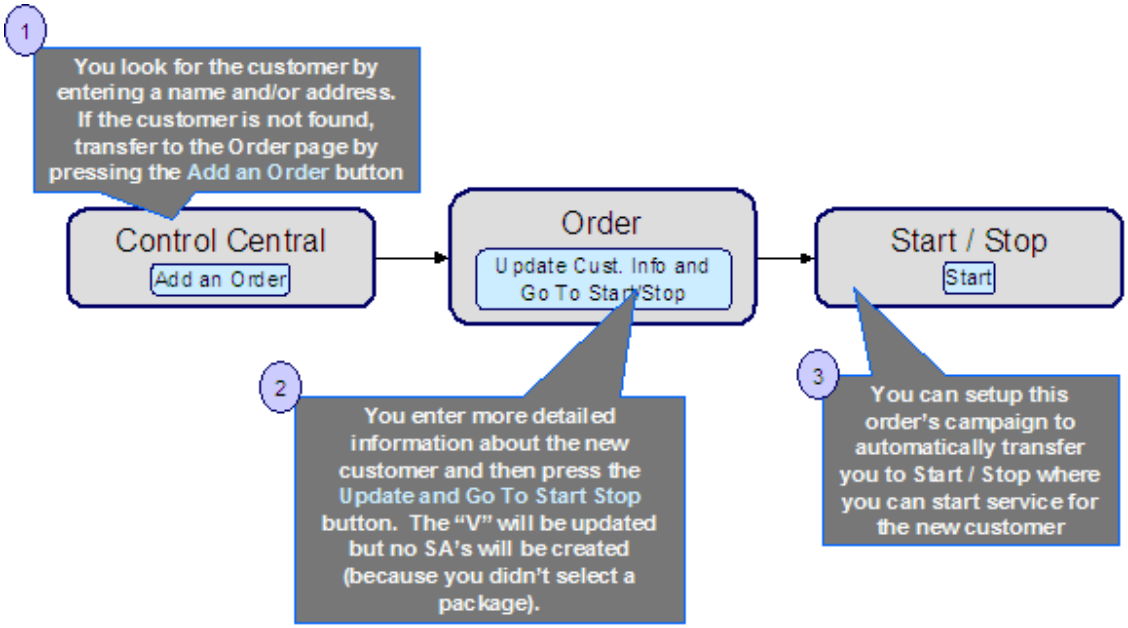
- When the order page is opened for a new customer, the order's campaign is defaulted from the installation record. The default campaign on the installation record is "generic" because we don't know anything about the customer (they are new) and therefore we cannot default a campaign targeted to their customer class or geography.
- This default campaign on the installation record must be constructed using one of the following approaches:
  - The default campaign has a package for every potential type of customer who can call up. While this is possible, it's probably not realistic as the data setup burden required to define every type of package could be onerous. However, if you have a limited number of service agreements, this approach is feasible.
  - The default campaign could just be a "dummy" campaign that has no packages. The "real" campaigns will only be shown after basic information about the customer has been recorded on the order and the "show eligibility" button is pressed. These campaigns appear in the order's eligibility tree. The selection of a campaign in this tree will cause the order's campaign to be switched. Note, because the "dummy" campaign has no packages, no packages will appear in the order's eligibility tree when the dummy campaign is referenced on an order.
- The order is then processed as described above.

Setting up a New Customer Prior To Using Start/Stop

The following business process flow illustrates how the sales and marketing functionality would be used to create a new customer prior to using Start/Stop to start service at an existing premise.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The following business process is similar to that described under How To Add A New Customer From Control Central. The only difference is that the Order transaction is used to create a new person and account rather than the Person transaction. The order transaction may be a better choice because it lets you define many attributes on person, account and premise that are not accessible on the person page. For example, the order transaction lets you define the customer's automatic payment option, bill after date, override due date, etc. In addition, on an order's campaign, you can define default values of many field values (both real fields and characteristics). These default values are presented on the order transaction and may be overridden by the user at will.



The following points describe the steps in this business process:

- You look for the customer using Control Central - Search. Because the customer is new, you push the "Add Order" button on Control Central to transfer to the Order transaction.
- You record additional demographic and geographic information about the new customer and confirm all values defaulted from the order's campaign. After all required information is entered, you press the Update Cust. Info and Go To Start/Stop button to create the new person and account and transfer to Start/Stop to identify the new services desired by the customer.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The button's existence and label are dynamic.** In the above example, we indicated that there is a button on the order page called **Update Cust. Info and Go To Start/ Stop**. This button's presence on the page and its label are controlled by fields on the campaign. When pressed, the "V" is updated with information on the order and the user is transferred to the "post completion" page (which is also defined on the campaign). You would only enable this button on campaigns whose orders can be completed without selecting a package. In the above example, the campaign doesn't have any packages (because service is intended to be started using Start/Stop). In Examples of Campaigns and Packages we show other examples of how these fields would be setup for different types of campaigns.

In order to take advantage of the above business process, it's important that you understand the following points:

- When the order page is opened as a result of pressing the Add Order button on Control Central - Search, the order's campaign is defaulted from the installation record. The default campaign on the installation record is "generic" because nothing is known about the new customer and therefore it is impossible to default a campaign targeted to their customer class or geography.
- The campaign on the installation record should be setup to contain default information for the "average" customer. For example, if you typically deal with residential customers, the information on the default campaign should be residential-oriented.
- When a user takes an order for a new customer, they'll change the default information when appropriate. For example, if an industrial customer were being added, they'd change the customer class to "industrial" and then ask to see the campaigns such a customer is eligible to choose (these alternate campaigns are displayed in the order's eligibility tree). For example, if you've setup campaigns with eligibility rules that are targeted at industrial customers and the customer class on the order has been changed to "industrial", the user will see these campaigns in the eligibility tree. The user can then change the campaign's order by simply clicking on the desired campaign.

> **Note**
>
> The above example illustrates how an order can be completed without causing the creation of service agreements. Refer to Marketing Surveys and Prelude to Start Stop for examples of such campaigns.

Examples of Campaigns and Packages

The topics that follow provide examples of campaigns and packages that support several different scenarios. Your organization's campaigns and packages may differ markedly from these examples. Factors that will influence your sales and marketing objects include:

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,

Copyright © 2026, Oracle and/or its affiliates.

  - Your company may have very straightforward packages that only differ based on whether the customer is industrial, residential or commercial. If this describes your company, the setup process will be simple.
  - Alternatively, your company might offer different packages to residential customers based on whether they pay automatically, whether the customer is a senior citizen, whether they purchase multiple services from you, etc. If this describes your company, then you'll have to define the various eligibility criteria and the respective packages applicable to customers who satisfy each criterion.
- Whether premises exist prior to order taking. For example, a company that distributes a commodity has a well-defined geographic area and therefore new premises typically don't arise when orders are taken.
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.
- Which of the Supported Business Processes you implement.

Use the information that follows to form an intuitive understanding of campaigns and packages. After attaining this understanding, you'll be ready to design your own campaigns and packages.

- Campaigns Without Packages
- Campaigns With Simple Packages
- Campaigns With Complex Packages

Campaigns Without Packages

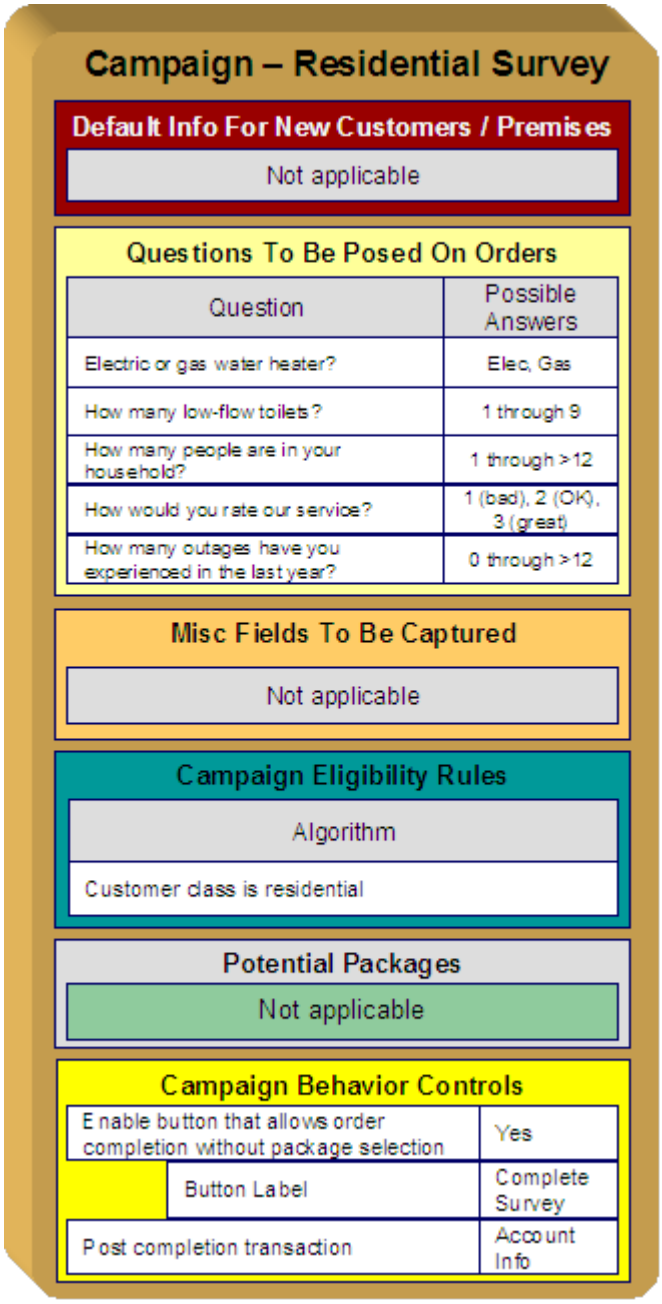
An order that references a campaign without packages will never create service agreements. The following topics describe scenarios when such a campaign would be used.

- Marketing Surveys
- Prelude to Start/Stop

Marketing Surveys

Some organizations conduct "marketing surveys" to determine general trends amongst their customer base. The following diagram illustrates how such a campaign might look:

Copyright © 2026, Oracle and/or its affiliates.



Note the following about this campaign:

- If the campaign is used to establish new customers, you can define information to default onto orders created for new customers. Because marketing surveys are targeted at existing customers, you don't need new customer default information on the campaign.
- You can define questions to be posed to customers when an order is taken when you setup a campaign. For a "marketing survey" campaign, you'd define each of the survey questions and the permissible answers. When a customer is surveyed, an order is created and their answers are saved on the order.
- On a campaign, you can define additional fields to be updated on persons, accounts and premises when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. Because marketing surveys are typically only asking

Copyright © 2026, Oracle and/or its affiliates.

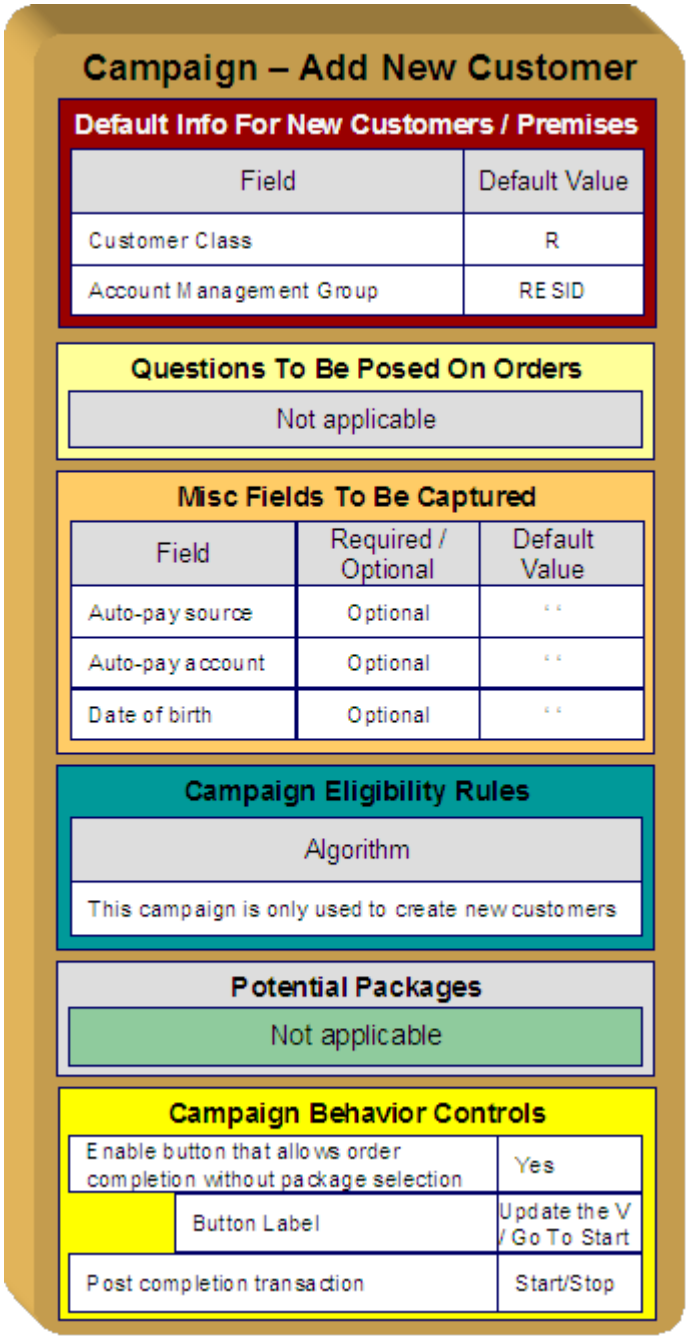
questions, you wouldn't need to define additional customer information fields to be confirmed. However, you can if you want.

- All campaigns need at least one eligibility rule. These rules control whether the campaign appears in the "eligible campaigns" content zone when a customer is selected on Control Central. These rules also control whether the campaign appears on an order's eligibility tree (campaigns that appear in this tree are alternate campaigns that may be used on the order). Our example has assumed that this is a residential customer survey and therefore it only needs a single eligibility rule. Refer to The Big Picture Of Campaign Eligibility Rules for more information.
- Because this campaign is used purely to manage a marketing survey, it doesn't need any packages. Packages are only necessary if service agreements are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows orders to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete a marketing survey (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Complete Survey.
- This campaign should be setup to transfer the user to Control Central - Account Info after the survey is completed.

Prelude to Start/Stop

Consider the example illustrated above under Setting up a New Customer Prior To Start/Stop. In this example, the order transaction is used to create a new person and account; no service agreements are created when the order is completed. Rather, after the order is completed, the user is automatically transferred to Start/Stop where they can create service agreements. The following diagram illustrates how such a campaign might look:

Copyright © 2026, Oracle and/or its affiliates.



Note the following about this campaign:

- Because this campaign is used to establish new customers, you should define new customer default information. The order taker can override these default values.
- If we assume you don't ask marketing / eligibility-oriented questions when you take-on a new customer then this campaign would not require any questions.
- On a campaign, you can define additional fields to be updated on the related person, account and premise when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. In our example, we are assuming that we want to define the customer's automatic payment options and their date of birth.

Copyright © 2026, Oracle and/or its affiliates.

- As described under Marketing Surveys, a campaign's eligibility rule(s) control whether the campaign appears in the "eligible campaigns" content zone and on an order's eligibility tree. This campaign is odd in that we don't want it to ever appear as an "eligible campaign" because it's only used for NEW customers. In order to make this work you must:
  - Define an eligibility algorithm on this campaign that prevents it from appearing as an "eligible campaigns". Refer to A Campaign That Is Never Eligible for more information.
  - Update the installation record to indicate that this campaign is the one that should be used on orders created when a user presses the "Add an Order" button on Control Central - Search. Refer to Setting up a New Customer Prior To Starting Service for more information about the flow of this business process.
- Because this campaign is used purely to add accounts and persons, it doesn't need any packages. Packages are only necessary if service agreements are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows an order to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete the order (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Update The V and Go To Start.
- This campaign should be setup to transfer the user to Start/Stop after the order is completed.

Campaigns With Simple Packages

As explained under Campaigns Without Packages, there are several situations that might necessitate a campaign without packages. In this section, we describe simple scenarios of campaigns with packages.

- An Easier Way To Create One Time Charges

An Easier Way To Create One Time Charges

We use the term "one-time charge" to describe ad hoc charges, e.g., tree trimming charges, damage assessments, etc. The following points describe how one-time charges are levied if you don't use the order transaction:

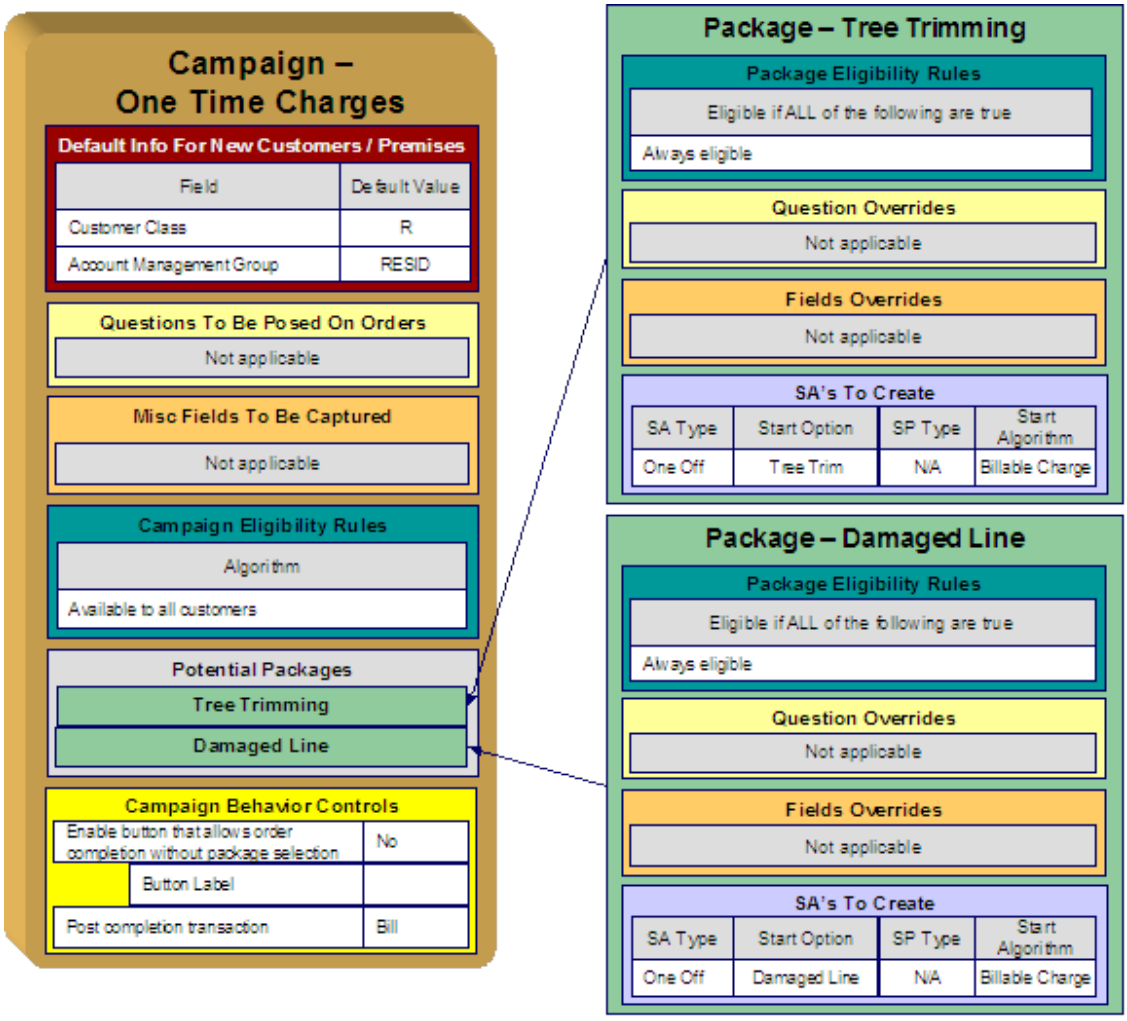
- Determine if the customer to be invoiced already has an appropriate "one-time charge" service agreement. If one doesn't exist, you must create one using Start/Stop Service.
- Next, a billable charge is added to the "one-time charge" service agreement (using Billable Charge Maintenance). The billable charge contains the invoice lines and amounts. The system will produce the bill for this charge when the account's bill cycle next executes.

Rather than going through the above multi-step process, you can also use the order transaction to create both a "one-time charge" service agreement AND create the billable charge. In addition, if the customer receiving the charge is new, the order transaction will also create a new person and account at the same time.

To take advantage of this facility, you must set up a campaign that has a separate package for each type of one-time charge as illustrated below. When an order is taken for such a campaign, the user simply selects the package that corresponds with the desired one-time charge. The system will then create the necessary customer and financial information when the package is confirmed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Billing of one-time charges.** The order transaction will not cause a bill to be generated for the customer. Rather, the next time the account is billed, the system will see that there's an unbilled billable charge and create a bill segment for it. This bill segment will appear on the customer's regular bill along with their other charges. If you'd prefer to create the bill for the one-time charges immediately, you could setup the campaign to automatically transfer the user to the bill transaction where they can create an ad hoc bill. If you do this, you'll need to create a new plug-in algorithm on the billable charge SA type to automatically activate the service agreement when it is created (otherwise the user would have to manually activate the billable charge service agreement before the bill can be generated).



Note the following about this campaign:

- One-time charges can be levied against both new and existing customers. This means that orders associated with this campaign may cause new customers to be created. Therefore, we have specified information to default onto orders for new customers on the campaign. Note that we defined "residential-oriented" default values as the majority of one-time charges are levied against residential customers.

Copyright © 2026, Oracle and/or its affiliates.

- If we assume you don't ask marketing / eligibility-oriented questions when you levy a one-time charge then this campaign would not require any questions.
- We have assumed that you are not capturing miscellaneous fields when you levy a one-time charge.
- As described under Marketing Surveys, all campaigns need at least one eligibility rule. We have assumed that this campaign's packages can be selected by all types of customers and have therefore specified an eligibility algorithm that will be true for all customers.
- Because this campaign has packages, we have indicated that the button that allows an order to be completed without selecting a package is disabled. This means a user must choose one of the packages to complete the order.
- You might want to setup this campaign to automatically transfer the user to the bill transaction after the order is completed. See the note above the illustration for more information.
- We have assumed that this campaign can be used to levy two kinds of one-time charges - a tree trimming charge and a damaged line charge - and therefore this campaign requires two packages. Note the following about each package:
  - A package's eligibility rules control whether the package can be selected when an order is taken. With more sophisticated campaigns, you may have packages that are only applicable to certain types of customers. In our example, we've assumed that these types of one-time charges can be levied against all customers and have therefore used "universal" eligibility criteria (i.e., there are no eligibility restrictions). Refer to A Package That Is Always Eligible for more information.
  - On a package, it's possible to define additional questions to be posed if a package is selected. In our example, there are no additional questions to be posed for the one-time charge packages.
  - On a package, it's possible to define additional fields to be captured if a package is selected. In our example, there are no additional fields to be captured.
  - On a package, you define the types of service agreements to be created if a package is selected. In our example, we only need a single service agreement created / updated if the package is selected and therefore there's just one entry in **SAs To Create**. Please note the following about the information used to create this service agreement:
- The **SA Type** is one that's used for a billable charge service agreement.
- The SA Type's **Start Option** is one that will automatically create a billable charge when a service agreement is created (if a start option references a billable charge template, a billable charge will be created when this start option is used). In our example, we have assumed a different start option has been setup for each type of one-time charge.
- Billable charge service agreements do not require service points and therefore **SP Type** is not applicable.
- We've indicated a **Start Algorithm** of Billable Charge . Refer to STRM-AS for an example of such an algorithm (type).

Campaigns With Complex Packages

The Campaigns With Simple Packages illustrated campaigns with very simple packages. In this section, we describe a campaign with more complex packages.

- Packages Limited By Answers And Field Values

Copyright © 2026, Oracle and/or its affiliates.

- Packages Limited By Current State Of Service

Packages Limited By Answers And Field Values

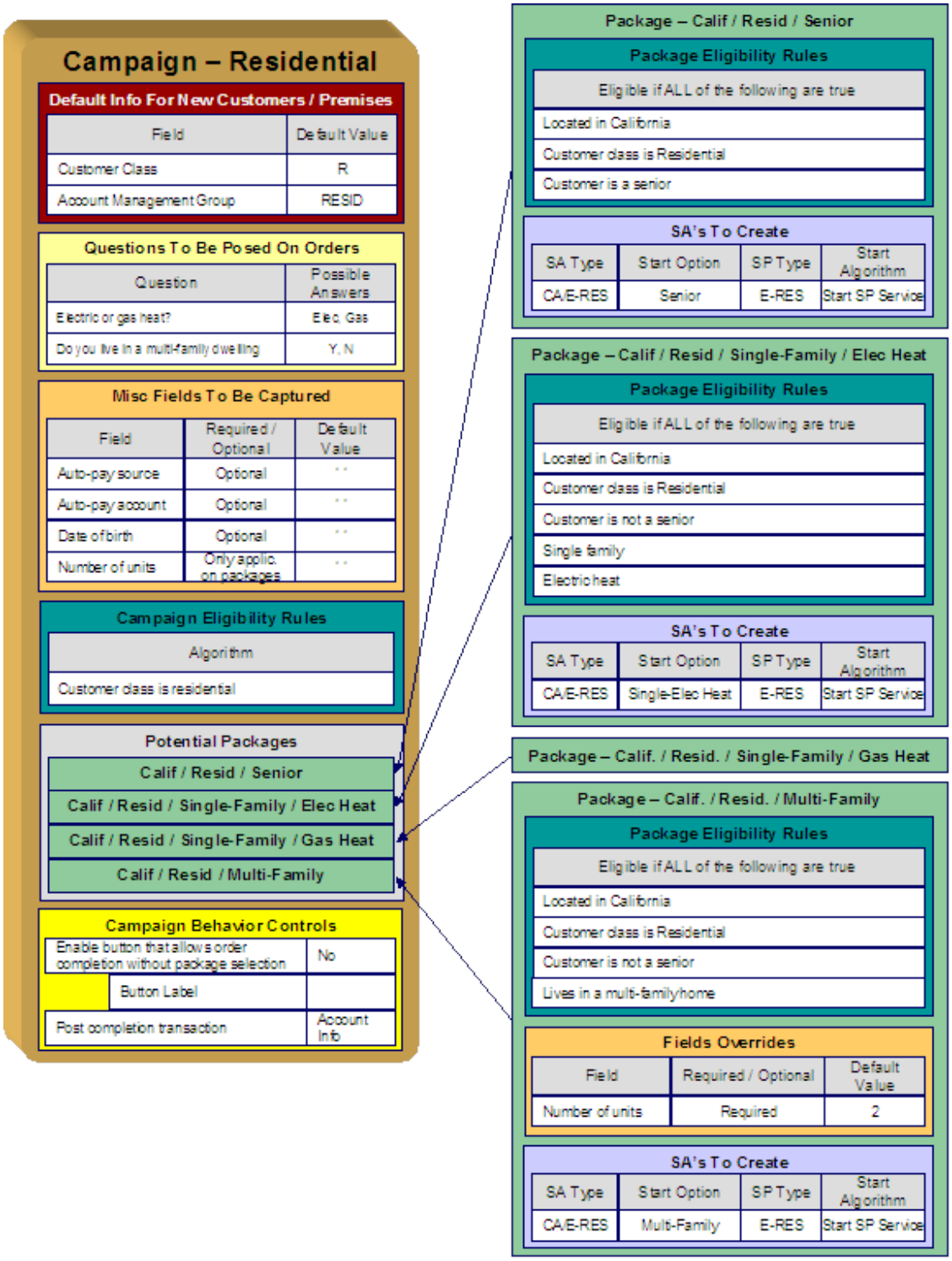
Assume your organization has a campaign with several packages. Each package has the following restrictions in respect of its use:

- "Package 1" can be offered to residential customers in California who are senior citizens.
- "Package 2" can be offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with electricity.
- "Package 3" can be offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with gas.
- "Package 4" can be offered to residential customers in California who live in multi-family homes who are not senior citizens.

Also assume that all customers are encouraged to pay automatically.

The following diagram illustrates how such a campaign might look:

Copyright © 2026, Oracle and/or its affiliates.



The topics that follow describe the campaign and packages used to support this scenario. We'll start by explaining each package. Then we'll describe how the campaign will be setup. This seems backwards, but it mirrors how you should design this type of campaign:

- Senior Citizen Package
- Single-Family Electric Package
- Single-Family Gas Package
- Multi-Family Package

Copyright © 2026, Oracle and/or its affiliates.

- The Campaign

Senior Citizen Package

The package that is offered to residential customers in California who are senior citizens has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer class = Residential
  - Division = California
  - Birth date equates to that of a senior citizen
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:
  - The **SA Type** is one that's used for an electric residential service agreement (hence the E-RES SA type).
  - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for a senior citizen.
  - The service agreement needs to be linked to an electric residential service point (hence the E-RES SP type).
  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Top of the Page

Single-Family Electric Package

The package that is offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with electricity has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer class = Residential
  - Division = California
  - Birth date does not equate to that of a senior citizen
  - The premise is a single-family home
  - The premise is heated with electricity
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:
  - The **SA Type** is one that's used for an electric residential service agreement (hence the E-RES SA type).
  - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
  - The service agreement needs to be linked to an electric residential service point (hence the E-RES SP type).

Copyright © 2026, Oracle and/or its affiliates.

  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Top of the Page

Single-Family Gas Package

The package that is offered to residential customers in California who live in single-family homes who are not senior citizens and who heat their homes with gas has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer class = Residential
  - Division = California
  - Birth date does not equate to that of a senior citizen
  - The premise is a single-family home
  - The premise is heated with gas
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:
  - The **SA Type** is one that's used for a gas residential service agreement (hence the G-RES SA type).
  - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
  - The service agreement needs to be linked to a gas residential service point (hence the G-RES SP type).
  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Top of the Page

Multi-Family Package

The package that is offered to residential customers in California who live in multi-family homes who are not senior citizens has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer class = Residential
  - Division = California
  - Birth date does not equates to that of a senior citizen
  - The premise is a multi-family home
- This package requires additional information that was not requested at the campaign-level. We're going to assume that the multi-family rate needs to know the number of units in the building. This means that this package needs an additional field (number of units).
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:

Copyright © 2026, Oracle and/or its affiliates.

  - The **SA Type** is one that's used for an electric residential service agreement (hence the E-RES SA type).
  - The SA Type's **Start Option** is one that will automatically setup the service agreement with the appropriate rate and contract options for this type of customer.
  - The service agreement needs to be linked to an electric residential service point (hence the E-RES SP type).
  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions in this package. This is because this package doesn't have additional questions.

Top of the Page

The Campaign

Note the following about this campaign:

- Orders can be created for both new and existing customers. Because new customers are possible, we have specified information to default onto orders for new customers. Note that we defined "residential-oriented" default values (but remember, these values can be overridden on the order).
- The questions on the campaign are those whose answers play a part in each package's eligibility. You could also ask additional marketing survey-oriented questions that have nothing to do with package eligibility.
- The miscellaneous fields highlight an interesting situation. Notice that the date-of-birth and "automatic payment" fields are optional. However the field that holds the number of dwelling units is "only applicable on packages". When a field has this designation, the user will only be prompted to supply this field's value if a package is chosen where this field is applicable. In our example, only the Multi Family Package uses this field. This package will be setup to indicate that this field is required whereas the other packages will be setup to indicate that this field is not used.

> **Note**
>
> **Bottom line.** The "miscellaneous fields" on a campaign must also include package-specific fields.

- All campaigns need at least one "highlight rule". These rules control whether the campaign appears in the "eligible campaigns" content zone when a customer is selected on Control Central. Also note that "highlight rules" also control whether the campaign appears on an order's eligibility tree (campaigns that appear in this tree are alternate campaigns that may be used on the order).
- We're going to assume that this is a residential campaign and therefore you would specify a highlight rule that returns true if the customer's customer class is residential.
- Because this campaign has packages, we have indicated that the button that allows an order to be completed without selecting a package is disabled. This means a user must choose one of the packages to complete the order.
- This campaign is setup so that the user will be transferred to Control Central - Account Info after the order is completed. This is a subjective decision as you could transfer the user to a variety of transactions. It's just that the Account Info page provides a nice confirmation of how the customer looks after the order has updated / created new information.

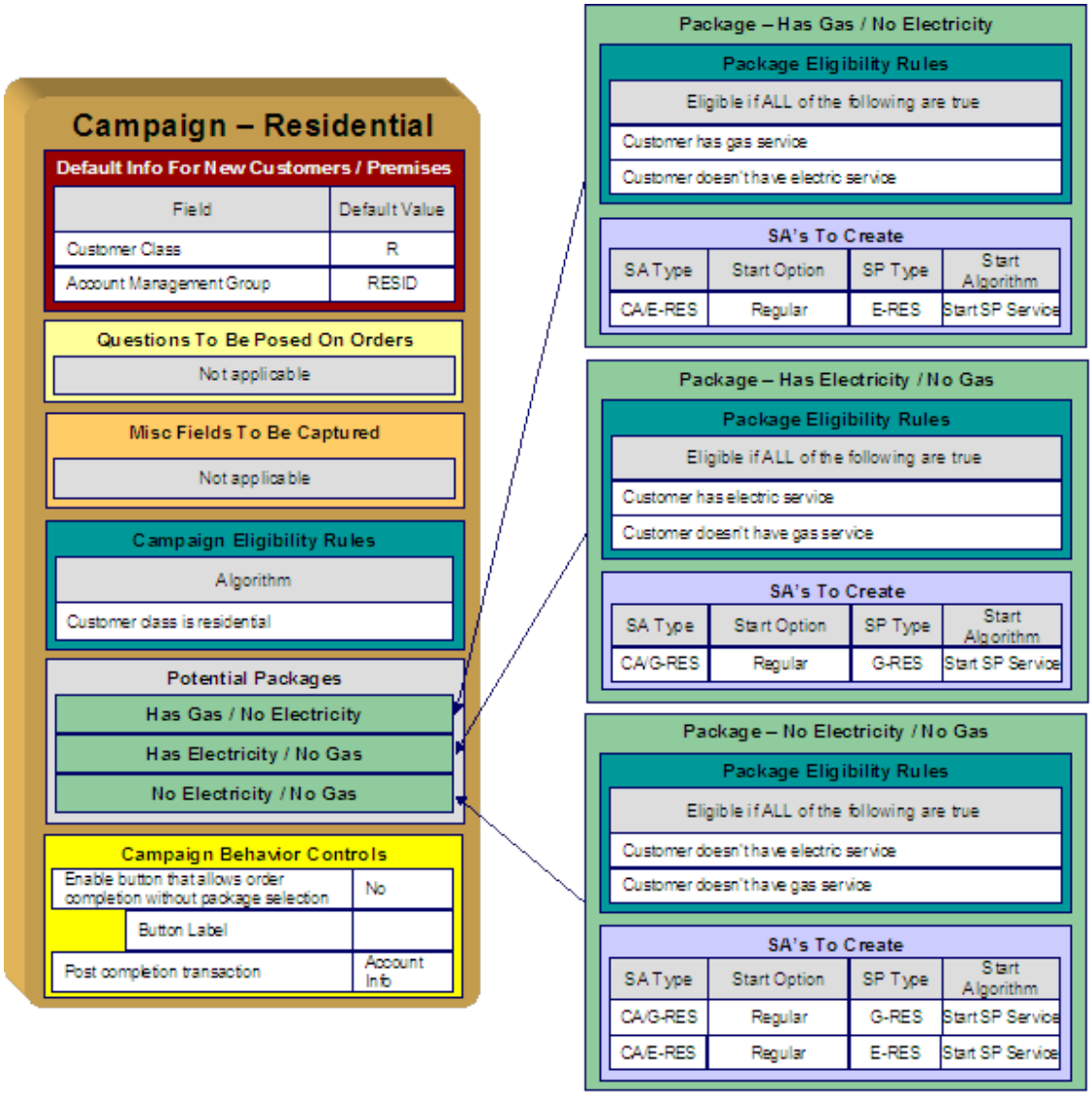
Copyright © 2026, Oracle and/or its affiliates.

Packages Limited By Current State Of Service

Assume your organization has a campaign with several packages. Each package has the following restrictions in respect of its use:

- "Package 1" can be offered to residential customers in California who have gas service, but don't have electric service.
- "Package 2" can be offered to residential customers in California who have electric service, but don't have gas service.
- "Package 3" can be offered to residential customers in California who have neither electric nor gas service.

The following diagram illustrates how such a campaign might look:



The topics that follow describe the campaign and packages used to support this scenario. We'll start by explaining each package. Then we'll describe how the campaign will be setup. This seems backwards, but it mirrors how you should design this type of campaign:

Copyright © 2026, Oracle and/or its affiliates.

- Has Gas / No Electricity Package
- Has Electricity / No Gas Package
- No Electricity / No Gas Package

Has Gas / No Electricity Package

The package that is offered to customers that currently have gas service, but don't have electricity service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer has gas service
  - Customer doesn't have electricity service
- Refer to A Package With Service Type Comparisons for more information.
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:
  - The **SA Type** is one that's used for an electric residential service agreement (hence the E-RES SA type).
  - The SA Type's **Start Option** is one that will setup the service agreement with the appropriate rate and contract options.
  - The service agreement needs to be linked to an electric residential service point (hence the E-RES SP type).
  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Top

Has Electricity / No Gas Package

The package that is offered to customers that currently have electric service, but don't have gas service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer has electric service
  - Customer doesn't have gas service
- We only need a single service agreement created if the package is selected and therefore there's just one entry in SAs To Create. Please note the following about the information used to create this service agreement:
  - The **SA Type** is one that's used for a gas residential service agreement (hence the G-RES SA type).
  - The SA Type's **Start Option** is one that will setup the service agreement with the appropriate rate and contract options.
  - The service agreement needs to be linked to a gas residential service point (hence the G-RES SP type).
  - We've indicated a Start Algorithm of Start SP Service. Refer to STRM-VT for an example of such an algorithm (type).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

Copyright © 2026, Oracle and/or its affiliates.

Top

No Electricity / No Gas Package

The package that is offered to customers that don't currently have electricity or gas service has the following traits:

- It has eligibility rules that restrict its use to customers who meet the following criteria:
  - Customer doesn't have electric service
  - Customer doesn't have gas service
- We need two service agreements created (one for electricity, one for gas).
- We have not shown additional questions or miscellaneous fields in this package. This is because this package doesn't have additional questions or additional fields.

The Big Picture Of Package Eligibility Rules

The packages described under Examples of Campaigns and Packages have a variety of eligibility rules. Designing these rules can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

> **Note**
>
> Don't confuse package eligibility rules with those associated with a campaign. Package eligibility rules control whether a customer is allowed to choose a package on the order transaction. Campaign eligibility rules control whether a campaign is highlighted on the "eligible campaigns" content zone and on an order's eligibility tree.

- Criteria Groups versus Eligibility Criteria
- Defining Logical Criteria
- Examples Of Package Eligibility Rules

Criteria Groups versus Eligibility Criteria

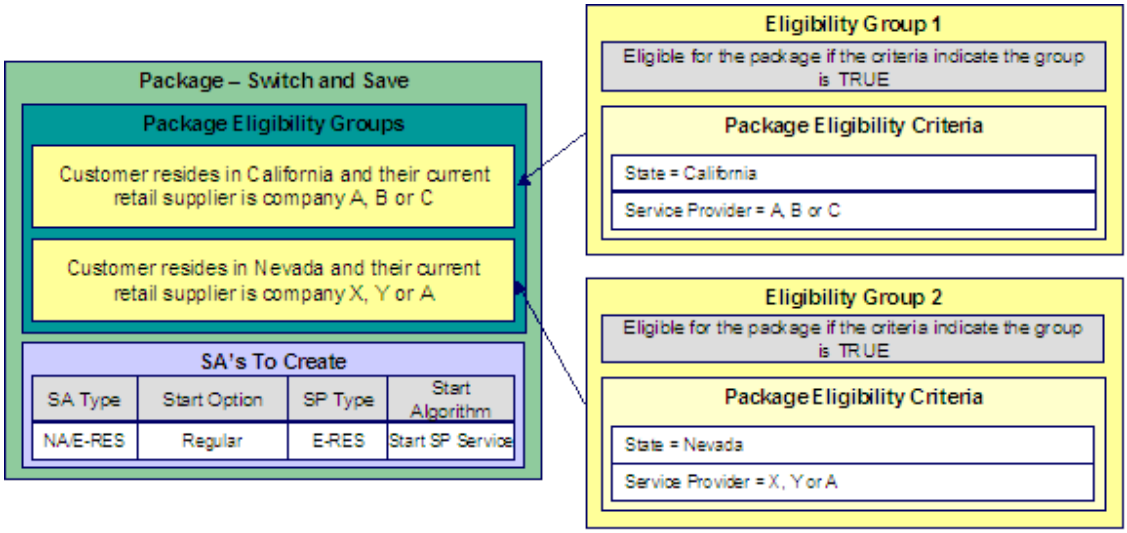
Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A package's criteria groups control whether a customer is eligible to choose a package. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered TRUE or FALSE for a given customer.
- When you create a group, you define what should happen if the group is TRUE or FALSE. You have the following choices:
  - The customer is eligible to choose the package
  - The customer is not eligible to choose the package
  - The next group should be checked.

Copyright © 2026, Oracle and/or its affiliates.

We'll use the following example to help illustrate these points. Assume a package exists that can only be selected if:

- The customer resides in California and their current retail supplier is company A, B or C
- OR, the customer resides in Nevada and their current retail supplier is company X, Y or A



This package requires two eligibility groups because it has two distinct conditions:

- IF (STATE = CALIFORNIA AND (SERVICE PROVIDER IN (A, B, C))
- IF (STATE = NEVADA AND (SERVICE PROVIDER IN (X, Y, A))

If either condition is true, then the customer is eligible for the package.

You'd need to setup the following criteria groups in order to support this requirement:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer resides in California and their current retail supplier is company A, B or C	Eligible	Check next group
2	Customer resides in Nevada and their current retail supplier is company X, Y or A	Eligible	Ineligible

The following criteria will be required for each of the above groups:

Group 1: Customer resides in California and their current retail supplier is company A, B or C				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	State = California	Check next condition	Group is false	Group is false
20	Electric retail supplier in (A, B, C)	Group is true	Group is false	Group is false

Copyright © 2026, Oracle and/or its affiliates.

Group 2: Customer resides in Nevada and their current retail supplier is company X, Y or A				
Seq	Logical Criteria	If Eligibility Criteria is TRUE	If Eligibility Criteria is FALSE	If Insufficient Data
10	State = Nevada	Check next condition	Group is false	Group is false
20	Electric retail supplier in (X, Y, A)	Group is true	Group is false	Group is false

The next section describes how you'd setup the specific logical criteria in each of the groups.

Defining Logical Criteria

When you setup an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the **field to be compared**:

- You can choose one of the miscellaneous fields on the order (keep in mind that miscellaneous fields also hold the answers to any questions posed on the order).
- You can execute an algorithm to retrieve a field value from someplace else in the system. This is a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

You have the following choices in respect of identifying the **comparison method**:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm whose job it will be to perform the comparison (and return TRUE, FALSE or INSUFFICIENT DATA). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The Examples Of Package Eligibility Rules provide numerous examples to help you understand this design.

Examples Of Package Eligibility Rules

The topics in this section provide examples about how to setup package eligibility rules.

- A Package That Is Always Eligible
- A Package With A Time Span Comparison
- A Package With Service Type Comparison
- A Package With More Complex Operators

A Package That Is Always Eligible

Copyright © 2026, Oracle and/or its affiliates.

If a package is always eligible, it still needs an eligibility group. Good examples of such packages are shown under An Easier Way To Create One Time Charges. A package that is always eligible would require the following eligibility information:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Always eligible	Eligible	Eligible

The following criteria will be required for this group:

Group 1: Always eligible					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's account	Algorithm: Always true	Group is true	Group is true	Group is true

This group's criterion is odd in that it's not comparing anything. In order to setup this criterion, you have to use the following "trick":

- Field to Compare. You can pick any miscellaneous field on the campaign. If the campaign has no miscellaneous fields, you'll need to setup an algorithm to retrieve a field from the order (refer to PKEL-ENRFLD for an example of this type of algorithm). In the above example, we used an algorithm to retrieve the order's account.
- Comparison Method. We chose a comparison algorithm that always returns a value of TRUE (refer to PKCC-DEFAULT for an example of this type of algorithm).
- You'll notice that if a value of TRUE is returned, the Group is true (and we've setup the group to indicate a true group means the customer is eligible for the package).

We understand this is a little complicated, but this design provides the power necessary to support very complex criteria. The following sections provide more examples to help you form a better understanding of these concepts.

A Package With A Time Span Comparison

The Senior Citizen Package has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date equates to that of a senior citizen

These rules require only one eligibility group on the package. It would look as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Senior	Eligible	Ineligible

The following criteria will be required for this group:

Copyright © 2026, Oracle and/or its affiliates.

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	Check next condition	Group is false	Group is false
20	Algorithm: retrieve order's division	= CA	Check next condition	Group is false	Group is false
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	Group is true	Group is false	Group is false

The first two criteria are easy; they simply retrieve a field on the order and compare it to a given value. If the comparison doesn't result in a TRUE value, the Group is false (and, the group indicates that if the group is false, the customer isn't eligible for the package). If the comparison results in a TRUE value, the next condition is checked.

The last criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered to be a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base package as described below.

- Field to Compare. The miscellaneous field in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of TRUE if the related field value (the customer's date of birth) is greater than 65 years (refer to PKCC-TIMESPN for an example of this type of algorithm).

You'll notice that if a value of TRUE is returned by the True if senior algorithm, the group is true (and we've setup the group to indicate a true group means the customer is eligible).

> **Note**
>
> **The time span algorithm can be used to compare days, weeks, months, etc.** Refer to PKCC-TIMESPN for more information about this algorithm.

A Package With Service Type Comparison

The Has Gas / No Electricity Package has the following eligibility rules:

- Customer has gas service
- Customer doesn't have electricity service

These rules require only one eligibility group on the package. It would looks as follows:

Copyright © 2026, Oracle and/or its affiliates.

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Has Gas, Doesn't Have Electricity	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Has Gas, Doesn't Have Electricity					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: check if customer has gas service	= TRUE	Check next condition	Group is false	Group is false
20	Algorithm: check if customer has electric service	= TRUE	Group is false	Group is true	Group is false

Both criteria are similar - they check if the customer has a given type of service and act accordingly. We used a "field to compare" algorithm to pull this off, but this algorithm is a bit counter intuitive (but understanding it will provide you with another way to implement complex eligibility criteria):

- Field to Compare. We chose a "field to compare" algorithm that checks if an account has service agreements that belong to a given set of service types. It returns a value of TRUE if the customer has an active service agreement that matches one of the service types in the algorithm. In our example, the "check if customer has gas service" algorithm returns a value of TRUE if the customer has at least one active service agreement whose SA type references the gas service type. The "check if customer has electric service" algorithm is almost identical, only the service type differs. Refer to PKEL-SVCTYP for an example of this type of algorithm.
- Comparison Method. We simply compare the value returned by the algorithm to TRUE and indicate the appropriate response.

> **Note**
>
> **Bottom line.** The "field to compare" algorithm isn't actually returning a specific field's value. Rather, it's returning a value of TRUE or FALS (this is not a misspelling). This value is, in turn, compared by the "comparison method" and the group is set to true, false or check next accordingly.

A Package With More Complex Operators

One of the eligibility groups described under Criteria Groups versus Eligibility Criteria has the following eligibility rules:

- Customer must live in California
- Service provider is A, B or C

Copyright © 2026, Oracle and/or its affiliates.

The eligibility group in which these rules would be defined looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Lives in California, current service provider is either retailer A, B or C	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Lives in California, current service provider is either retailer A, B or C					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's state	= CA	Check next condition	Group is false	Group is false
20	Misc Field On Order: Current electric supplier	IN A, B, C	Group is true	Group is false	Group is false

This example is very simple. The only thing worth pointing out is that a number of operators can be used when you define a "comparison method". Refer to Package - Eligibility for the complete list of operators.

The Big Picture Of Campaign Eligibility Rules

In the previous section, we explained how to Design Eligibility Rules For A Package. In this section, we explain how to design eligibility rules for a campaign.

A campaign's eligibility rules control two things:

- Whether the campaign appears in the Eligible Campaigns content zone. This content zone suggests possible campaigns for the current customer displayed on control central. You might want to use this zone if you have different campaigns that are offered to different types of customers. Refer to Real-time Marketing of Additional Services to a Customer for an example of how this content zone can be used to "up sell" additional services.
- Whether the campaign appears on an order's eligibility tree. Campaigns that appear in this tree are alternate campaigns that may be used on the order. Refer to Real-time Marketing of Services to a Prospect for an example of how an order's eligibility tree can contain alternate campaigns for an order.

Designing these criteria can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup campaign eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Don't confuse eligibility rules for a package with those associated with a campaign. Package eligibility rules control whether a customer is allowed to choose a package. Campaign eligibility rules control whether a campaign is highlighted on the Eligible Campaigns content zone and on an order's eligibility tree.

- Campaign Eligibility Rules Are Not Strictly Enforced
- Campaign Eligibility Rules Are Defined Using Algorithms
- Examples Of Campaign Eligibility Rules

Campaign Eligibility Rules Are Not Strictly Enforced

While the system prevents ineligible packages from being selected for an order (based on the package's eligibility rules), the system does not strictly enforce a campaign's eligibility rules. In other words, a user can change the campaign on an order to any active campaign. This "laxness" is intentional; otherwise it could become impossible to change an order's campaign.

It might be more helpful to think of campaign eligibility rules as "highlight conditions". These "highlight conditions" simply control whether the campaign appears in the Eligible Campaigns content zone and on an order's eligibility tree.

Campaign Eligibility Rules Are Defined Using Algorithms

You define a campaign's eligibility rules using one or more algorithms on the campaign. These algorithms are relatively straightforward as they simply return one of the following values:

- Eligible. Meaning the customer is eligible for the campaign.
- Ineligible. Meaning the customer is not eligible for the campaign
- Check Next Algorithm. Meaning that the next algorithm should be checked. You'd only design an algorithm to return such a value if multiple algorithms are used to determine a customer's eligibility for a campaign.

> **Note**
>
> **Bottom line.** A customer is eligible for a campaign if at least one of its eligibility algorithms returns a value of "eligible".

The Examples Of Campaign Eligibility Rules provide numerous examples to help you understand this design.

Examples Of Campaign Eligibility Rules

The topics in this section provide examples about how to setup campaign eligibility rules.

Copyright © 2026, Oracle and/or its affiliates.

- A Campaign That Is Always Eligible
- A Campaign That Is Never Eligible
- A Campaign For Residential Customers
- A Campaign For California, Residential Customers

A Campaign That Is Always Eligible

If a campaign is always eligible, it would require a simple eligibility algorithm that returns an indication that it is always eligible. A good example of such a campaign is shown under An Easier Way To Create One Time Charges. Refer to CAEL-DEFAULT for an example of this type of eligibility algorithm.

A Campaign That Is Never Eligible

If a campaign is never eligible, it would require a simple eligibility algorithm that returns an indication that it is ineligible. A good example of such a campaign is shown under Prelude To Start/Stop. Refer to CAEL-DEFAULT for an example of this type of eligibility algorithm.

A Campaign For Residential Customers

If a campaign should only be targeted at residential customers, it would require a simple eligibility algorithm that returns an indication of eligible if the customer's customer class is residential. A good example of such a campaign is shown under Marketing Surveys. Refer to CAEL-CC for an example of this type of eligibility algorithm.

A Campaign For California, Residential Customers

If a campaign should only be targeted at residential customers in California, it would require two eligibility algorithms:

Algorithm No	Algorithm	Example Algorithm
1	Indicate Check Next Algorithm if customer class is "residential"	Refer to CAEL-CC for an example of this type of eligibility algorithm.
2	Indicate Eligible if division is "California"	Refer to CAEL-DIV for an example of this type of eligibility algorithm.

Designing Campaigns and Packages

Designing campaigns and packages can be simple or challenging. Factors that will influence the complexity of the design process include:

Copyright © 2026, Oracle and/or its affiliates.

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,
  - Your company may have very straightforward packages that only differ based on whether the customer is industrial, residential or commercial. If this describes your company, the setup process will be simple.
  - Alternatively, your company might offer different packages to residential customers based on whether they pay automatically, whether they are a senior citizen, whether they purchases multiple services from you, etc. If this describes your company, then you'll have to define the various eligibility criteria and the respective packages applicable to customers who satisfy each criterion.
- Whether premises exist prior to order taking. For example, a company that distributes a commodity has a well-defined geographic area and therefore new premises typically don't arise when orders are taken.
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.
- Which of the Supported Business Processes you implement.

The topics in this section provide a structured approach to design your campaigns and packages.

- Determine The Place Of Start/Stop
- Determine Which Types Of Service Agreements Are Created
- Determine The Different Start Options That Exist For Each Type Of Service Agreement
- Determine The SP Type (if any) For Each Service Agreement
- Determine The Method Used To Create Each Service Agreement
- Extract Algorithms From The Start Method
- Determine Which Combinations Can Be Grouped Under A Package
- Determine Package Eligibility Rules
- Extract Algorithms From The Eligibility Rules
- Extract Miscellaneous Fields From The Eligibility Rules
- Determine Additional Information That Is Required If A Package Is Selected
- Determine If There Are Additional Questions / Fields For Your Packages
- Determine If Additional Instructions Should Be Shown
- Determine If There Are Campaign-Level Fields / Questions
- Determine The Properties Of Every Miscellaneous Field
- Extract Characteristic From The Miscellaneous Fields
- Extract Column References From The Miscellaneous Fields
- Extract Algorithms From The Column References
- Determine Account / Premise Dependency For Each Column Reference
- Determine If Any Properties Are Overridden On Any Package
- Determine The Eligibility Rules For The Campaign

Copyright © 2026, Oracle and/or its affiliates.

- Determine How The Campaign Behaves
- Setup Sequence

Determine The Place Of Start/Stop

Before you can design your campaigns and packages, you must determine if your organization will use start / stop to initiate new service agreements or whether new service can be initiated using the order transaction. Refer to Order versus Start / Stop for information to help you make this decision.

If you decide to use start / stop to initiate service, we'd recommend creating the following types of campaigns:

- Create a campaign similar to that described under Prelude to Start / Stop.
- If you levy one-time charges, you may also want to create a campaign similar to that described under An Easier Way To Create One Time Charges.
- If your organization performs marketing surveys, you may also want to create a campaign similar to that described under Marketing Surveys.

If you decide to use the order transaction to initiate new service, follow the steps outlined in the following sections. Note that we use the example described under Packages Limited By Answers And Field Values when describing the design process.

Determine Which Types Of Service Agreements Are Created

> **Warning**
>
> We have assumed that you have already designed your organization's SP types, SA types, start options and rates. If you have not yet done this, we recommend familiarizing yourself with the concepts described in Designing SP Types, Rates and Defining Service Agreement Types before designing your campaigns and packages.

Start off by determining the different types of service agreements that can be created if a package is selected. In our example campaign, only one SA type is used - E-RES (electric residential service).

Determine The Different Start Options That Exist For Each Type Of Service Agreement

After you know which SA types are going to be created, determine the potential start options that can be used for each SA type. In our example campaign, several start options are possible for the E-RES (electric residential service):

SA Type	Start Option

Copyright © 2026, Oracle and/or its affiliates.

E-RES	Senior
E-RES	Single-Elec Heat
E-RES	Single-Gas Heat
E-RES	Multi-Family

Determine The SP Type (if any) For Each Service Agreement

The SA types and start options control how the service agreement will look (i.e., they control the rate, contract riders, characteristics, interval profiles, time-of-use maps, etc. defaulted onto new service agreements created when a package is selected). During this step, you define the SP type of the service point that will be linked to the service agreement. If several different types of SP types can be used for a service agreement, you'll have to choose one as this is the type of service point that will be created when the service agreement is created. If you plan to allow the package to be used to start service at an existing premise (and you don't want the system to create new service points), you will have to develop new plug-ins. Please speak to your implementation group if you need more information.

In our example campaign, only one SP type is created (or reused if the premise already has a service point of this type) - E-RES (electric residential service point):

SA Type	Start Option	SP Type
E-RES	Senior	E-RES
E-RES	Single-Elec Heat	E-RES
E-RES	Single-Gas Heat	E-RES
E-RES	Multi-Family	E-RES

> **Note**
>
> **Many service agreements don't need SP types.** For service agreements that don't use service points (e.g., deposit service agreements, charitable contributions, one-time charges, etc.), you don't have to reference a SP type.

Determine The Method Used To Create Each Service Agreement

For each SA Type / Start Option / SP Type combination, determine the algorithm that will be used to create the service agreement and the service point. The base package is supplied with a limited number of such algorithm types (refer to STRM-AS and STRM-VT for examples).

SA Type	Start Option	SP Type	Start Method
E-RES	Senior	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists

Copyright © 2026, Oracle and/or its affiliates.

E-RES	Single-Elec Heat	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists
E-RES	Single-Gas Heat	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists
E-RES	Multi-Family	E-RES	Algorithm: Create new service agreement and stop old service agreement if active SA with the same service type exists

Extract Algorithms From The Start Method

The start methods illustrated above all require the same algorithm:

Algorithm Type	Algorithm	Parameter Values
Create SP and create SA (see STRM-VT)	Always create new SA and stop existing SA	2 (stop existing service)

Determine Which Combinations Can Be Grouped Under A Package

Now you determine which combinations of service agreements can be offered to a customer as a package. In our example campaign, a customer can only choose one type of service agreement (i.e., each entry in our table is mutually exclusive). This means that each entry is a separate package:

Package	SA Type	Start Option	SP Type	Algorithm
1	E-RES	Senior	E-RES	STRM-VT (Switch Option)
2	E-RES	Single-Elec Heat	E-RES	STRM-VT (Switch Option)
3	E-RES	Single-Gas Heat	E-RES	STRM-VT (Switch Option)
4	E-RES	Multi-Family	E-RES	STRM-VT (Switch Option)

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Packages Limited By Current State Of Service for an example of a package that has multiple service agreements.

Determine Package Eligibility Rules

The topics in this section describe the eligibility rules for each package in our example campaign.

Note

Assumption. This section assumes you are comfortable with the information in The Big Picture Of Package Eligibility.

- Eligibility Rules For Package 1
- Eligibility Rules For Package 2
- Eligibility Rules For Package 3
- Eligibility Rules For Package 4

Eligibility Rules For Package 1

Package 1 (the senior package) in our example campaign has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date equates to that of a senior citizen

These rules require only one eligibility group on the package. It would looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Senior	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	Check next condition	Group is false	Group is false

Copyright © 2026, Oracle and/or its affiliates.

20	Algorithm: retrieve order's division	= CA	Check next condition	Group is false	Group is false
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	Group is true	Group is false	Group is false

Eligibility Rules For Package 2

Package 2 (the single-family / electric heat package) in our example campaign has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a single-family home
- House is heated with electricity

These rules require only one eligibility group on the package. It would looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Single Family, Electric Heat	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Electric Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	Check next condition	Group is false	Group is false
20	Algorithm: retrieve order's division	= CA	Check next condition	Group is false	Group is false
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	Group is false	Check next condition	Group is false
40	Misc Field On Order: Type of Residence	= S (single family home)	Check next condition	Group is false	Group is false
50	Misc Field On Order: Type of Heat	= E (electric)	Group is true	Group is false	Group is false

Copyright © 2026, Oracle and/or its affiliates.

Eligibility Rules For Package 3

Package 3 (the single-family / gas heat package) in our example campaign has the following eligibility rules:

- Customer class = Residential
- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a single-family home
- House is heated with gas

These rules require only one eligibility group on the package. It would looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Single Family, Gas Heat	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Gas Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	Check next condition	Group is false	Group is false
20	Algorithm: retrieve order's division	= CA	Check next condition	Group is false	Group is false
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	Group is false	Check next condition	Group is false
40	Misc Field On Order: Type of Residence	= S (single family home)	Check next condition	Group is false	Group is false
50	Misc Field On Order: Type of Heat	= G (gas)	Group is true	Group is false	Group is false

Eligibility Rules For Package 4

Package 4 (the multi-family package) in our example campaign has the following eligibility rules:

- Customer class = Residential

Copyright © 2026, Oracle and/or its affiliates.

- Division = California
- Birth date does not equate to that of a senior citizen
- Lives in a multi-family home

These rules require only one eligibility group on the package. It would looks as follows:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Residential, Calif, Non Senior, Multi Family	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Residential, Calif, Non Senior, Single Family, Gas Heat					
Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: retrieve order's customer class	= R	Check next condition	Group is false	Group is false
20	Algorithm: retrieve order's division	= CA	Check next condition	Group is false	Group is false
30	Misc Field On Order: Date of Birth	Algorithm: True if senior	Group is false	Check next condition	Group is false
40	Misc Field On Order: Type of Residence	= M (multi family home)	Group is true	Group is false	Group is false

Extract Algorithms From The Eligibility Rules

The eligibility rules illustrated above require the following algorithms:

Algorithm Type	Algorithm	Parameter Values
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Customer Class	Field name: CUST_CL_CD
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Division	Field name: CIS_DIVISION
Time span comparison (see PKCC-TIMESPN)	Determine if birth date equates to that of a senior	Time span type: YEAR Operator: >= Comparison Value: 65

Extract Miscellaneous Fields From The Eligibility Rules

Copyright © 2026, Oracle and/or its affiliates.

During this step, you extract miscellaneous fields that must be captured on the order in order drive your eligibility rules.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package

> **Note**
>
> **Fields explicitly referenced on every order.** You'll notice that we didn't list those fields referenced in the eligibility rules that are explicitly referenced on every order (e.g., customer class, state, division, etc.). This is because these types of fields come "for free".

Determine Additional Information That Is Required If A Package Is Selected

During this step, you determine if there's any information that needs to be created when a package is selected that can't be defaulted from its start option. In our example campaign, package 4 (the Multi-Family Home package), also needs to know the number of housing units at the premise (this information is probably needed to calculate a specific charge in the rate). None of the other packages require additional information. The extra field has been appended to our miscellaneous field table.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Number of Units in House	Premise characteristic	The electric SA created by package 4 requires the number of units when it calculates the ongoing bill segments

Determine If There Are Additional Questions / Fields For Your Packages

Copyright © 2026, Oracle and/or its affiliates.

Our example campaign indicates that all customers are encouraged to pay automatically. In order to pay automatically several fields are required (Start Date, End Date, Autopay Source (i.e., bank / credit card), Account Number, Account Name, Expiration Date). These fields are not "explicitly referenced" on the order and therefore must be captured as miscellaneous fields. We have appended these fields to our list of miscellaneous fields:

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Number of Units in House	Premise characteristic	The electric SA created by package 4 requires the number of units when it calculates the ongoing bill segments
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically

We're going to assume that we don't capture any additional marketing-oriented questions. If we did, we'd need additional fields to hold the response to each question.

Determine If Additional Instructions Should Be Shown

If a package is selected, you can present additional instructions to the user informing them of special tasks they should perform (or special instructions they should give to the customer). For example, you could instruct the user to inform customers who aren't senior citizens that they should call back if they become seniors as they may qualify for a better rate. These instructions can be defined on each package.

For the sake of brevity, we will assume no additional instructions are necessary when a package is selected.

Determine If There Are Campaign-Level Fields / Questions

Copyright © 2026, Oracle and/or its affiliates.

Next, you determine if there are additional fields that need to be captured regardless of the package that's selected. For example, you may want to capture additional demographic or geographic information for marketing purposes.

While our example campaign has no such requirement, for illustration purposes we'll assume you ask the customer to rate the quality of service they have experienced (assuming they are a current customer). This additional question requires another entry in our list of miscellaneous fields.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Type of Residence	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Type of Heat	Premise characteristic	Numerous eligibility rules use this field to qualify a customer for a package
Number of Units in House	Premise characteristic	The electric SA created by package 4 requires the number of units when it calculates the ongoing bill segments
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically
Service Rating	Person characteristic	Campaign level marketing question

Determine The Properties Of Every Miscellaneous Field

At this point, you have amalgamated all miscellaneous fields required for the campaign and its packages. Now you have to define the following for each field:

- Prompt On Order. This is the prompt that will prefix the field when the order is taken. The following table provides examples.
- Campaign Applicability. This controls whether the field is optional , required, or only applicable on package(s).
  - A value of required means the order taker must enter a valid value before packages are displayed for selection.

Copyright © 2026, Oracle and/or its affiliates.

  - A value of optional means the order taker can leave the field blank prior to package selection. You can make an optional field required on individual packages. For example, you would probably make **Date of birth** required on the Senior package.
  - A value of only applicable on packages is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a package, you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- Type Of Response. This controls where the field is stored when the order is completed and how it is validated. You have two options: characteristic or column.
  - If a field is a Characteristic, you must define which entity the characteristic value is to be stored on when the order is completed (Person, Account, Premise or Order). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
  - If a field's value resides is a column (as opposed to a characteristic value), you must define the column reference. A field's column reference controls three things:
- How the field's value is retrieved. A field's value is retrieved when an existing person / account / premise is referenced on an order.
- How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
- How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

NOT_SUPPORTED

Refer to Extract Column References From Miscellaneous Fields for more information about column references.

- Default Value On Order. This controls whether the order taker is supplied with a default value when an order is created. Note well: for optional and only applicable on package fields, you can specify a different default value on each package.

Misc. Field	Prompt On Order	Campaign Applicability	Type Of Response	Default Value On Order
Date of birth	What is your date of birth (used to qualify for senior discount)?	Optional	Characteristic **Entity:** Person **Char Type:** Date of birth	
Type of Residence	What type of residence do you live in?	Required	Characteristic **Entity:** Premise **Char Type:** Type of residence	S (single family)
Type of Heat	How do you heat your home?	Required	Characteristic **Entity:** Premise **Char Type:** Type of heat	E (electric)

Copyright © 2026, Oracle and/or its affiliates.

Number of Units in House	How many units are in the entire house?	Only applicable on package(s)	Characteristic **Entity:** Premise **Char Type:** Number of units in house	
Auto Pay Start Date	What day would you like to start automatic payment processing (YYYY-MM-DD)?	Optional	Column **Column Reference:** Auto Pay Start Date	
Auto Pay End Date	What day would you like to stop automatic payment processing (leave blank if not applicable)?	Optional	Column **Column Reference:** Auto Pay End Date	
Auto Pay Source	From what type of credit card / bank will the funds be debited?	Optional	Column **Column Reference:** Auto Pay Source	
Auto Pay Account Number	What is your credit card / bank account number?	Optional	Column **Column Reference:** Auto Pay Account Number	
Auto Pay Account Name	What is the name on your account?	Optional	Column **Column Reference:** Auto Pay Account Name	
Auto Pay Account Expiration Date	If paying with a credit card, what is the expiration date (MM-YYYY)?	Optional	Column **Column Reference:** Auto Pay Credit Card Expiration Date	
Service Rating	Please rate our service	Optional	Characteristic **Entity:** Person **Char Type:** Service rating	1 (excellent)

Extract Characteristic From The Miscellaneous Fields

Characteristic Type	Type	Valid Values	Entities On Which Characteristic Can Be Defined

Copyright © 2026, Oracle and/or its affiliates.

Date of birth	Ad hoc value: use validation algorithm to confirm date is > 1-Jan-1900 and < 28-Feb-2002	Not applicable	Person, Order
Type of Residence	Predefined value	S: Single-Family M: Multi-Family	Premise, Order
Type of Heat	Predefined value	E: Electric G: Gas	Premise, Order
Number of Units in House	Ad hoc value: use validation algorithm to confirm value is > 0	Not applicable	Premise, Order
Service Rating	Predefined value	0: Not applicable 1: Excellent 2: Very Good 3: Good 4: Poor	Person, Order

Extract Column References From The Miscellaneous Fields

The miscellaneous fields illustrated above require the following column references (we describe the referenced algorithms at the bottom of the table):

Column Reference	Retrieval Algorithm	Validation Algorithm	Post Algorithm	Foreign Key Reference
Auto Pay Start Date	Retrieve account's current automatic payment information	Validate automatic payment information	Post automatic payment information	
Auto Pay End Date	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Source	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	Auto pay source
Auto Pay Account Number	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	

Copyright © 2026, Oracle and/or its affiliates.

Auto Pay Account Name	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date
Auto Pay Expiration Date	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date

There are three types of algorithms referenced in the above table:

- Retrieval algorithm. A column's retrieval algorithm retrieves the column's current value from the database. You'll notice that only one of our columns uses a retrieval algorithm. This is because we've used a sophisticated example of a group of columns that are retrieved at the same time. When you have such a group of columns, nominate one of the columns to do the retrieval work and specify the retrieval algorithm on it.
- Validation algorithm. A column's validation algorithm validates the column's value prior to the database being updated. You'll notice that only one of our columns uses a validation algorithm. This is because we've used a sophisticated example of a group of columns that are validated as a group. When you have such a group of columns, nominate one of the columns to do the validation work and specify the validation algorithm on it.
- Post algorithm. A column's post algorithm updates the column's value on the database when the order is completed. You'll notice that only one of our columns uses a post algorithm. This is because we've used a sophisticated example of a group of columns that are posted together. When you have such a group of columns, nominate one of the columns to do the posting work and specify the posting algorithm on it.

Notice that we have indicated that one of the columns has a "foreign key reference". You declare a "foreign key reference" if the valid values for a column are defined on another table in the system (this also affects the order user interface as a dropdown appears for columns that have foreign key references). In our example, Auto Pay Source is the only column whose valid values are defined on the automatic payment source control table.

Extract Algorithms From The Column References

The column references illustrated above require the following algorithms:

Algorithm Type	Algorithm	Parameter Values

Copyright © 2026, Oracle and/or its affiliates.

Retrieve automatic payment information from an account (see CRRT-AP)	Retrieve automatic payment information from an account	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Validate automatic payment information (see CRVL-AP)	Validate automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Post automatic payment information (see CRPS-AP)	Post automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference

Determine Account / Premise Dependency For Each Column Reference

Later in the design process (see Determine How The Campaign Behaves), you define on the campaign whether an account (and premise) is required, optional or not allowed on its orders. Your selection impacts how you setup column references. This is a little complicated so let's use an example:

- Assume you've setup a campaign to indicate a premise can optionally be defined on its orders.
- If the order-taker doesn't define a premise on an order, any premise-oriented questions and miscellaneous fields (Q&MF) should be suppressed on the order.
- It's easy for the system to know which Q&MFs to suppress when a Q&MF's response is held in a characteristic (because you must define on the campaign if the response will reside in an account, premise, person or order characteristic). For column reference

Copyright © 2026, Oracle and/or its affiliates.

Q&MFs, the system doesn't know if the response is account-oriented, premise-oriented or neither because algorithms handle the validation / updating of the response.

- Therefore, when you use column references on a campaigns Q&MFs, you must define if the responses are account-oriented or premise-oriented (so that the system can suppress the Q&MF when the user doesn't enter an account or premise on the order).

The following table contains the column references defined above as well as their dependency

Column Reference	Question Dependency
Auto Pay Start Date	Must have account
Auto Pay End Date	Must have account
Auto Pay Source	Must have account
Auto Pay Account Number	Must have account
Auto Pay Account Name	Must have account
Auto Pay Expiration Date	Must have account

You'll notice that each column reference is account-oriented (this is because automatic payment fields are stored on the respective account).

Determine If Any Properties Are Overridden On Any Package

At this point, you have defined the properties of all miscellaneous fields at the campaign level. You now need to determine if any of your packages need to override any of these properties. In general, you can only override miscellaneous fields whose campaign applicability is optional or only applicable on package(s).

Our example campaign has only two overrides:

Package 1 requires the following miscellaneous field overrides:

Misc. Field	Override Prompt (seen if a package is selected)	Override Package Applicability	Override Default Value
Date of birth	Not overridden	Required	Not overridden

Package 4 requires the following miscellaneous field overrides:

Misc. Field	Override Prompt (seen if a package is selected)	Override Package Applicability	Override Default Value
Number of Units in House	Not overridden	Required	2

> **Warning**
>
> If you override field properties on a campaign's packages, you will not be able to change the properties of the field at the campaign level.

Copyright © 2026, Oracle and/or its affiliates.

Determine The Eligibility Rules For The Campaign

> **Note**
>
> **Assumption.** This section assumes you are comfortable with the information in The Big Picture Of Campaign Eligibility.

The easiest way to determine a campaign's eligibility is to examine each package's eligibility criteria and abstract common eligibility restrictions onto the campaign. It's important to remember that campaign-level eligibility is only used to highlight if the campaign MAY contain applicable packages; campaign-level eligibility is not strictly enforced.

The packages on our example campaign are all targeted at residential customers in California. Therefore we'll take advantage of two base package algorithms to construct the campaign's eligibility criteria.

Algorithm Type	Algorithm	Parameter Values
Check if account's customer class is in a predefined list (see CAEL-CC)	If customer class is "residential", check next algorithm	If customer's customer class is in list: 30 (check next) If customer's customer class is not in list: 20 (ineligible) List of customer classes: R
Check if account's division is in a predefined list (see CAEL-DIV)	If division is "California", customer is eligible	If customer's CIS division is in list: 10 (eligible) If customer's CIS division is not in list: 20 (ineligible) List of customer classes: CA

Determine How The Campaign Behaves

The final step before you're ready to setup the campaign involves defining a variety of behavioral functions as described below:

- You need to define how the campaign uses accounts and premises. Specifically, on a campaign you define if an account / premise is required / optional / not allowed on its orders. In addition, if an account / premise is required or optional, you can control whether new accounts / premises can be created when an order is completed (the alternative is to force each order to use an existing account / premise). These controls prevent the unwanted proliferation of new accounts and premises for campaigns that are targeted at existing accounts and premises.
- Many fields on an order (e.g., phone type, customer class, division, account management group) can be defaulted on orders created for new customers. You will see these fields on Order - Main. Think about each such field and whether a default value is appropriate. Keep in mind that the order taker can override any of these default values.
- An order taker can indicate that they want to hold an order if they need to do further research before continuing with an order. A background process (TD-ECBK) will create a

Copyright © 2026, Oracle and/or its affiliates.

To Do entry for held orders as a reminder to get back in touch with the customer on a future date. You can define the default To Do role assigned to such To Do entries when you create a campaign. Keep in mind that this value can be overridden when the order taker holds an order.

- You can indicate on a campaign the transaction to which the user should be transferred when orders are completed. Refer to Supported Business Processes for examples of how you might want to use this field.

Setup Sequence

And now you're ready to set up your campaign. The following points describe the order in which this should be done:

> **Warning**
>
> We have assumed that you have already designed your organization's SP types, SA types, start options and rates. If you have not yet done this, we recommend familiarizing yourself with the concepts described in Designing SP Types, Rates and Defining Service Agreement Types before designing your campaigns and packages.

- Setup any new column references required by the campaign's miscellaneous fields. You will have to leave the retrieve, validate and post algorithms blank until after the next step.
- Setup any new algorithms required by:
  - The packages' eligibility rules.
  - The packages' start methods.
  - The campaign's eligibility rules.
  - The campaign's column references.
- Update the relevant column references with the appropriate retrieve, validate and post algorithms.
- Setup any new characteristic types required by the campaign's miscellaneous fields.
- Create the campaign using the information you designed above. Specifically:
  - You defined the question and miscellaneous fields under Determine The Properties Of Every Miscellaneous Field.
  - You defined the eligibility rules under Determine The Eligibility Rules For The Campaign.
  - You defined everything else under Determine How The Campaign Behaves.
  - We recommend leaving the campaign's status as Inactive until all packages are entered.
- Create the campaign's packages using the information you designed above. Specifically:
  - You defined the question and miscellaneous field overrides under Determine If Any Properties Are Overridden On A Package.
  - You defined the eligibility rules under Determine Package Eligibility Rules .
  - You defined the instructions under Determine If Additional Instructions Are Required.

Copyright © 2026, Oracle and/or its affiliates.

- Change the campaign's status to Active.

> **Note**
>
> **The default campaign on the installation record.** If the campaign you've setup is the "default" campaign used when a user presses the Add Order button on Control Central, make sure to specify it as the default campaign on the installation record. Refer to Setting up a New Customer Prior To Starting Service for more information.

Other Useful Information

The topics in this section contain miscellaneous information about the sales & marketing functions.

- Order versus Start/Stop
- Orders Cannot Be Changed After They Are Completed
- Premise and SP Characteristics May Be Populated From Postal Defaults
- To Do List Processing
- Hold and Cancel Reason Codes
- Automatically Cancelling Old Orders
- Additional Things Can Happen When An Order Is Completed
- New Premises and Accounts Are Only Created if an Order is Completed
- Premise Geographic Types Are Populated In An Unusual Way
- Account And Premise Usage Are Controlled By The Campaign

Order versus Start/Stop

There are three different ways to start service for a customer:

- There is the process flow based start service process that supports starting premise-based service, adding new or using existing customers, and has many user configurable settings. For details please refer to The Start Service Process and Request. The following explains the feature of the other two methods.
- As described under definitions, you can use the order transaction to start service.
- Alternatively, you can use start / stop.

The following table provides insight into the pros and cons of each method (after this table you will find our recommendations):

Issue	Order	Start/Stop

Copyright © 2026, Oracle and/or its affiliates.

New customer setup	Information about the new customer can be entered on the order page. A person / account will be setup when the order is completed.	A new person and account must be setup prior to using Start/Stop. See How To Add A New Customer From Control Central for more information.
New premise setup	Information about the new premise can be entered on the order page. A premise (and service points) will be setup when the order is completed.	A new premise and service point(s) must be setup prior to using Start/Stop.
Updates to existing customer information	Changes to a customer's existing account and/or person info may be entered on the order page. The related person and/or account will be updated when the order is completed.	Start/Stop cannot be used to update person or account information. Rather, you must transfer to the person or account page and make the necessary changes.
Incomplete orders	The order transaction allows you to record customer information and hold it without starting service. The customer can call back in the future and you can continue where you left off.	You cannot leave a start half-finished - you either have to complete the start or cancel it.
Stop current service, start new service	The order transaction does not currently provide a mechanism for stopping service (it can only be used to start service).	Start/Stop can be used to stop a customer's current service and start new service.
Automatic versus manual start option selection (start options are used to default a great deal of information onto a service agreement)	If a customer selects a package, the system creates service agreements using the start options defined in the package (i.e., you do not have to select a start option for each service agreement that's created).	When you start service, you must define the specific start option for each service agreement created (actually, you only have to do this for those types of service agreements that use start options).
Starting premise and non-premise oriented services	If a customer selects a package, the system can create both premise and non-premise oriented service agreements. For example, when a package is selected, the system can create electric, gas, deposit, and charitable contribution SAs (based on the information defined on the package).	Multiple premise-oriented service agreements can be created in one interaction. However, a separate interaction is required for each non-premise oriented service agreement.
Changes after completion	An order cannot be modified after it is completed. Any changes (cancellations prior to starting service, changes of the start date, etc.) must be performed using the start/stop transaction.	The start/stop transaction allows you to cancel and make changes to pending start service agreements.

Copyright © 2026, Oracle and/or its affiliates.

Field activity visibility	After an order is completed, the user is transferred to control central where field activities associated with the premise can be seen in trees and context zones.	All field activities associated with all premises associated with all pending starts / stops are shown on the start / stop transaction.
Address changes	Address changes entered on the order transaction are applied when an order is completed (i.e., when a package is selected). They are not held awaiting service agreement activation.	When you enter an address change on the start / stop transaction, this address changed is saved until the first pending start or stop is activated. This means that the address change is effectively held until the start/stop is activated.

Based on the above, we'd recommend the following:

- If your organization has well-defined service locations (i.e., premises) AND a premise's service point type controls the types of services offered to the customer, you should probably only use the order transaction to setup new customer information. The new service agreements should be setup using start / stop. See Prelude To Start/Stop for more information.
- In all other situations, we'd recommend using the order transaction to start service. If customers typically stop and start service in tandem, we'd recommend the following steps:
  - Stop the customer's existing service using start / stop.
  - Return to Control Central - Account Info and select the appropriate campaign from the appropriate portal zone. Keep in mind that the portal zones displayed on this page are based on the user's preferences and security profile. Therefore this zone may not be available to all users.
  - When you select the appropriate campaign, you'll be transferred to the order transaction where you can start the desired package for the customer.

Orders Cannot Be Changed After They Are Completed

The Lifecycle of an Order indicates that once an order is complete, it cannot become pending again. In addition to this rule, we'd like to stress that once an order is complete, you cannot make any changes to the order except to add new log entries.

Premise and SP Characteristics May Be Populated From Postal Defaults

If you specify a new premise on the order transaction and you complete the order, a new premise will be created. This premise's characteristics can be populated from two different sources:

- You could have designed the campaign to have its orders capture premise characteristics. You do this by adding miscellaneous fields to the campaign that are premise characteristics.
- You could have set up postal defaults for the premise's country and postal code.

Copyright © 2026, Oracle and/or its affiliates.

If a given characteristic is referenced on both of the above, the characteristic value on the campaign is used as the premise's characteristic value.

As described above, service points may be created when a package is selected (it depends on the SA to Create algorithm that you've used on the selected package). The base package algorithms that create a new service point when a package is selected will populate the service point's characteristics using the information on the postal defaults (if specified). Refer to STRM-VT for an example of this type of SA to create algorithm.

To Do List Processing

In the base package, there are two To Do lists that may be affected by orders:

- Highlight pending orders. This To Do list contains orders that are in the pending state. Refer to Pending To Do Entries (TD-EPND) for more information about this To Do list.
- Highlight held orders. This To Do list contains orders that are in the held state. Refer to Held To Do Entries (TD-ECBK) for more information about this To Do list. Refer to Order - Hold Dialog for more information about holding orders.

Hold and Cancel Reason Codes

In Lifecycle of an Order, we illustrated how a user can hold a pending order. Before a user can hold an order, they must supply an Order Hold Reason .

The same is true if a user wants to cancel a pending or held order (i.e., they must supply a Order Cancellation Reason).

These reason codes are purely for audit purposes.

Automatically Cancelling Old Orders

The system is delivered with a batch process, Cancel Pending/Held Orders (C1-CNPHO), that can be used to cancel held and/or pending orders that have been open for more than a specified number of days. Refer to the batch control description for more detail.

The delivered process is built using the Plug-in Driven Background Processes framework. Your implementation can introduce alternate logic, by duplicating this batch control and replacing the select and/or process records algorithms. For example, you can introduce a process record algorithm to create a To Do Entry instead of cancelling the order. Adjust the batch parameters to align with any custom logic you introduce.

Additional Things Can Happen When An Order Is Completed

An optional plug-in spot exists on customer class where you can introduce additional logic to be executed when an order is completed for an account that belongs to this customer class.

Another optional plug-in spot exists on campaign where you can introduce additional logic to be executed when an order is completed for the campaign.

Copyright © 2026, Oracle and/or its affiliates.

New Premises and Accounts Are Only Created if an Order is Completed

It's important to understand that a premise and account are only created when an order is completed. This means you can explore alternatives with a prospect who resides at a new premise without introducing the premise / account. However, the system creates a person if you save an order so that an audit of the customer interaction exists in the system.

Premise Geographic Types Are Populated In An Unusual Way

A premise's geographic ID's are used by Control Central when you look for a customer / premise using geographic identifiers.

> **Note**
>
> **Geographic coordinates are optional.** An installation option controls whether at least one geographic coordinate is required on every premise.

The following points describe how the geographic ID's are populated on premises that are created when an order is completed:

- If a postal default exists for the new premise's country / postal code, any geographic identifiers on the postal default are copied onto the new premise.
- If a postal default hasn't been setup for the new premise's country / postal code, the premise will have a single geographic identifier. It is constructed as follows:
  - The geographic type is defaulted from the campaign.
  - The geographic value is set to ***.

> **Note**
>
> **Potential error.** If neither the campaign nor the postal defaults results in a geographic type AND the installation record indicates at least one geographic type is required on a premise, an error is generated explaining that either postal defaults or the campaign should be populated with default geographic information.

Account And Premise Usage Are Controlled By The Campaign

When you set up a campaign, you define how the campaign's orders use accounts and premises. Refer to the description of Account Usage and Premise Usage under Campaign - Main for more information.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Orders

The order transaction can be used to satisfy many diverse requirements. For example, you can use an order to:

- Enroll new customers using a single transaction (i.e., you don't have to use the person, premise, service point, and start / stop transactions to enroll a new customer who resides at a new premise).
- Sell new products to existing customers.
- Update existing person, account and premise information using a single transaction.
- Market your services to prospects uploaded from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer, premise and related service agreements.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Set up proposals for prospective services (and then send a quotation to the customer for these services).
- And more...

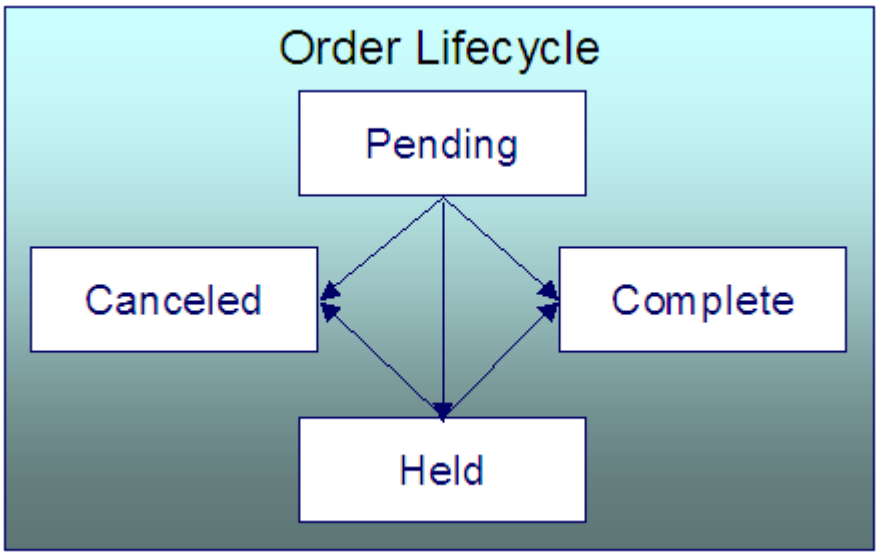
The topics in this section describe the order transaction. Refer to The Big Picture of Campaigns, Packages and Orders for general information about orders.

- Lifecycle of an Order
- Order User Interface Flow
- Order - Main
- Order - Questions & Misc Fields
- Order - Log
- Order - Package Confirmation
- Order - Hold Dialog
- Order - Cancel Dialog

Lifecycle of an Order

The following diagram illustrates the lifecycle of an order.

Copyright © 2026, Oracle and/or its affiliates.



Pending An order starts its life in the Pending state. An order remains in this state until you Cancel, Complete or Hold it.

Held You Hold an order when you want to save an order pending future information. For example, you would hold an order if you need to perform research on the perfect rate for the customer before completing the order. A Held order can be Completed or Canceled.

Complete When a customer selects a package of goods and services, the "V" is updated, service agreements are initiated, and the order becomes Complete. No changes may be made to a Complete order.

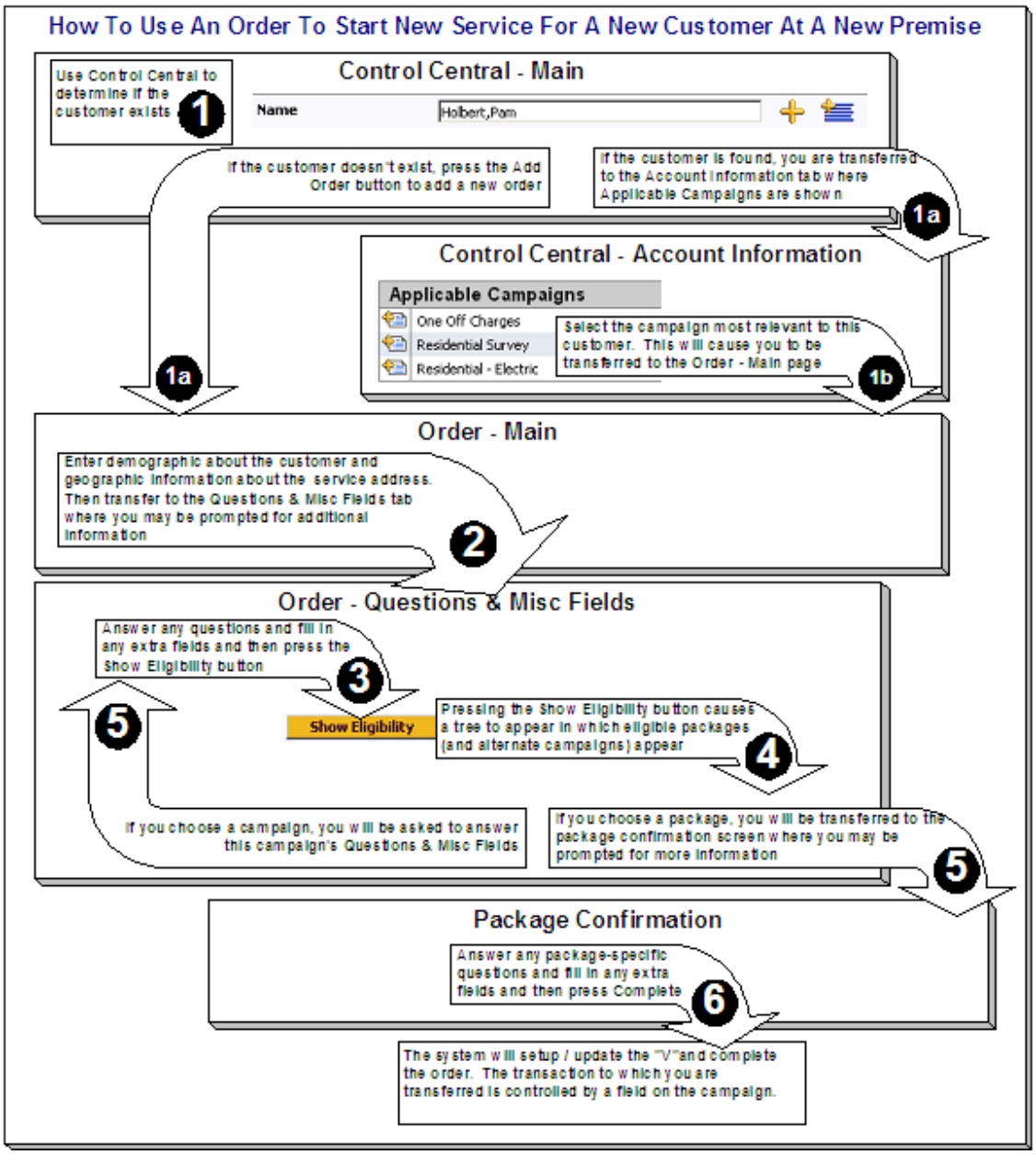
Canceled A Pending or Held order may be Canceled.

Order User Interface Flow

The diagram that appears below illustrates the anticipated navigation path used to process an order.

> **Warning**
>
> An order's campaign controls many aspects of the user interface flow. The illustration that appears below is a sample of what's possible; it may not mirror how your team has setup your company's campaigns.

Copyright © 2026, Oracle and/or its affiliates.



Order - Main

This page is used to define demographic and geographic information about the customer / prospect.

NOT_SUPPORTED

Refer to The Big Picture of Campaigns, Packages and Orders for more information about orders.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Order**, and then **Search.**

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Order User Interface Flow for information about how to use this page.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is Held, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is Canceled, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is Complete, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a Pending order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a Pending or Held order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- The "V" is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- Optional plug-ins are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person, account, or premise and the related person / account / premise has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

An order's **Campaign** controls many aspects of the order's behavior. For example, it controls:

- The type of information defaulted onto an order.
- The eligibility-oriented questions that are displayed on the next tab.
- The superset of packages that can be offered to a customer.
- In addition to the above, campaigns also control high-level eligibility rules and business process flows.

Copyright © 2026, Oracle and/or its affiliates.

An order's **Campaign** is defaulted as follows:

- When you select a customer on Control Central - Search, you are automatically transferred to the Account Information portal. One of the zones on this portal page contains the campaigns that can be offered to the customer. If you select a campaign from this zone, you are transferred to the Order transaction with the respective Campaign defaulted. Note, the Applicable Campaigns zone only appears if the CSR has modified their preferences to display this zone.
- In all other situations, the order's campaign defaults from the Installation Record.

You may change an order's **Campaign** at will. Note that changing the campaign will result in the **Questions & Misc Fields** (the next tab) being refreshed with values associated with the new campaign. However, other values typically defaulted from the campaign (such as default customer class) do not get refreshed so that data you may have entered does not get overwritten.

Start Date defines the date on which service will start. This date becomes the start date on service agreements created when this order is completed. If you need to change the start date on these service agreements, transfer to Start/Stop (where all of the pending start service agreements will appear).

> **Note**
>
> Manual date entry is disabled on the start date and the date picker must be used when the Disable Manual Date Entry **Option Type** on the Customer Information Options Feature Configuration has been enabled.

The topics that follow describe the remaining sections on this page

- Order - Main - Person Information
- Order - Main - Service Address
- Order - Main - Account Information

Order - Main - Person Information

The **Person Information** section contains demographic information about the customer / prospect.

> **Warning**
>
> This page contains a subset of person-oriented information. If you need to define other person-oriented information, transfer to Person - Main after the order is completed.

If this is a new customer, select Create New Person and enter demographic information in the fields that follow. Otherwise, select Use Existing Person and use the **Name** field or press the search button to find the customer. After you select an existing person, the following events takes place:

- The person's name appears beneath the person search (and the person context menu can now be used to drill down to person-oriented pages). Note, the person's name is formatted

Copyright © 2026, Oracle and/or its affiliates.

by a plug-in algorithm on the installation record. Refer to the base package's name format algorithm (PERS-INFO) for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

- If any of the Questions & Misc Fields (on the next tab) are used to capture person-oriented characteristics, the person's existing characteristic values are defaulted.
- If the selected person has at least one account, information about the account is displayed in the **Account Information** section (if the **Campaign** makes uses of accounts). In addition, if any of the Questions & Misc Fields (the next tab) are used to capture account-oriented characteristics, the account's existing characteristic values are populated in the respective rows on this tab.
- Information about the person is displayed in the fields that follow.

Use **Person/Business** to indicate if the customer is a Person or a Business. This value controls how the person's primary name is validated. Note, for new customers, this value defaults based on the **Campaign**.

Names are used by Control Central when you look for a customer based on their name. In addition, a person's primary name is the addressee on the person's bill unless overridden by the **Override Mailing Name** (defined on Person - Correspondence Information). To modify a name, simply move to a field and change its value. To add a new name, press + to insert a row, then fill in the information for each field. The following fields display:

- Use **Name Type** to indicate if the name is an Alias, Alternate Representation, Doing Business As, Legal, or Primary name. Note, for new persons, a value of Primary is defaulted.
- Use **Person Name** to define the person's name. Note well, the name is case sensitive.

> **Note**
>
> **Alternate representations of a person's name.** You would use an Alternate Representation for a person's name when you have an alternate ways to define the person's primary name. Alternate representations are typically used in countries that use multiple character sets (e.g., the Primary name is entered in Chinese, the Alternate Representation is entered in English). When a person has an alternate name, both the main and alternate names can be used to search for a person. The Alternate Representation **Name Type** only appears if you have enabled alternate addresses on the installation record. Refer to the description of the Alternate Representation field under Installation - Main for more information.
>
> **Validation and formatting are performed by plug-ins.** The validation that is applied to **Person Name** (e.g., a comma separating the last and first name - Smith,Patricia) is controlled by a plug-in algorithm on the installation record. More than one algorithm can be plugged into this plug-in spot. Algorithms plugged into this plug-in spot can also perform formatting such as removing special characters that your implementation's business rules do not allow in person names. Other algorithms can be configured to remove or replace text.

- The **Do Not Format** switch can be used to bypass any person name formatting done by the algorithms described above. Validation such as requiring name separator is still applied, but no further formatting is applied. Setting the switch only applies while the record is being saved and is then reset. The switch is reset because the formatting is only applied when the record is subsequently changed. This switch is only shown if the Enable Person

Copyright © 2026, Oracle and/or its affiliates.

Name Format Bypass **Option Type** on the Customer Information Options Feature Configuration has been enabled

Person Contacts are used by Control Central to look for accounts and persons. For new customers, the person contact type(s) default based on the Campaign. To modify a person contact, simply move to a field and change its value. To add a new person contact, press + to insert a row, then fill in the information for each field. The following fields display:

- **Person Contact Type** indicates the type of contact such as, home phone, cell phone, or work email.
- **Contact Information** contains the specific contact information such as, the phone number or email address. Enter the contact information in the format described.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to **Contact Information** is controlled by the algorithm that is plugged in on the respective person contact type (refer to Setting Up Person Contact Types for more information). If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the contact information. (This field is only active when the contact's contact routing is phone or fax.)
- The **Primary** switch must be set for exactly one person contact for each contact routing.
- **Do Not Disturb (From)** and **Do Not Disturb (To)** indicate the times when a person contact should not be used. (**Note:** these fields are active when the Person Contact Type's Allow Do Not Disturb information is set to *Optional*.

Phone numbers are used by Control Central to look for accounts and persons. For new customers, the phone type(s) default based on the **Campaign**. To modify a telephone number, simply move to a field and change its value. To add a new telephone number, press + to insert a row, then fill in the information for each field. The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business, ...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. You can also have this plug-in convert input values into formatted values, e.g., this algorithm could transform all numbers into a formatted string. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Person Phone grid appears when the Legacy Person Phone and Email Option Type on the Customer Information Options Feature Configuration has been enabled.

A person's **ID** has several uses:

- It is used by Control Central when you look for a customer / premise based on their ID.
- It is used to highlight potential duplicate persons.
- Control Central displays a person's primary identification in the search results area to help a user identify the customer when multiple customers match the search criteria.

The person ID usage flag on the installation record indicates whether or not at least one id for a person is required or optional. The ID Type defaults from the installation record based on the **Person Type** (Person versus Business). The following fields are used to define a customer's primary ID.

- Select the type of identification from the **ID Type** drop down.
- Enter the identification number in the **ID Number** field. Please note that if the ID number should be formatted (e.g., dashes in an American social security number), you do not have to include the delimiter (e.g. dashes). You can enter the ID number as a contiguous value and the system will format it for you. The format is shown adjacent to the **ID Number** field.

> **Note**
>
> **Formatting is performed by a plug-in.** The format that is applied to a person ID number(e.g., dashes in an American social security number) is controlled by the algorithm that is plugged in on the respective ID Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.
>
> **A person can have many forms of identification.** The order transaction only caters for a single form of ID (as this is the most typical situation). If you want to specify additional person ID's, you must transfer to Person - Main after the person is created (persons are created when the order is saved).

Specify the customer's **Email ID** (if any).

> **Note**
>
> **EmailID** appears when the Legacy Person Phone and Email Option Type on the Customer Information Options Feature Configuration has been enabled.

Define the **Language** in which the person prefers their bills and correspondence printed.

> **Note**
>
> **Default note.** The person's language defaults from Installation Options - Person.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Customer Language for more information on options for supporting multiple languages for your customers.

Use **Life Support / Sensitive Load** to indicate if the person has life support or sensitive load equipment. Valid values are: LS/SL (i.e., the person has life support / sensitive load equipment), None. If the customer has LS/SL equipment, make a note of the type of equipment in the **Life Support / Sensitive Load Notes**.

Note

A premise can also have life support / sensitive load information. If the equipment is physically linked to the premise (e.g., a hospital has life support equipment), you should NOT specify the life support information on the person. Rather, after the premise is added, transfer to Premise - Misc and specify life support information on the premise.

Life support affects alerts and C&C. If life support / sensitive load information is specified, an alert will appear when the person is displayed on control central. In addition, if a person has life support or sensitive load equipment, it is possible for a different credit & collection severance process to be kicked off if the person's account associated has overdue debt. Refer to Designing Your Severance Procedures for more information.

Order - Main - Service Address

The **Service Address** section contains geographic information about the service address:

Warning

This page contains a subset of premise-oriented information. If you need to define other premise-oriented information, transfer to Premise - Main after the order is completed.

Whether or not a premise is required or even allowed on an order is controlled by an option on the **Campaign**. If the **Campaign** disallows the use of a premise, this section will be suppressed. If the **Campaign** allows the use of a premise, an indication of whether a premise is optional or required is displayed adjacent to **Service Address**.

If a premise is allowed on the order, use the dropdown at the top of this section to tell the system about this order's premise:

- Select No Premise Information if you do not want to associate a premise with this order (e.g., for a one-time charge).
- Select Create New Premise if this is a new premise. You should enter the new premise's address in the remaining fields in this section.

Copyright © 2026, Oracle and/or its affiliates.

- Select Use Existing Premise if an existing premise should be used on the order. Use the **Address** and **City** fields or press the search button to find the premise. Refer to Control Central - Search Facilities for suggestions in respect of how to use these fields. After you select an existing premise, the following events takes place:
  - The premise's description appears beneath the premise search fields (and the premise context menu can now be used to drill down to premise-oriented pages).
  - If any of the Questions & Misc Fields (the next tab) are used to capture premise-oriented characteristics, the premise's existing characteristic values are defaulted.
  - Information about the premise is displayed in the fields that follow.

Use **Premise Type** to describe the type of service location.

The address's constituent fields vary based on the **Country**. Please refer to the Country page for more information.

Populate **Trend Area** if the **Postal** is blank. The **Postal** and the **Trend Area** cannot both be blank as the **Trend Area** is a required field on the premise and its value is typically defaulted from postal defaults.

> **Note**
>
> **Default note.** The values of several fields on this and the next page default based on the **Country** and the **Postal**. Refer to Setting Up Premise & Service Point Postal Defaults for more information. Also note, there are several premise-oriented fields that do not appear on this transaction that will be populated on the premise based on the **Country** and **Postal** (e.g., characteristics, time zone, trend area, geo ID's).

Use **CIS Division** to define the jurisdiction in which the premise is located.

Order - Main - Account Information

The **Account Information** section contains account-oriented information.

> **Warning**
>
> This page contains a subset of account-oriented information. If you need to define other account-oriented information, transfer to Account - Main after the order is completed.

Whether or not an account is required or even allowed on an order is controlled by an option on the **Campaign**. If the **Campaign** disallows the use of an account, this section will be suppressed. If the **Campaign** allows the use of an account, an indication of whether an account is optional or required is displayed adjacent to **Account Information**.

If an account is allowed on the order, use the dropdown at the top of this section to tell the system about this order's account:

- Select No Account Information if you do not want to associate an account with this order (e.g., for a person-oriented marketing survey).

Copyright © 2026, Oracle and/or its affiliates.

- Select Create New Account if this is a new account. You should enter the new account's information in the remaining fields in this section.
- Select Use Existing Account if an existing account should be used on the order. If an existing person was selected above, one of the person's accounts (if any) will be automatically displayed in the **Account Information**. In addition, if the order references an existing person, the account search will be limited to accounts associated with this person. If an **Existing Account** is selected, the following takes place:
  - Basic information about the account appears beneath the search area (and the account context menu can now be used to drill down to account-oriented pages).
  - If any of the Questions & Misc Fields on the next tab are used to capture account-oriented characteristics, the account's existing characteristic values are defaulted.
  - Information about the account is displayed in the fields that follow.

If you have set the Installation Option CIS Division Control flag**,** you must define the new account's **CIS Division.** The values are restricted to the user's CIS Division(s). See Users and CIS Division for more information. The value is defaulted to user's primary CIS Division.

Customer Class plays a part in:

- If and when a customer is subject to late payment charges.
- The account's default collection class and when the account debt monitor reviews an account. Refer to The Big Picture of Credit & Collections for more information about how and when an account's debt is reviewed.
- And several other functions. Refer to and Setting Up Customer Classes for more information.

> **Note**
>
> **Default note.** For new accounts, the **Campaign** can default the **Customer Class**. If the campaign does not define a customer class, it can be defaulted from the CIS Division, if shown, or installation options.

The optional Account **Management Group** controls the roles assigned to To Do entries associated with an account. Refer to Setting Up Account Management Groups for more information.

> **Note**
>
> **Default note.** For new accounts, **Management Group** defaults based on the **Campaign**.

The remaining fields are used to describe how the bill should be routed to the customer.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> When an order is completed, several fields that do not appear on this page that reside on Account - Person are populated. For example, **Account Relationship Type, Bill Format**, **Number of Copies**, **Receives Notification** are all populated with default values. If you need to override the default values, transfer to Account - Person after the account is added.

Use **Bill Route Type** to define how the bill is sent to the customer. This field's value defaults from the Installation Record. This value is important as it controls many options on the remainder of this page. If the **Bill Route Type** indicates that bills are routed via Fax, the person's fax number is displayed adjacent (the system knows which of a person's phone numbers is a fax number by the **Phone Type**). If the **Bill Route Type** indicates that bills are routed via Email, the person's **Email** address is displayed adjacent. If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose an appropriate **Address Source** to define which address should be used. Refer to Setting Up Bill Route Types for more information about bill route types.

If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose the appropriate **Address Source** to define which address should be used.

- Choose Mailing Premise on Account if bills should be sent to the address associated with the Mailing Premise on the first page. This address is displayed adjacent.
- Choose Person if bills should be sent to the person's mailing address . This address is displayed adjacent.
- Choose Account Override if bills should be sent to an override address specified below. Typically, you would only choose this option if the person has multiple accounts and each account's bills should be sent to a different address.

If you select a **Address Source** of Account Override , you must enter the address to which bills will be sent in the following address constituents. The number and type of address constituents is based on the **Country** (refer to Defining Countries for more information on address constituents). These fields will be protected for other **Address Sources**. Note, the **Country** defaults from Installation Options - System and several other constituents default based on the **Country** and **Postal** if a Postal Default exists for the postal code.

> **Note**
>
> By default, the **Address Source** field is protected on Order – Main page if the **Bill Route Type** is set to a value other than Postal. To enable the **Address Source** field for all **Bill Route Types** in most circumstances, set the Always Enable Address Source option type on the General System Configuration feature configuration to Y.
>
> Regardless of the **Always Enable Address Source** option value, **Address Source** and **Address Information** are displayed but both are disabled when **Bill Routing Method** is ***Not Postal*** and **Quote Routing Method** (on Account – Persons page) is ***Postal***. A message "*The Main Person receives Quotes at this address. To edit it, go to the 'Persons' tab on the Account.*" is displayed next to the **Address Source.**

Copyright © 2026, Oracle and/or its affiliates.

Order - Questions & Misc Fields

This page has several purposes:

- The order's Campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.). This page allows you to capture the customer's response.
- The order's Campaign can also be setup to capture additional fields to be updated on the order's person / account / premise when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken. This page allows you to capture these field values.
- The order's Campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service. This page allows you capture the customer's response to these types of questions.
- After entering the above information (and the information on the Main page), you can use this page to select a package of goods and services for the customer.
- You can use this page to switch the campaign on the order (which would result in new questions and miscellaneous fields appearing).

NOT_SUPPORTED

Refer to The Big Picture of Campaigns, Packages and Orders for more information.

Open this page using **Sales & Marketing, Order** and then navigate to the **Questions & Misc Fields** tab.

NOT_SUPPORTED

Refer to Order User Interface Flow for information about how to use this page.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is Held, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.

Copyright © 2026, Oracle and/or its affiliates.

- If the order's status is Canceled, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is Complete, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a Pending order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a Pending or Held order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- The "V" is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- Optional plug-ins are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person, account, or premise and the related person / account / premise has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

The number and type of **Questions & Misc Fields** is controlled by the Campaign.

> **Note**
>
> **Some questions can be suppressed.** The campaign contains an indication if each question is account-oriented, person-oriented, premise-oriented or order-oriented. If the question is account-oriented and no account is defined on the **Main** tab, the question will be suppressed. If the question is premise-oriented and no premise is defined on the **Main** tab, the question will be suppressed.

Each row in this grid has a **Prompt** and a **Response**. The value of each **Prompt** is defined on the **Campaign**. The **Response** holds the answer supplied by the customer. Please note the following about the **Response** field:

- On a **Campaign**, you can indicate which questions / miscellaneous fields are required. The Response for a required field is prefixed with an asterisk - * .
- The value of a Response can default from a variety of sources:
  - If an existing person was selected on the **Main** page AND the response is used to capture a person characteristic, the existing characteristic value will be defaulted.
  - If an existing premise was selected on the **Main** page AND the response is used to capture a premise characteristic, the existing characteristic value will be defaulted. If no value exists, the characteristic values will default based on the postal defaults associated with the service address's **Country** and **Postal**.

Copyright © 2026, Oracle and/or its affiliates.

  - If an existing account was selected on the **Main** page AND the response is used to capture an account characteristic, the existing characteristic value will be defaulted.
  - Otherwise, the characteristic value is defaulted from the **Campaign**.
- The type of field used to capture a Response is controlled by the Campaign. Depending on the type of Response you may see Go To buttons and Search buttons.

After entering the **Responses** to the **Questions & Misc Fields**, press the **Show Eligibility** button.

The following may occur when this button is pressed:

- If a new customer is specified on the Main page, a warning will advise you that a new person will be added. The system needs to add a new person so that an audit of the customer interaction exists.
- The tree expands to show the Order - Eligibility Tree containing package, ineligible package, error package, and other campaign nodes:
  - Each **Package** node contains a package that is linked to the campaign whose Eligibility Criteria allows its selection by the customer. You can expand this node to see the various service agreements that will be created if you select the package. Selecting one of these packages causes the Order - Package Confirmation page to appear in which you can confirm your selection.
  - The **Ineligible Package(s)** node contains packages whose Eligibility Criteria prevents their selection by the customer.
  - The **Error Packages** node contains packages whose Eligibility Criteria have returned errors (if any).
  - The **other campaign(s)** node contains alternative campaigns that may be substituted for the current campaign on the package. Selecting one of these campaigns will cause the campaign to be changed on this order. Changing a campaign on an order will result in new questions and miscellaneous fields appearing. Refer to The Big Picture Of Campaign Eligibility for more information.

> **Note**
>
> **Some nodes can be suppressed.** To increase page performance with large campaigns, you can define a feature configuration to suppress the display of the Ineligible Packages, Error Packages or Other Campaign nodes. If a node is suppressed, it does not appear on the eligibility tree and the system does process the information necessary to display the node.

Order - Log

This page contains an audit trail of significant events that transpired in the history of this order.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Order**, and then **Search** and navigate to the **Log** page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is Held, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is Canceled, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is Complete, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a Pending order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a Pending or Held order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

Campaign contains basic information about the order's campaign.

Package only appears if the order has been completed. It contains basic information about the package.

Start Date contains the date on which the service agreement's created by the order will be started.

The **Log** contains an audit trail of significant events that transpired in the history of this order. There are two types of entries in the log:

- The system automatically adds log entries when key events occur. These entries may not be removed or changed.
- Users may add log entries by pressing the + button and then specifying the Details.

Order - Package Confirmation

The Package Confirmation page allows you to check the package you selected before the V is created / updated for the customer. This page opens when you select a package from the eligibility tree on Order - Questions & Misc Fields.

> **Warning**
>
> If you press the **Complete** button, a variety of objects may be created (service agreements, service points, a premise, an account, field activities, etc.). If you need to change any of these objects after order completion, you must transfer to the respective page on which the object is maintained. You can also use Start/Stop to make changes to the service agreements created by this transaction.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Order contains a concatenation of basic information about the order. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Campaign contains basic information about the order's campaign.

Package contains basic information about the package.

Premise is the address at which service will be delivered.

Person is the name of the customer.

Account is the name of the customer and their customer class.

Start Date contains the date on which the service agreement's created by the order will be started.

The **Questions & Misc Fields** grid contains the questions and miscellaneous fields that appeared on the order. Note well, it is possible to design a campaign and a package where some fields are optional on the campaign but required for a specific package. The **Response** of all required fields for the selected package are prefixed with an asterisk. Refer to Order - Questions & Misc Fields for information about the rows in this grid.

> **Note**
>
> **Changing your answers could cause the package to become ineligible.** As described earlier, the fields you enter on an order play are what the system uses to determine if a package can be selected for a customer. If you change the answers to the **Questions & Misc Fields**, the package may no longer be eligible and an error will be presented to you. You will have the choice of reverting to the original answers or returning to the order page where you can select an eligible package.
>
> **Additional questions / misc fields are possible.** It is possible to design a campaign where certain field values are only visible on specific packages. In this scenario, the **Questions & Misc Fields** grid may contain additional rows for a given package. Refer to Designing Campaigns and Packages for more information.

The **Instructions** grid contains package-specific suggestions. These instructions are defined on the package.

The **Messages** grid contains system-generated messages that summarize what will happen if you press the **Complete** button.

Press the **Complete Order** button if you are satisfied with the information on the order. Otherwise, press **Do Not Use this Package** to change or cancel the order.

Order - Hold Dialog

The **Order - Hold** dialog opens if you press the **Hold** button on the order transaction. You use it to define why the order is being held and when you want to be reminded to check up on the order.

Description of Page

Use **Hold Reason** to define why the order is being held.

Copyright © 2026, Oracle and/or its affiliates.

The following fields are used to control the To Do entry that will be created to remind you about this order.

- Use **Reminder Date / Time** to define the latest date on which the To Do entry should be created. The reason we indicated this should be the latest date is because the background process (TD-ECBK) that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Reminder Date** that the To Do entry should be created.
- If the To Do entry should be addressed to a group of users, choose a **Reminder Type** of Send to Role and enter the user group's **To Do Role**. Note, the To Do Role will default from the order's campaign.
- If the To Do entry should be addressed to a specific user, choose a **Reminder Type** of Send to User and enter the user's User ID in **To Do User**.
- **Log Notes** are used to describe why the order is being held. This information appears in the log entry that is created by the system when you hold an order.

If there are any To Do entries (with a status of Open or Assigned) that drill down to this order, a message is displayed indicating that existing To Do entries for this order will be completed. If the order is already in Hold status, existing To Do entries are completed when you click the OK button and a new To Do entry is created when the background process responsible for creating the To Do entries runs.

Order - Cancel Dialog

The Order - Cancel dialog opens if you press the **Cancel** button on the order transaction. You use it to define why the order is being canceled.

Description of Page

Use **Cancel Reason** to define why the order is being canceled.

Use **Log Notes** to describe why the order is being canceled. This information appears in the log entry that is created by the system when you cancel an order.

Maintaining Campaigns

When you add an order, you must reference its campaign. An order's campaign controls:

- The type of information defaulted onto an order. For example, you can setup a campaign to default a given account management group on all orders linked to this campaign. This account management group will then default onto accounts created when the campaign's orders are completed.
- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above - Who is your current energy service provider? / Would you like to pay automatically? / What is your date of birth? - are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level eligibility rules and business process flows.

Copyright © 2026, Oracle and/or its affiliates.

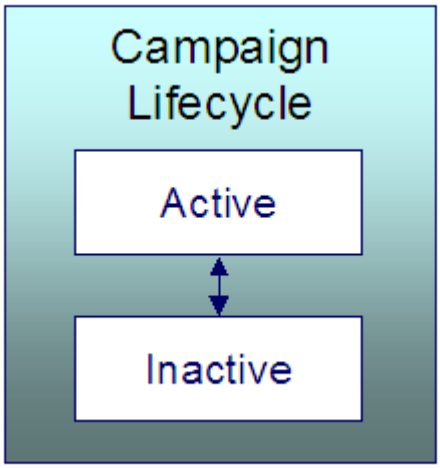
The topics in this section describe the campaign transaction.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Campaigns, Packages and Orders for general information about campaigns. Refer to The Big Picture Of Campaign Eligibility Rules for a discussion of how to control which campaigns can be used for different types of customers. Refer to Designing Campaigns and Packages for guidelines describing how to design your campaigns.

- Lifecycle of a Campaign
- Campaign - Main
- Campaign - Questions and Misc Fields
- Campaign - Algorithms
- Campaign - Package

Lifecycle of a Campaign

The following diagram illustrates the lifecycle of a campaign.



Active An Active campaign may have orders created in respect of it.

Inactive By Inactivating the campaign, you are preventing it from being used on future orders. In addition, Inactive campaigns do not appear in the Applicable Campaigns zone.

Campaign - Main

This page is used to define basic information about a campaign.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Campaign**, and then **Search.**

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

Description briefly describes the campaign.

Campaign Status describes the state of the campaign. Refer to Lifecycle of a Campaign for more information.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new persons - they will not override field values defined on existing persons.

- The system requires all persons to be defined as either a Person or a Business. Use **Default Person / Business** to define the default value for persons created by this campaign's orders.
- Every person contact defined for a person must reference a person contact type. Use **Default Person Contact Type 1**, **Default Person Contact Type 2**, and **Default Person Contact Type 3** to define the first three person contact type defaults for persons created by this campaign's orders.

Note

Person Contact Types appears when the Legacy Person Phone and Email Option Type on the Customer Information Options Feature Configuration has **not** been enabled

- Every phone number defined for a person must reference a phone type. Use **Default Phone Type 1** and **Default Phone Type 2** to define the first two phone types defaults for persons created by this campaign's orders.

Note

Phone Types appears when the Legacy Person Phone and Email Option Type on the Customer Information Options Feature Configuration has been enabled.

Premise Usage controls if and how this campaign's orders use premises. You have the following options:

- Select Not Allowed if this campaign's orders do not reference a premise.
- Select Optional if this campaign's orders do not have to reference a premise. This option also means that a new OR existing premise may be specified on the order. Note, if an order references a new premise, the premise will be created when the order is completed.

Copyright © 2026, Oracle and/or its affiliates.

- Select Optional - Existing Premise Only if this campaign's orders do not have to reference a premise. This option also means that only an existing premise may be specified on an order.
- Select Required if this campaign's orders must reference a premise. This option also means that a new OR existing premise may be specified on the order. Note, if an order references a new premise, the premise will be created when the order is completed.
- Select Required - Existing Premise Only if this campaign's orders must reference a premise. This option also means that only an existing premise may be specified on an order.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new premises - they will not override field values defined on existing premises.

- The system requires all premises to reference a premise type. Use **Default Premise Type** to define the default value for premises created by this campaign's orders.
- Every geographic ID defined for a premise must reference a geographic type . Use **Default Premise Geo Type** to define the default value for premises created by this campaign's orders. Refer to Premise Geographic Types Are Populated In An Unusual Way.

Account Usage controls if and how this campaign's orders use accounts. You have the following options:

- Select Not Allowed if this campaign's orders do not reference an account.
- Select Optional if this campaign's orders do not have to reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select Optional - Existing Account Only if this campaign's orders do not have to reference an account. This option also means that only an existing account may be specified on an order.
- Select Required if this campaign's orders must reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select Required - Existing Account Only if this campaign's orders must reference an account. This option also means that only an existing account may be specified on an order.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new accounts - they will not override field values defined on existing accounts.

- The system requires all accounts to belong to a customer class. Use **Default Customer Class** to define the default value for accounts created by this campaign's orders.
- An account may reference an account management group. Use **Default Acct Management Group** to define the default value for accounts created by this campaign's orders.

If a user holds one of the campaign's orders, they can have a To Do entry generated on a future date to remind them about the order. Use **Default Hold To Do Role** to define the group of users to whom the To Do entry will be sent. Note well, the user who holds an order can indicate that the related To Do entry should be addressed to a specific user ID rather than a **Role**.

Use **Order Open Option** to define the page that is displayed when a user adds a new order for this campaign. This field is only relevant when the order transaction is invoked in "add mode"

Copyright © 2026, Oracle and/or its affiliates.

and a campaign is passed to this transaction (e.g., if a user adds an order from the Applicable Campaigns Zone).

A user can be automatically transferred to a different transaction when they complete an order. Refer to Supported Business Processes for examples of transactions that we anticipate being used. Use **Navigation Option** to define this transaction. Note,

- Leave **Navigation Option** blank if the user should remain on the order transaction after completion.
- You can setup **Navigation Options** to transfer to the user to a tab other than the main tab.

If a user is allowed to complete this campaign's order's without selecting a package,

- Turn on **Allow Order Compl w/o Package** and
- Use **Order Compl w/o Package Label** to define the message to appear in the button used to complete such orders.

> **NOT_SUPPORTED**
>
> Refer to Campaigns Without Packages for examples of campaigns where you would enable the above fields.

Use **Long Description** to provide an overview of this campaign.

The grid at the bottom of the page summarizes this campaign's questions and miscellaneous fields. To change this information, press the adjacent go to button (which will transfer you to the Questions and Misc Fields tab). To add additional information, transfer to the Questions and Misc Fields tab and press the + button.

Campaign - Questions and Misc Fields

This page is used to define the following type of information:

- An order's campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.).
- An order's campaign can also be setup to capture additional fields to be updated on the order's person / account / premise when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken.
- An order's campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service.

> **NOT_SUPPORTED**
>
> Refer to Designing Campaigns and Packages for information describing how to design this type of information for your campaigns.

Copyright © 2026, Oracle and/or its affiliates.

Open this page using **Sales & Marketing, Campaign** and navigate to the **Question & Misc Fields** tab.

Description of Page

> **Warning**
>
> This information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The **Questions & Misc Fields** scroll contains one row for each question / miscellaneous field captured on this campaign's orders. The following fields are used:

- **Prompt on Order** contains the prompt that is shown to the user when the order is taken. Refer to Determine The Properties Of Every Miscellaneous Field for examples.
- **Brief Description** is simply a brief description of the question / miscellaneous field.
- **Sort Sequence** controls the order in which the question appears on the order.
- **Applicability** controls whether the field is Optional , Required, or Only Applicable on Package.
  - A value of Required means the order taker must enter a valid value before packages are displayed for selection.
  - A value of Optional means the order taken can leave the field blank prior to package selection. You can make an Optional field Required on individual packages.
  - A value of Only Applicable on Package is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a package, you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- **Type of Response** controls where the field is stored when the order is completed and how it is validated. You have two options: Characteristic or Column.
  - If a field is a Characteristic, you must define which entity the characteristic value is to be stored on when the order is completed (Person, Account, Premise or Order). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
  - If a field's value resides in a Column (as opposed to a characteristic value), you must define the column reference. A field's **Column Reference** controls three things:
- How the field's value is retrieved. A field's value is retrieved when an existing person / account / premise is referenced on an order.
- How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
- How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Extract Column References From Miscellaneous Fields for more information about column references.

- If a field's value resides in a Column (as opposed to a characteristic value), you must also define whether the question / field is dependent on the order referencing an account or premise. You do this by populating **Question Dependency** with one of the following values:
- Select Must Have Account if this question / field should only appear if the order references an account (new OR existing).
- Select Must Have Premise if this question / field should only appear if the order references a premise (new OR existing).
- Select No Dependency if this question / field should always appear.

NOT_SUPPORTED

Refer to Determine Account / Premise Dependencies For Column References for more information.

- **Default Column Value** / **Default Char Value** controls whether the order taker is supplied with a default value when an order is created. Note well: for Optional and Only Applicable on Package fields, you can specify a different default value on each package.

Campaign - Algorithms

This page defines plug-in algorithms that are executed for orders associated with this campaign. Open this page using **Menu**, and then **Sales & Marketing**, and then **Campaign**, and then **Search** and navigate to the **Algorithms** page.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The grid contains **Algorithms** that control campaign-oriented functions. You must define the following for each plug-in algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each **System Event**. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

You can define algorithms for the following **System Events**:

System Event	Optional / Required	Description

Copyright © 2026, Oracle and/or its affiliates.

Campaign Eligibility	Required	A campaign's eligibility algorithms control two things: - Whether the campaign appears in the Applicable Campaigns Zone. This content zone suggests possible campaigns for the current customer displayed on control central. You might want to use this zone if you have different campaigns that are offered to different types of customers. Refer to Real-time Marketing of Additional Services to a Customer for an example of how this content zone can be used to "up sell" additional services. - Whether the campaign appears on an order's eligibility tree. Campaigns that appear in this tree are alternate campaigns that may be used on an order. Refer to Real-time Marketing of Services to a Prospect for an example of how an order's eligibility tree can contain alternate campaigns for an order. Refer to The Big Picture Of Campaign Eligibility for information about these algorithms.
Order Completion	Optional	When an order is completed for this campaign, algorithms plugged into this spot are called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when an order is completed for this campaign.

Campaign - Package

A package controls the various types of service agreements that will be created if the customer selects the package. This page summarizes all packages that may be selected on orders associated with this campaign. Open this page using **Menu**, and then **Sales & Marketing**, and then **Campaign**, and then **Search** and navigate to the **Package** page.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

Copyright © 2026, Oracle and/or its affiliates.

The grid contains an entry for every package associated with this campaign. To view or change a package, press the adjacent go to button (which will transfer you to Package - Main). To add a new package, use the **Campaign** context menu to transfer to Package - Main in add mode.

Maintaining Packages

After a user enters basic information about a customer on an order, they are shown the various packages that may be offered to the customer. A package controls the various types of service agreements that will be created if the customer selects the package.

If the customer elects to take a package, the system sets up / updates the "V" (along with all of the ancillary things that happen when service is initiated, e.g., field activities are created).

The topics in this section describe the package transaction.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Campaigns, Packages and Orders for general information about packages. Refer to The Big Picture Of Package Eligibility Rules for a discussion of how to control which packages can be used for different types of customers. Refer to Designing Campaigns and Packages for guidelines describing how to design your packages.

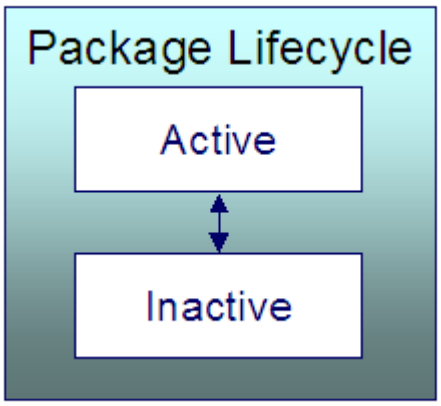
> **Note**
>
> **An order can be completed without selecting a package.** It is possible to use the order transaction to simply create / update persons and accounts. Refer to Campaigns Without Packages for more information.

- Lifecycle of a Package
- Package - Main
- Package - SAs To Create
- Package - Questions & Misc Fields
- Package - Eligibility
- Package - Instructions

Lifecycle of a Package

The following diagram illustrates the lifecycle of a package.

Copyright © 2026, Oracle and/or its affiliates.



Active An Active package may be selected on an order.

Inactive By Inactivating a package, you are preventing it from being selected on future orders.

Package - Main

This page is used to define basic information about a package.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Package**, and then **Search.**

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Use **Campaign** to define the campaign to which this package belongs.

Use **Description** and **Long Description** to describe the package.

Status defines the state of the package. Refer to Lifecycle of a Package for information about the valid values and the state transition rules.

The tree at the bottom of the page summarizes the information that appears on the remaining tabs. You can click on a hyperlink to transfer you to the appropriate tab with the relevant information displayed.

Package - SAs To Create

This information on this page controls the type of service agreement(s) / service point(s) that are created if this package is selected on an order.

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **SAs To Create** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

> **Warning**
>
> This following information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

The **SAs To Create** scroll contains one row for every service agreement that will be created if this package is selected on an order. The following fields are used:

- **CIS Division** and **SA Type** define the type of service agreement to be created.
- **Start Option** controls how the system will populate the new service agreement with default values. Refer to start options for more information.
- **SP Type** defines the type of service point that will be created / reused. Refer to Designing SP Types for more information.
- **Start Method Algorithm** controls how the system creates the new service agreement and service point. Refer to Determine The Method Used To Create Service Agreements for more information.
- **Create Proposal SA** controls if the new service agreement is a real or a proposal SA. This switch only appears if the quotation Contract Management module is not turned off. Refer to Proposal Service Agreements Must Exist for more information.

Package - Questions & Misc Fields

It is possible to design a campaign so that the user is prompted for additional information when a package is selected on an order. This page is used to define how the package overrides the questions and miscellaneous fields defined on the campaign.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Copyright © 2026, Oracle and/or its affiliates.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Package**, and then **Search** and navigate to the **Questions & Misc Fields** page.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Campaign contains a concatenation of basic information about the campaign.

> **Warning**
>
> The following information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

The grid contains one row for every Questions & Misc Field defined on the **Campaign**. You need only modify this information if this package behaves differently than the campaign (e.g., a field is required on the package, but optional on the campaign). To override a field on this package, turn on the **Override** switch and change the appropriate fields. There are several restrictions in respect of what can be overridden as described below:

- If a question / miscellaneous field is Required on the campaign, no fields may be overridden on the package.
- If a question / miscellaneous field is Optional on the campaign, you may override all field values. However, if the package has been used on a Completed order, you may not change its **Applicability** to be Required.
- If a question / miscellaneous field is designated as only applicable on package on the campaign, you should designate its applicability as Required, Optional or Not Used On This Package. If the field is either Required or Optional on this package, you may also override the other field values. Please note that if the package has been used on a Completed order, you may not change its **Applicability** to be Required.

> **NOT_SUPPORTED**
>
> Please refer to the Description of Page that appears under Campaign - Questions & Misc Fields for a description of the fields on the page.

Package - Eligibility

This page is used to define the conditions under which a package can be selected on an order.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Package Eligibility and Determine Package Eligibility Rules for more information.

Copyright © 2026, Oracle and/or its affiliates.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Package**, and then **Search** and navigate to the **Eligibility** page.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

> **Warning**
>
> The following information is not intuitive; we strongly recommend that you follow the guidelines under The Big Picture Of Package Eligibility and Determine Package Eligibility Rules before attempting to enter this information.

The **Eligibility Criteria Group** scroll contains one entry for each group of eligibility criteria. The following fields may be defined for each group:

- Use **Sequence** to control the relative order in which the group is executed when the system determines if the package can be selected on an order (smaller numbers are executed before larger numbers).
- Use **Description** and **Long Description** to describe the criteria group.
- Use **If Group is True** to define what should happen if the eligibility criteria (defined in the following grid) return a value of True.
  - Choose Eligible if this package can be selected on the order.
  - Choose Ineligible if this package cannot be selected on the order.
  - Choose Check Next Group if the next criteria group should be checked.
- Use **If Group is False** to define what should happen if the eligibility criteria (defined in the following grid) return a value of False.
  - Choose Eligible if this package can be selected on the order.
  - Choose Ineligible if this package cannot be selected on the order.
  - Choose Check Next Group if the next criteria group should be checked.

The grid that follows contains the package's eligibility criteria. Think of each row as an "if statement" that can result in the related eligibility group being True or False. For example, you might have a row that indicates the customer is eligible for the package if their customer class is residential. The following bullets provide a brief description of each field on an eligibility criterion. Please refer to Defining Logical Criteria for several examples of how this information can be used.

- Use **Sequence** to control the order in which the criteria are checked.
- Use **Criteria Field** to define the field to compare:
  - Choose Field if you want to compare a response to a question / miscellaneous field to a given value. Push the adjacent search button to select the field.
  - Choose Algorithm if you want to compare anything other than a response to a question / miscellaneous field. Push the adjacent search button to select the algorithm that is responsible for retrieving the comparison value.
- Use **Criteria Comparison** to define the method of comparison:

Copyright © 2026, Oracle and/or its affiliates.

- Choose Algorithm if you want an algorithm to perform the comparison and return a value of True, False or Insufficient Data.
- Choose any other option if you want to compare the **Criteria Field Type** using a logical operator. The following options are available:

- Use >, <, =, >=, <=, <> (not equal) to compare the **Criteria Field Type** using standard logical operators. Enter the comparison value in the adjacent field.
- Use In to compare the **Criteria Field Type** to a list of values. Each value is separated by a comma. For example, if a field value must equal 1, 3 or 9, you would enter a comparison value of 1,3,9.
- Use Between to compare the **Criteria Field Type** to a range of values. For example, if a field value must be between 1 and 9, you would enter a comparison value of 1,9. Note, the comparison is inclusive of the low and high values.
  - The next three fields control whether the related logical criteria cause the eligibility group to be considered True or False:
    * Use **If True** to control what happens if the related logical criterion returns a value of True. You have the options of Group is true, Group is false, or Check next condition. If you indicate Group is true or Group is false, then the package will be judged Ineligible or Eligible based on the values defined above in **If Group is False** and **If Group is True**.
    * Use **If False** to control what happens if the related logical criterion returns a value of False. You have the options of Group is true, Group is false, or Check next condition. If you indicate Group is true or Group is false, then the package will be judged Ineligible or Eligible based on the values defined above in **If Group is False** and **If Group is True**.
    * Use **If Insufficient Data** to control what happens if the related logical criterion returns a value of "Insufficient Data". You have the options of Group is true, Group is false, or Check next condition. If you indicate Group is true or Group is false , then the package will be judged Ineligible or Eligible based on the values defined above in **If Group is False** and **If Group is True**.

Package - Instructions

This page is used to define special instructions that should be presented to the user when they select this package. This information appears on the Order - Package Confirmation page.

NOT_SUPPORTED

Refer to Determine If Additional Instructions Are Needed for more information.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Package**, and then **Search** and navigate to the **Instructions** page.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Copyright © 2026, Oracle and/or its affiliates.

The grid contains a row for each instruction to be presented to the user when they select a package. The following fields are used:

- Use **Sort Sequence** to control the relative order of the instruction.
- Use **Display Icon** to control the icon that prefixes the instruction. Refer to Display Icons for more information.
- Use **Description** to define the text of the instruction.

The Big Picture of Program Management

The program management functionality satisfies many diverse requirements. For example, you can use this functionality to:

- Market your products and/or services to customers and measure the success of your efforts.
- Encourage customers to enroll in various sales and marketing initiatives. For example, you can use program management to organize a campaign to encourage customers to participate in a conservation program, or to sign up for special contract options.
- Set up marketing surveys and record your customer's responses.
- Assign representatives to accounts.
- And more…

The topics in this section provide more information about the program management functionality.

- The Big Picture of Initiatives, Leads, and Representatives
- Maintaining Initiatives
- Maintaining Leads
- Maintaining Sales Representatives
- Maintaining Third Party Representatives

The Big Picture of Initiatives, Leads, and Representatives

Three objects are at the heart of the program management functionality: **Initiatives**, **Leads**, and **Representatives.**

- An **initiative** defines a structured campaign that outlines the efforts to encourage customers to participate in a program targeted at a specific market segment.
- A **lead** is an instance of an initiative for targeted customers.
- There are two types of Representatives:
  - A **Sales Representative** is a user or group of users who can be assigned to follow up or manage leads. This representative is also responsible to sell or encourage the customer to participate in a program.
  - A **Third Party Representative** is a third party (typically external) who can receive assignments to perform specific tasks within a lead; such as, undertaking a visit to a property to perform an inspection, and installation work)

The topics in this section provide background information about these objects.

Copyright © 2026, Oracle and/or its affiliates.

- The Big Picture of Initiatives
- The Big Picture of Leads
- The Big Picture of Sales Representatives
- The Big Picture of Third Party Representatives

The Big Picture of Initiatives

An initiative defines a structured campaign designed to encourage customers to participate in a program. An initiative can:

- Target a market segment by specifying criteria that must be satisfied before leads are generated for customers.
- Ensure that customers are still eligible to participate in a program at sign up or enrollment time.
- Define a structured marketing effort for the events or actions that should take place during the life of a lead to encourage a customer to participate in the program.
- Define whether the lead is related to other objects (such as, the premise) in addition to the account.
- Contain sub-initiatives. Together, an initiative and its subsidiary initiatives form an initiative hierarchy. This supports reporting purposes and the analysis of the success of different market segments and marketing efforts.

Refer to Maintaining Initiatives for more information.

- About Initiative Hierarchies
- The Components of an Initiative
- The Lifecycle of an Initiative
- Initiative Information Can Be Overridden

About Initiative Hierarchies

Initiatives form an initiative hierarchy when one initiative is configured as a sub-initiative to a parent initiative. An initiative hierarchy can be used when a program is targeted at several market segments, or when different market segments each require a unique marketing effort to be executed. Initiative hierarchies can be up to five levels deep.

The following is a general example of a multi-level initiative hierarchy:

- Level 1 could be a “strategy” initiative.
- Level 2 could contain multiple campaigns. Each campaign can have a different way of marketing the strategy.
- Level 3 could define multiple marketing efforts. Each effort can have a different method for marketing the campaign.
- Level 4 could contain market segments that are the targets of a marketing effort.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The terms used above are just for the example. Your implementation would define the actual terms used for the levels in an initiative hierarchy.

A **parent initiative** may be set up for reporting purposes, in which case you could choose to configure the initiative so that no leads are generated for it. For example, a “strategy” may not have eligibility criteria or lead event type profile defined against it. It may simply have sub-initiatives to define segments, and those segments could be the initiatives for which leads are generated. Reporting is able to be performed at the multiple levels in order to summarize the effectiveness of the sub-initiatives.

The Components of an Initiative

The topics in this section provide information about the components of an initiative.

- About Initiative Eligibility Criteria
- About Related Object Criteria
- Examples of Initiative Criteria
- About the Lead Event Type Profile
- About the Initiative Log

About Initiative Eligibility Criteria

Each initiative can define its target market by specifying the criteria that must be satisfied before a customer is invited to participate in a program. It can also confirm that the customer is still eligible to participate in a program at sign up or enrollment time. An initiative may have zero, one, or many eligibility criteria. Each customer is subjected to these criteria to determine whether they will have a lead generated for them for the initiative and/or whether they are still eligible to sign up or enroll into a program.

Designing these criteria can be easy or time-consuming; it all depends on the complexity of your business rules. The base product provides a number of initiative eligibility criteria types. Most of them perform comparisons based on a specific field; for example, one of the criteria types checks whether an account has submitted a rebate claim for a specific conservation program. Some types of eligibility criteria could simply check whether or not a specific condition is true. For instance, a base-supplied eligibility criteria business object checks that there is no active credit and collection process for the account.

Other types of eligibility criteria perform comparisons with scalar values (such as, a number of days, number of kwh) or with derived values (such as, attributes of an account). The base product provides a Freeform Criteria business object that can be used for these purposes. This business object provides flexibility in that the value to supply in the comparison field is driven by a Derive Value Algorithm Type. A number of these algorithm types are delivered with this business object. Your implementations can add to these algorithm types as needed.

Your implementation can decide to use specific criteria business objects (base-supplied and/or implementation-specific), extend the base-supplied Freeform Criteria business object, or a mixture of both. Refer to Designing Initiative Eligibility Criteria for more details.

To set up an eligibility criterion, you must define the following:

- When the eligibility criterion should be executed - also called criteria execution point

Copyright © 2026, Oracle and/or its affiliates.

- The field to be compared
- The comparison method

For the **criteria execution point**, the options include:

- Prior to a lead being generated for a customer
- Prior to the customer being signed up or enrolled into a program
- Both of the above

For the **field to be compared**, you can execute an algorithm to:

- Retrieve a field value or values from someplace else in the system (such as, the customer class or types of services associated with the customer)
- Derive a value based on some data from someplace else in the system (such as, the number of days since the last lead was created for the customer)

For the **comparison method**, you select the operator to perform the comparison on the field to be compared. The result from the algorithm will be TRUE, FALSE, or INSUFFICIENT DATA. Each of these results should specify the next action. The valid actions include:

- Indicate that the customer is eligible
- Indicate that the customer is not eligible
- Check the next condition
- Issue an error (this option is only available for Insufficient Data)

Refer to Defining Eligibility Criteria for an Initiative for more information about setting up eligibility criteria on an initiative.

About Related Object Criteria

Each initiative can define its target market by specifying the criteria that must be satisfied before a customer is invited to participate in a program (see About Initiative Eligibility Criteria for more information). An initiative may also be targeted at a specific entity (or related object) for that customer; for example, assume a customer has two properties in two different states - one in State A and one in State B. The customer may only be invited to participate, or be eligible, for a program for their specific property located in State A. In this situation, each eligible customer may then be subjected to Related Object Criteria following the execution of Initiative Eligibility Criteria to determine which of their properties are eligible to participate in a program (meaning, have one lead generated just for the account if one of their properties meets the criteria or a lead generated for each of the customer's eligible properties).

An initiative defines the related object(s) to associate with leads via the Object Nomination algorithm on the initiative business object. The lead generation process executes the object nomination algorithm after executing initiative eligibility criteria. The base-supplied Initiative parent business object (C1–Initiative) is configured with an object nomination algorithm that retrieves the premises that are associated with the account's active and pending start service agreement's service points. The algorithm applies the related object criteria (configured on the initiative) to each premise and if the premise is deemed eligible, it is returned as a related object for the lead.

Designing related object criteria can be easy or time-consuming. It all depends on the complexity of your business rules. The base product provides examples of related object criteria that looks for one or more premises to associate with a lead. One example checks for premises with a specific service point type. Another looks for a premise using its geographic value. For additional ways to find premises to link to a lead, implementations can either add specific related object criteria business objects (similar to the base samples) or extend the

Copyright © 2026, Oracle and/or its affiliates.

base-supplied Freeform Premise Criteria business object. The Freeform Premise Criteria business object provides flexibility in that the value to supply in the comparison field is driven by a Derive Value Algorithm Type. A number of these algorithm types are delivered with this business object. Your implementation can add to these algorithm types as needed. Refer to Designing Related Object Criteria for more details.

When you set up a related object criterion, you must define two things:

- The field to be compared.
- The comparison method.

For the **field to be compared**, you can execute an algorithm to retrieve a field value or values from someplace else in the system; for example, retrieve a premise's service point types for which the customer has service.

For the **comparison method**, select the operator to perform the comparison on the field to be compared. The result from the comparison will be TRUE, FALSE, or INSUFFICIENT DATA. Each of these results should specify the next action. The valid actions include:

- Indicate that the related object is eligible
- Indicate that the related object is not eligible
- Check the next condition
- Issue an error

Refer to Defining Related Object Criteria for an Initiative for more information.

Examples of Initiative Criteria

The following is an example of how to set up initiative criteria.

Initiative: Smart Rate Program For Residential Customers

The program targets summertime pricing for residential customers. Customers are required to reduce energy use between 2 PM and 7 PM for no more than 15 summer days, which are referred to as smart days. A higher rate is charged during this time period. A discounted rate is applied for all other times during the season.

Initiative Eligibility Criteria

To be eligible for this initiative, the account must be a residential account with an average daily usage that is more than 10 kWh, not currently on a smart rate program, and is on either the ERES1 or ERES2 rate schedule. The account must also not have any active credit and collection process.

Sequence	Description	Criteria Field Algorithm	Operator	Comparison Value	If True	If False	If Insufficient Data	Criteria Execution Point
10	Residential customers	**Retrieve Customer Class** (Define an algorithm instance for base algorithm type C1-ICGETACEL, Retrieve account attribute)	=	R	Check next condition	Not Eligible	Not Eligible	Both (lead generation and enrollment)

Copyright © 2026, Oracle and/or its affiliates.

Sequence	Description	Criteria Field Algorithm	Operator	Comparison Value	If True	If False	If Insufficient Data	Criteria Execution Point
20	Average daily usage > 10 kWh	**Retrieve average daily kWh usage** (Define an algorithm for base algorithm type C1-ICGETAVSQ, Retrieve average service quantity)	>	10	Check next condition	Not Eligible	Not Eligible	Both (lead generation and enrollment)
30	Not on a smart rate program	**Retrieve all contract options** (Use base algorithm C1-ICGETCOPT)	=	295076574659 (Contract option for smart day events)	Not Eligible	Check next condition	Not Eligible	Both (lead generation and enrollment)
40	Account is current	**Check if Account Is Current** (Use base algorithm C1-ICCHKACCU)	=	true	Check next condition	Not Eligible	Not Eligible	Both (lead generation and enrollment)
50	Active service on eligible rate	**Retrieve all active rates** (Use base algorithm C1-GETACTVRT)	IN	ERES1, ERES2	Eligible	Not Eligible	Not Eligible	Both (lead generation and enrollment)

Related Object Criteria

If the account is eligible, the premise(s) associated with the account's active and pending start service agreement's service points of type E-RES will be linked to the lead as related object(s).

Sequence	Description	Criteria Field Algorithm	Operator	Comparison Value	If True	If False	If Insufficient Data
10	Premises with E-RES (electric residential) service points	**Retrieve Premise SP Types** (Base algorithm C1-ICPRMSPTY)	=	E-RES	Eligible	Not Eligible	Not Eligible

About the Lead Event Type Profile

Each initiative can define a structured marketing effort for the events or actions that should take place during the life of a lead to encourage a customer to participate in a program. Each event or action is defined by a Lead Event Type. For example, one event could be that a letter or an email is sent to a customer. The collection and sequencing of lead event types defined on an initiative is referred to as the Lead Event Type Profile.

Refer to Creating Initiatives and Maintaining Initiatives for more information on setting up and maintaining the Lead Event Type Profile on an initiative.

Copyright © 2026, Oracle and/or its affiliates.

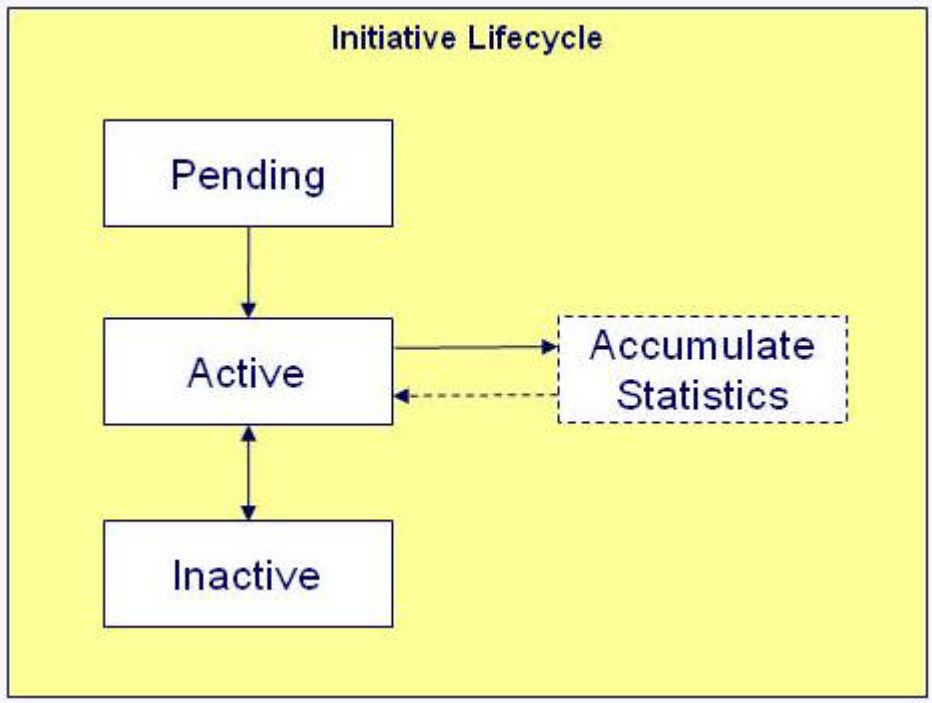
About the Initiative Log

Every initiative has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When an initiative is created.
- When an initiative's status changes (e.g. transitions to the active, accumulate statistics, inactive states).
- When lead related background processes are submitted for execution.
- When users want to manually add log entries as desired.

The Lifecycle of an Initiative

The following diagram shows the possible lifecycle of an Initiative:



Pending state: An initiative starts out in a pending state.

Active state: An initiative is transitioned manually by a user to the active state. Leads can only be generated for an initiative in an active state.

Accumulate Statistics state: A user or the system may transition an active initiative to a periodically transitioning accumulate statistics state. During this state, statistics for the initiative are accumulated prior to returning back to the active state.

Inactive state: An initiative may be transitioned manually by a user to the inactive state when they want to deactivate the initiative either temporarily or permanently. A user can manually place the initiative back to the active state from the inactive state.

Refer to Maintaining Initiatives for more information.

Copyright © 2026, Oracle and/or its affiliates.

Initiative Information Can Be Overridden

"Initiative info" is the concatenated string of information that summarizes an initiative throughout the system. The base-package logic constructs this string by concatenating the following information:

- Initiative description
- Status of the initiative
- Start date and end date (if available) of the initiative

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on the Initiative's maintenance object.

The Big Picture of Leads

A lead is an instance of an initiative for a targeted customer. A lead may contain events or actions that are there to encourage a customer to participate in a program.

Refer to Maintaining Leads for more information about working with Leads.

- How Leads Are Created
- The Components of a Lead
- The Lifecycle of a Lead
- Advanced Analysis of Leads
- Lead Information Can Be Overridden
- Leads Are Highlighted As Alerts
- How and When Are Leads Activated
- How are Leads Completed
- How Are Leads Discarded
- Sign Up Options Can Be Overridden
- The Big Picture of Lead Events

How Leads Are Created

Leads are created as follows:

- The system can generate a lead, or leads, for each customer that satisfies the initiative eligibility criteria and/or related object criteria that is defined on an initiative.
- Users can manually create a lead at their discretion.

Refer to section Creating Leads from an Initiative and Manually Creating a Lead for more information.

The Components of a Lead

A lead has two components: A **Lead Event** and a **Lead Log.**

Lead Event

Copyright © 2026, Oracle and/or its affiliates.

A lead has one or more lead events. These events are the actions designed to encourage a customer to participate in a program. For example, one event could be that a letter or an email is sent to a customer.

Refer to The Big Picture of Lead Events for more information.

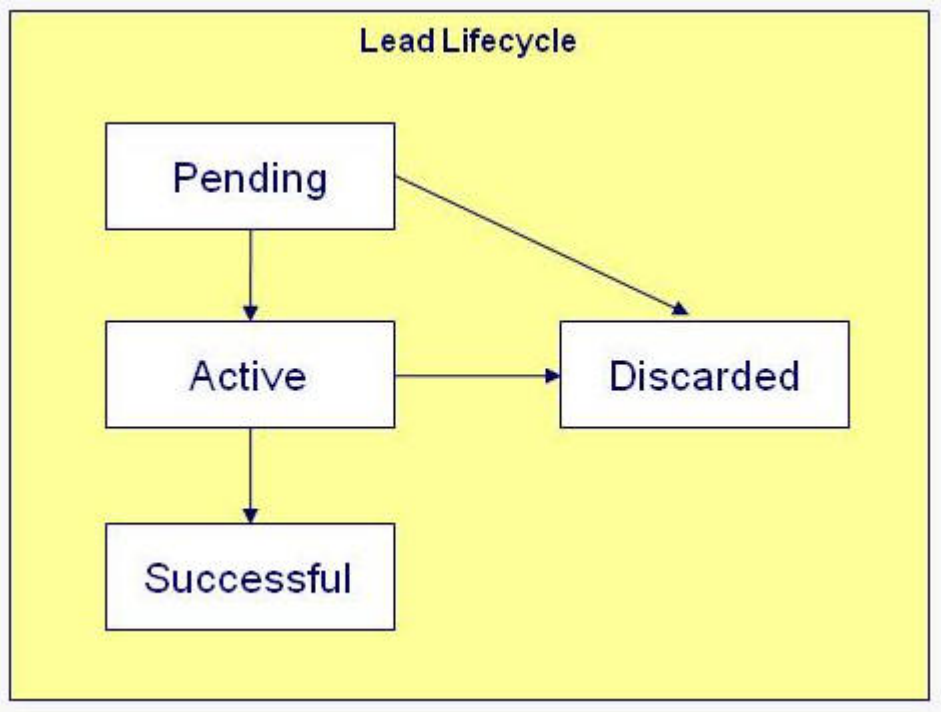
Lead Log

Every lead has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When a lead is created
- When a lead's status changes (e.g. transitions to the active, successful or discarded state)
- When a lead event executes
- Users can manually add log entries as desired.

The Lifecycle of a Lead

The following diagram shows the possible lifecycle of a Lead:



Pending state: Leads can be created in either the **Pending** or **Active** state. By generating them in the **Pending** state, you can verify the targeted customers before actively marketing the initiative.

Active state: Leads can be created in either the **Pending** or **Active** state. When a lead is created or transitioned to the **Active** state, lead events are created to encourage the customers to participate in the initiative. For example, one event could be that a letter or an email is sent to a customer. The lead events and the sequence in which they occur are defined on the initiative's lead event type profile. The lead remains in the active state while it is being actively marketed to the customer.

Successful state: The system infers that a lead has succeeded by the creation of other data in the system, at which time the lead transitions to the **Successful** state. For example:

Copyright © 2026, Oracle and/or its affiliates.

- If an active lead for a customer is related to a rebate claim conservation program, the lead is determined to be successful when that rebate claim is filed for the customer.
- If an active lead for a customer is related to a contract option, the lead is determined to be successful when the customer's service agreement is linked to that contract option.
- If an active lead for a customer is related to a campaign, the lead is determined to be successful if a completed order exists for the customer for the campaign.
- If an active lead is assigned to a third party representative (to perform one or more tasks), the lead is determined to be successful if the Notify Third Party lead event is completed.

Discarded state: An unsuccessful lead transitions to the **Discarded** state after a certain amount of time has passed, as defined on the initiative. If the initiative does not specify a time period, the lead remains in the active state until a user manually discards it.

Refer to Maintaining Leads for more information.

Advanced Analysis of Leads

Leads can be sent to an external system (such as Oracle DataRaker) for advanced analysis prior to activating the lead.

The base product provides Advanced Analysis System Lead (C1–AdvancedAnalysisLead) business object for this purpose. Its lifecycle includes steps to send the lead to the external system and activate/discard the lead based on the result of analysis..

Additional configuration is needed in Program Management master configuration to enable external analysis/validation. See master configuration embedded help for more details.

- The Lifecycle of a Lead with Advanced Analysis

The Lifecycle of a Lead with Advanced Analysis

The following describes the states of a Lead that is sent to an external system for advanced analysis prior to activation..

(Refer to the C1–AdvancedAnalysisLead business object to view the lifecycle diagram.)

Pending state: Leads can be created in either the **Pending** or **Ready for Extract** state. By generating them in the **Pending** state, you can verify the targeted customers before actively marketing the initiative.

Ready for Extract state: At this state, leads are ready to be sent to the external system for analysis. A monitor batch process is defined on this state, to ensure that leads in this state are transitioned only when the batch process runs.

Waiting for Analysis state: When a lead is transitioned to this state, an outbound message containing lead information is sent to the external system.

Error state: The lead is transitioned to this state if the system is unable to send the outbound message. Retry logic on this state will attempt to resend the lead to the external system. Leads in this state can also be **Discarded** if needed.

Active state: When a lead completes external analysis, the External Validation and External Activation Priority flags on the lead are updated. An inbound web service sends back the lead and transitions the lead into the **Active** state. An enter algorithm evaluates the External Activation Priority flag. If the lead is marked as being a priority for activation (based on settings in master configuration), the lead stays in the **Active** state. Otherwise, the lead is **Discarded**.

Copyright © 2026, Oracle and/or its affiliates.

When a lead is in the **Active** state, lead events are created to encourage the customers to participate in the initiative. For example, one event could be that a letter or an email is sent to a customer. The lead events and the sequence in which they occur are defined on the initiative's lead event type profile. The lead remains in the active state while it is being actively marketed to the customer.

Successful state: The system infers that a lead has succeeded by the creation of other data in the system, at which time the lead transitions to the **Successful** state. For example:

- If an active lead for a customer is related to a rebate claim conservation program, the lead is determined to be successful when that rebate claim is filed for the customer.
- If an active lead for a customer is related to a contract option, the lead is determined to be successful when the customer's service agreement is linked to that contract option.
- If an active lead for a customer is related to a campaign, the lead is determined to be successful if a completed order exists for the customer for the campaign.
- If an active lead is assigned to a third party representative (to perform one or more tasks), the lead is determined to be successful if the Notify Third Party lead event is completed.

An enter algorithm on this state sends an outbound message, to notify the external system about lead success.

Discarded state: Leads that are evaluated as non-priority for activation are transitioned to this state. Unsuccessful leads also transition to this state after a certain amount of time has passed, as defined on the initiative. If the initiative does not specify a time period, unsuccessful leads remain in the **Active** state until a user manually discards it.

Refer to Maintaining Leads for more information.

Lead Information Can Be Overridden

"Lead info" is the concatenated string of information that summarizes a lead throughout the system. The base-package logic constructs this string by concatenating the following information:

- Initiative description
- Status of the lead
- Creation date/time of the lead.

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on the Lead's maintenance object.

Leads Are Highlighted As Alerts

If you plug-in the appropriate alert algorithm (C1-OPENLDALR) on the Installation record, alert(s) will be shown for open leads associated with the person, account and premise in context. A separate alert is shown for each unique Lead that is in a non-final state. Each alert message shows the Lead's info string.

How and When Are Leads Activated

Leads are activated as follows:

- The system can generate a lead for each customer that satisfies the initiative eligibility criteria and/or related object criteria that is defined on an initiative. The leads can be

Copyright © 2026, Oracle and/or its affiliates.

created in either a Pending or Active state. These leads can then be activated by a background process when required by the user.

- Users can manually activate a lead at their discretion.

Refer to section Creating Leads from an Initiative and Maintaining a Lead for more information.

How are Leads Completed

The system infers that a lead has succeeded by the creation of other data in the system, at which time the lead transitions to the **Successful** state. For example:

- If an active lead is related to a rebate claim conservation program, the lead is determined to be successful when that rebate claim is filed for the customer.
- If an active lead is related to a contract option, the lead is determined to be successful when the customer's service agreement is linked to that contract option.
- If an active lead for is related to a campaign, the lead is determined to be successful if a completed order exists for the customer for the campaign.
- If an active lead is assigned to a third party representative, to perform one or more tasks, the lead is determined to be successful if the Notify Third Party lead event is completed.

Users can also manually transition a lead to the Successful state at their discretion. Refer to Maintaining a Lead for more information about working with Leads.

How Are Leads Discarded

When leads are sent to an external system for analysis prior to lead activation, a possible result of the analysis is that the lead is flagged as non-priority for activation. When this occurs, the lead is discarded.

Unsuccessful leads transition to the Discarded state after a certain amount of time has passed, as defined on the initiative. The initiative's lead event type profile can include an event that automatically discards a lead after a designated number of days. If the initiative does not specify this auto-discard event, the lead remains in the active state until a user manually discards it.

Some types of lead events can also trigger a lead to be discarded when specific conditions are met; for example, a lead that is assigned to a third party representative can be automatically discarded when the third party representative finds that the account is ineligible for the initiative. Refer to The Lifecycle of a Third Party Representative for more details.

In addition, automatic lead discards capture a lead discard reason. Lead event types that support auto-discard define their respective default lead discard reasons. Refer to Setting Up Lead Event Types and Maintaining a Lead for more information

Sign Up Options Can Be Overridden

The lead's Sign Up special action button performs the following:

- Displays a Sign Up UI map (if one exists) to capture additional inputs for sign up processing.
- Executes a Sign Up service script to process the sign up.
- Displays a confirmation map if the sign up is successful.

Copyright © 2026, Oracle and/or its affiliates.

The Sign Up UI map and Sign Up service script can be overridden via options on the initiative's business object. Refer to Overriding Initiative Sign Up Options for more details.

The Big Picture of Lead Events

A lead has one or more lead events. These events are the actions designed to encourage a customer to participate in a program. For example, one event could be that a letter or an email is sent to a customer.

You define the number and type of events to be initially inherited by a lead by configuring the Lead Event Type Profile on an initiative. Refer to About the Lead Event Type Profile for more information.

A lead event can include one or more steps. The lead event type defines a list of step types, if applicable.

The topics in this section provide additional information about lead events.

- How are Lead Events Created
- The Components of a Lead Event
- The Lifecycle of a Lead Event
- The Lifecycle of a Third Party Lead Event
- Lead Event Information Can Be Overridden
- Third Party Lead Event Step Information Can Be Overridden
- How and When Lead Events Are Activated
- Activating Events Should Add a Log Entry
- How Lead Events Are Cancelled

How are Lead Events Created

If a lead is created in a Pending state, there are no lead events associated with the lead. It is only when leads are created in or transitioned to the Active state, that lead events are created for the lead. The initial number of events and when the events shall occur are inherited from the lead's initiative.

Some types of events can also get created as a result of special actions on the lead. For instance, when a lead process requires a third party representative to perform one or more tasks after signup, the Sign Up action on the Lead portal automatically creates the Notify Third Party lead event.

The Components of a Lead Event

A lead event can have one or more steps. The lead event's business object defines whether or not steps are applicable to that type of event. The corresponding lead event type defines the valid step types. For an example, refer to the Third Party Lead Event (C1-ThirdPartyLeadEvt) and Send Notification to Third Party Lead Event Type (C1-LETypeNotifyThirdParty) business objects provided in base.

Lead Event Log

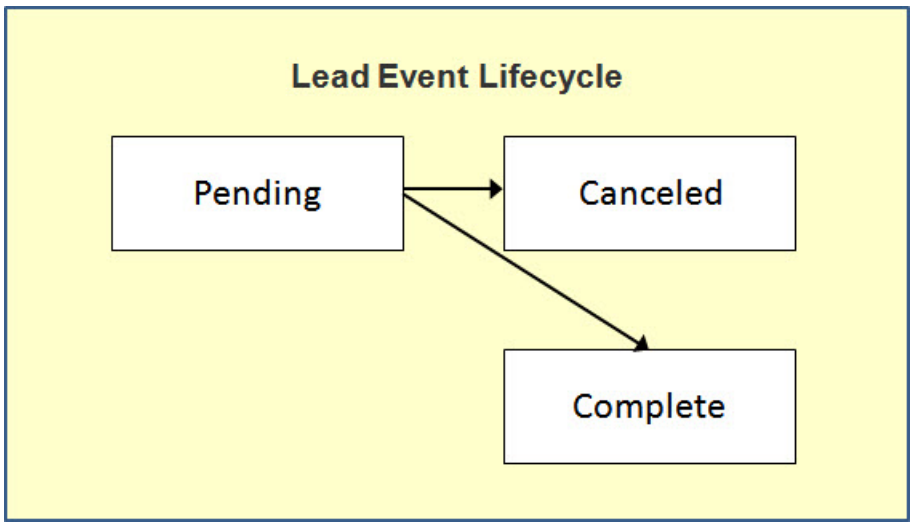
Every lead event has a log holding its history. Entries are added to the log when meaningful events occur, for example:

Copyright © 2026, Oracle and/or its affiliates.

- When a lead event is created.
- When a lead event's status changes (e.g. transitions to the completed or cancelled state).
- Details of the content sent (if applicable).
- Users can manually add log entries as desired.

The Lifecycle of a Lead Event

The following diagram shows the possible lifecycle of a Lead Event business object:



Pending state: Lead events are initially created in the Pending state. A lead event can take a number of paths after it is created; it all depends on how you've configured the system.

Complete state: A lead event contains a date when it should be activated. This is referred to as its trigger date. On this date, the system invokes the algorithm associated with the lead event to execute the event or action. A user may complete a pending event at their discretion.

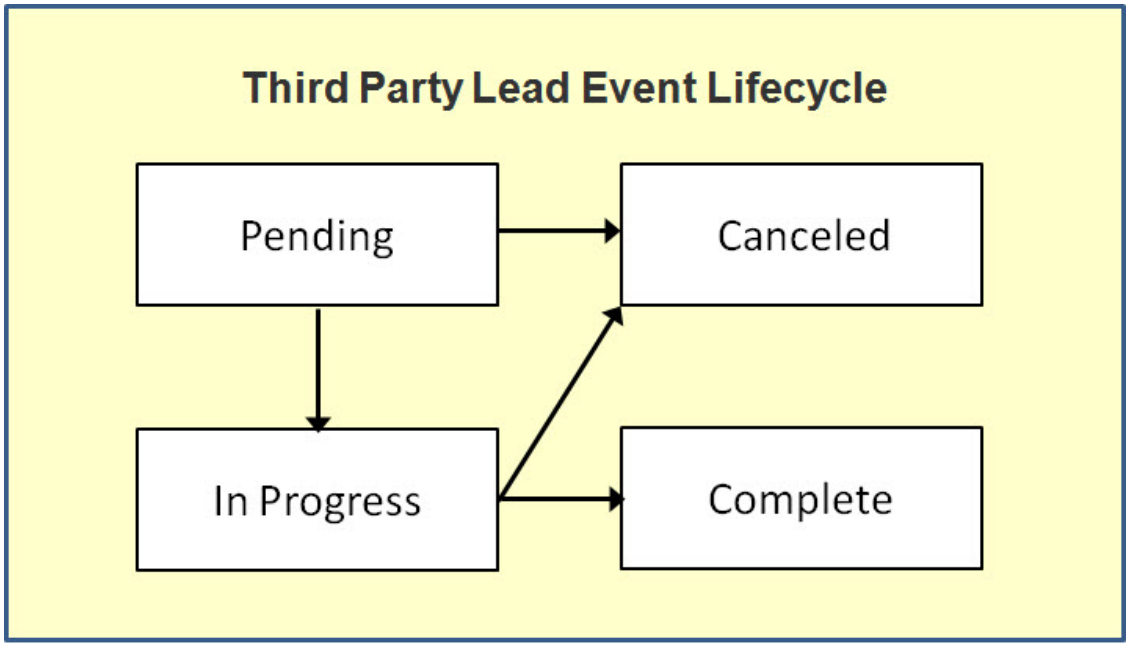
Cancelled state: A pending event will be cancelled automatically by the system when a lead is canceled. A user may cancel a pending event at their discretion.

Refer to Maintaining Lead Events for more information.

The Lifecycle of a Third Party Lead Event

The following diagram shows the possible lifecycle of a Third Party Lead Event business object:

Copyright © 2026, Oracle and/or its affiliates.



Pending state: Third party lead events are initially created in the Pending state. A third party lead event can take a number of paths after it is created; it all depends on how you've configured the system.

In Progress state: A third party lead event contains a date when it should be activated. This is referred to as its trigger date. On this date, the system invokes the algorithm associated with the lead event to execute the event or action. The lead event is set to In Progress and the lead event is sent to the third party representative.

Complete state: A third party lead event may be completed based on updates received from the third party. In most cases, the third partly lead event is completed when all the event steps are complete. In addition, a user may complete an In Progress event at their discretion.

Cancelled state: A Pending or In Progress third party lead event will be cancelled automatically by the system when the lead is canceled. A user may cancel a Pending or In Progress event at their discretion.

Refer to Maintaining Third Party Lead Events for more information.

Lead Event Information Can Be Overridden

"Lead Event info" is the concatenated string of information that summarizes a lead event throughout the system. The base-package logic constructs this string by concatenating the following information:

- Lead event type description
- Status of the lead event
- Trigger date of the lead event

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on the Lead Event's maintenance object.

Copyright © 2026, Oracle and/or its affiliates.

Third Party Lead Event Step Information Can Be Overridden

The **Initiative Leads** zone on the **Third Party Representative** portal shows an event step information string for **Last Complete Step** and **Next Pending Step**. A base-package service script constructs this string by concatenating the following information:

- Step type description
- Intended Work Date/Time
- Completion Date/Time

You can override this by specifying an **Override Step Information Script** on the Program Management Master Configuration.

See Program Management Master Configuration for more details.

How and When Lead Events Are Activated

A lead event contains the date it should be activated; this is referred to as its trigger date. On this date, the Lead Event Periodic Monitor Process (a background process (C1-LDEVT)) invokes the algorithm associated with a lead event's event type. The algorithm, in turn, will decide on the state in which to leave the lead event.

If a user can't wait for the Lead Event Periodic Monitor Process real-time, they can click a button on Lead Event – Main to activate / trigger an event online in real-time. Refer to Maintaining Lead Events for more information.

You control how a lead event's trigger date is populated by configuring the Lead Event Type profile in an initiative. You indicate the number of days the lead event should be activated after the creation of the lead. Refer to Maintaining Initiatives for more information.

Activating Events Should Add a Log Entry

As described previously, a lead and lead event have a log holding a history of meaningful events in their lives. Most algorithms associated with a lead event's event type will add an entry to both the lead and lead event logs.

These log entries are more than just an audit trail as they also reference the objects that are created during execution. For example, if an algorithm creates a customer contact, the ID of the customer contact will be referenced on the log (and end-users will be able to drill down on the log entry to see the customer contact).

How Lead Events Are Cancelled

A pending lead event will be cancelled automatically by the system when the lead is canceled.

A user may cancel a pending lead event at their discretion. Refer to Maintaining Lead Events for more information.

A third party representative can also initiate the cancellation of a lead event. Refer to The Lifecycle of a Third Party Lead Event for more details.

Copyright © 2026, Oracle and/or its affiliates.

The Big Picture of Sales Representatives

A sales representative is a group or individual who can be assigned to follow up and manage leads.

Individual sales representatives can be members of one or more groups, and they can be an internal user or an external party.

Groups can be used to control which sales representatives are allocated to the leads for an initiative. For example, you can set up a group whose sales representatives are responsible for industrial customers. Initiatives that target industrial customers can then be configured so that this group is specified as the only one whose sales representatives can be assigned the initiative's leads.

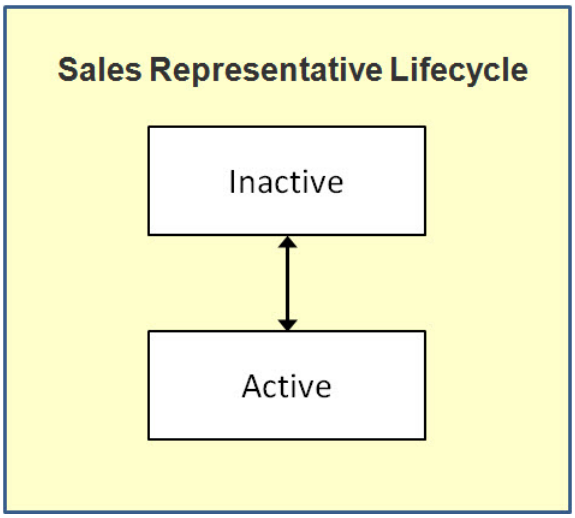
Because many reports will aggregate information for all sales representatives in a group, you may choose to set up a sales representative group just for reporting purposes.

The topics in this section provide additional information about Sales Representatives.

- The Lifecycle of a Sales Representative
- Sales Representative Log

The Lifecycle of a Sales Representative

The following diagram shows the possible lifecycle of a Sales Representative:



Active state: A Sales Representative starts out in an active state.

Inactive state: A Sales Representative may be transitioned manually by a user to the inactive state when they want to deactivate the Sales Representative either temporarily or permanently. A user can manually place the Sales Representative back to the active state from the inactive state.

Refer to Maintaining Sales Representatives for more information.

Copyright © 2026, Oracle and/or its affiliates.

Sales Representative Log

Every Sales Representative has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When a Sales Representative is created
- When a Sales Representative's status changes (e.g. transitions to the inactive state)
- Users can manually add log entries (you might want to think of these as "diary" entries) as desired.

The Big Picture of Third Party Representatives

A third party representative is similar to a sales representative except that they are typically not internal employees and they have a different set of responsibilities.

Lead-related tasks may be outsourced to a third party representative who is better equipped to perform the work. These tasks may include pre-installation visits, energy audits, equipment installation, and equipment recycling. An initiative's lead event type profile defines whether or not lead-related work is sent to a third party representative.

Work is typically sent to a third party representative when a customer signs up for an initiative. The base product provides an online Sign Up action on active leads. This action creates a Third Party Lead Event that triggers the third party notification.

The third party representative is also responsible for providing updates to the system about the assigned leads. These updates can include event status updates, updates to important date/time information (e.g., completion date/time) and comments. This feedback enables users of the system to monitor the progress of leads and provide any necessary customer service.

Refer to the The Lifecycle of a Third Party Lead Event for more details on third party lead events.

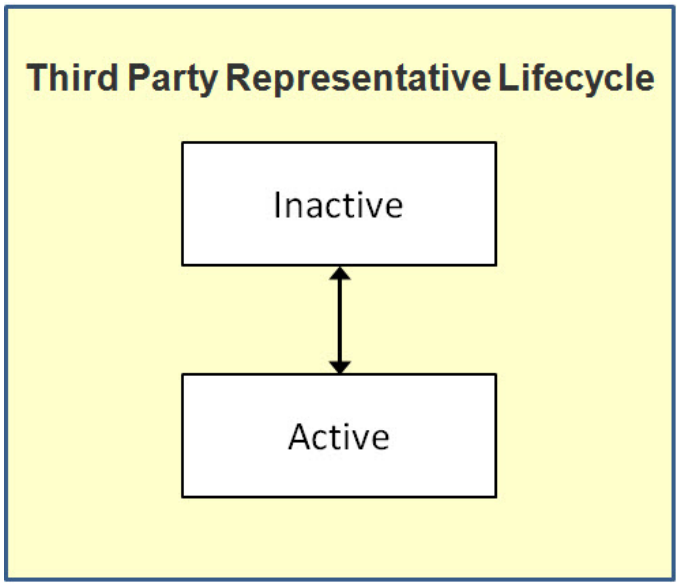
The topics in this section provide additional information about Third Party Representatives.

- The Lifecycle of a Third Party Representative
- Third Party Representative Log

The Lifecycle of a Third Party Representative

The following diagram shows the possible lifecycle of a Third Party Representative:

Copyright © 2026, Oracle and/or its affiliates.



Active state: A Third Party Representative starts out in an active state.

Inactive state: A Third Party Representative may be transitioned manually by a user to the inactive state when they want to deactivate the Third Party Representative either temporarily or permanently. A user can manually place the Third Party Representative back to the active state from the inactive state.

Refer to Maintaining Third Party Representatives for more information.

Third Party Representative Log

Every Third Party Representative has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When a Third Party Representative is created.
- When a Third Party Representative's status changes (e.g. transitions to the inactive state).
- Users can manually add log entries as desired.

Maintaining Initiatives

An initiative defines a structured campaign that outlines the efforts to encourage customers to participate in a program targeted at a specific market segment.

Refer to The Big Picture of Initiatives for more information about initiatives.

- Initiative Portal
- How To Perform Common Initiative Functions

Initiative Portal

You use the Initiative portal to maintain initiatives.

The **Main** page contains the following zones:

- **Initiative:** Displays the basic information about the initiative.

Copyright © 2026, Oracle and/or its affiliates.

- **Initiative Eligibility Criteria Overview:** Displays initiative eligibility criteria associated with the initiative.
- **Related Object Criteria Overview:** Displays criteria for finding one or more related objects (such as, a premise) to link to a lead.
- **Initiative Hierarchy:** Displays the entire hierarchy of initiatives that the selected initiative is a part of.
- **Initiative Leads:** Displays the leads that are associated with the initiative.
- **Lead Processing:** Allows you to submit the lead generation or lead disposition batch process.

The **Statistics** page contains the following

- **Lead Summary by Representative:** Displays the summary of leads for the initiative hierarchy, grouped by representatives.
- **Initiative Leads:** Displays the leads that are associated with the initiative.
- **Lead Outcome Graph:** Displays the number of leads linked to the initiative hierarchy that entered the final state within the last 24 months.
- **Lead Statistics:** Displays statistics for the leads associated with the initiative.
- **Lead Monthly Statistics:** Displays statistics for the leads associated with the initiative on a monthly basis.
- **Unassigned Leads:** Displays leads that are not yet assigned to a sales representative and are not in a final state. (**Note:** Leads can be assigned to a sales representative in bulk.)

How To Perform Common Initiative Functions

The topics in this section describe how to perform common initiative functions from the Initiative portal.

- Creating Initiatives
- Maintaining an Initiative
- Viewing Initiative Hierarchy
- Creating Leads from an Initiative
- Managing Leads from an Initiative
- Viewing Initiative Statistics

Creating Initiatives

Use this procedure to create a new Initiative.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Add.**
2. Select the business object that will define the initiative and click **OK.**
3. Enter the initiative's name, (short) description, detailed description, start date, and end date.
4. Specify if the initiative is a master or a subsidiary initiative.
    - If the initiative is a **Subsidiary Initiative**, search for and select the initiative that is the parent to this initiative.

Copyright © 2026, Oracle and/or its affiliates.

5. Specify whether the master initiative will participate in demand-side analytics processing through Contract Option Participation Analysis.
6. Specify whether or not leads can be generated for the initiative.
    - If leads can be generated, complete the options regarding communication channel and related object creation.
7. In the Third Party Notification section, specify the Lead Event Type to use to notify a third party. This lead event type is used when creating a third party notification lead event, if applicable, when a customer signs up or enrolls into a program.
8. In the Lead Reassignment Notification section, indicate how the system should respond if a lead is reassigned to another sales representative.
9. In the Statistics Notification section, specify how often, in hours, the statistics for the initiative should be updated.

    A background process accumulates the statistics for an initiative when enough time has elapsed since the statistics were last updated. A user may also manually initiate the process to update the statistics for an initiative.
10. If this initiative will have leads, add lead event types in the Lead Event Type Profile section to define the events or actions that will occur during the lifetime of the lead to encourage the customer to participate in the initiative. To add lead event types in the Lead Event Type Profile section:
    a. Assign an **Event Sequence** to each list entry.
    b. Search for and select a **Lead Event Type** to indicate the type of lead event to generate for each list entry (for example send customer an email or create a customer contact).
    c. Specify the number of **Days After Lead Creation** that the related lead event will be triggered.
11. Click **Save.**

Maintaining an Initiative

The topics in this section describe procedures for maintaining an initiative:

- Allowable Record Actions on an Initiative
- Defining Eligibility Criteria for an Initiative
- Defining Related Object Criteria for an Initiative

Allowable Record Actions on an Initiative

Use this procedure to maintain an initiative using standard record actions.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain. The selected initiative opens in the Initiative portal.
5. Click the appropriate record action button:
    - **Edit** – Click the **Edit** button to edit the main attributes of an initiative. When you click **Save**, the system updates the record.

Copyright © 2026, Oracle and/or its affiliates.

- **Duplicate** – Click the **Duplicate** button to create a duplicate of the initiative being displayed. Before the new object is added to the database, a window appears asking you to enter the unique identifier of the new object. When you click **OK**, the system creates the record by copying the original record.
- **Activate** – Click the **Activate** button to transition a pending or inactive initiative to the active state.
- **Deactivate** – Click to **Deactivate** button to transition an active initiative to an inactive state.
- **Refresh Statistics** - Click the **Refresh Statistics** button to manually initiate the accumulation of the statistics for the initiative.
- **Delete** – Click the **Delete** button and click **OK** to delete the initiative from the database. Before you delete an initiative, the system verifies that the initiative is not referenced anywhere in the system. If it is, you cannot delete it. For example, if you attempt to delete an initiative, the system ensures the initiative has not been referenced on a lead, etc.

Defining Eligibility Criteria for an Initiative

Use this procedure to define eligibility criteria for an initiative. The following steps apply to each criterion.

Prerequisites: You must create an initiative that can have leads before you can define eligibility criteria for the initiative.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh**.
4. In the search results list, click the link for the initiative that you want to maintain. *(The selected initiative opens in the Initiative portal.)*
5. To define eligibility criteria for the initiative, click the **Add** link in the Initiative Eligibility Criteria Overview zone title bar. *(This zone is displayed for initiatives that can have leads.)*
6. Select the business object that will define the initiative criteria and click **OK**.

   The base package provides a number of initiative eligibility criteria business objects. Each business object defines its valid comparison algorithm(s), operator(s), and value(s). Refer to the maintenance maps of these business objects for details. For more information about eligibility criteria types, refer to About Initiative Eligibility Criteria .
7. Enter a sequence number, description, detailed description, and criteria execution point for the criterion.
8. Complete the Criteria Comparison section. Each business object defines the valid selections for: **Algorithm**, **Comparison Operator** and **Comparison Value**.

   Refer to the embedded help on the maintenance map for more details on these fields.
9. Complete the Comparison Results section. Specify the action for each possible comparison outcome - meaning, **If True**, **If False**, and **If Insufficient Data**.

   Refer to the embedded help on the maintenance map for more details on these fields.
10. Click **Save.**

Repeat steps 5 to 10 to add additional criterion for the initiative.

Copyright © 2026, Oracle and/or its affiliates.

Defining Related Object Criteria for an Initiative

Use this procedure to define related object criteria for an initiative. The following steps apply to each criterion.

Prerequisites: You must create an initiative that can have leads before you can define related object criteria for the initiative.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain. The selected initiative opens in the Initiative portal.
5. To define related object criteria for the initiative, click the **Add** link in the Related Object Criteria Overview zone title bar. *(This zone is displayed for initiatives that can have leads.)*
6. Select the business object that will define the initiative criteria and click **OK**.

   The base package provides a number of related object criteria types that attempt to link an account's premise(s) to the lead. Each initiative criteria business object defines its valid comparison algorithm(s), operators and value(s). Refer to the maintenance maps of these business objects for details. For more information on eligibility criteria types, refer to About Related Object Criteria .
7. Enter a sequence number, description, and detailed description for the criterion.
8. Complete the Criteria Comparison section. Each business object defines the valid selections for: **Algorithm**, **Comparison Operator**, and **Comparison Value**.

   Refer to the embedded help on the maintenance map for more details on these fields.
9. Complete the Comparison Results section. Specify the action to take for each possible outcome of the comparison - meaning, **If True**, **If False**, and **If Insufficient Data**

   Refer to the embedded help on the maintenance map for more details on these fields.
10. Click **Save**.

Repeat steps 5 to 10 to add additional criterion for the initiative.

Viewing Initiative Hierarchy

The Initiative Hierarchy zone on the Initiatives portal displays the hierarchy the selected initiative is a part of. By clicking on any initiative in the hierarchy, you will be navigated to the selected initiative.

Creating Leads from an Initiative

The topics in this section describe how to create leads from an Initiative.

- Bulk Creation of Leads
- Uploading Leads

Bulk Creation of Leads

Copyright © 2026, Oracle and/or its affiliates.

Use this procedure to create leads in bulk.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain.

   The selected initiative opens in the Initiative portal.
5. To create leads for the initiative, click **Lead Generation** on the Lead Generation and Disposition zone. This zone is displayed for initiatives in the active state, can have leads and allows a batch job to be submitted for the creation of leads.
   - Select the **Lead Status** to leave the created leads in. Valid values are Pending or Active. Creating leads in a pending state allows a user to confirm the lead population prior to activating these leads
   - Specify the **Number of Leads to Generate.** This may be used to create a sample set of leads to test the initiative's eligibility criteria.
   - Specify the **Email Address.** You may optionally specify an email address if you want to be informed when the batch job completes
6. Click **OK.**

Uploading Leads

> **Note**
>
> This functionality is to be used on small volume uploads.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain. The selected initiative opens in the Initiative portal.
5. To upload Leads for an initiative, click the **Lead Upload** link in the Initiative Leads zone title bar. This displays the Lead Upload page, which you can use to upload a CSV file containing account IDs for which leads will be generated.
6. Click **Submit.**

Managing Leads from an Initiative

The procedures in this section describe how to manage leads from an initiative.

- Viewing Initiative Leads
- Activating, Deleting, and Discarding Leads for an Initiative in Bulk

Viewing Initiative Leads

The Initiative Leads zone is displayed on both the Main page and the Statistics page.

Copyright © 2026, Oracle and/or its affiliates.

By default, the Initiative Leads zone is populated with all leads for the initiative (or initiative hierarchy group) for all sales representatives.

On the Statistics page, you can limit the set of leads using the Lead Summary by Representative zone. See Viewing Lead Summary by Representativefor details.

The following is displayed in the result set for each lead:

- Basic **Lead** information for the initiative's lead. You can navigate to the lead by clicking on the lead.
- **Customer Information** related to the lead. You can navigate to the account related to the lead by clicking on the customer information.
- The **Last Complete Event.** You can navigate to the lead event by clicking on the last complete event for a lead.
- For open leads, the **Next Pending Event.** You can navigate to the lead event by clicking on the next pending event for a lead.
- The **Sales Representative** shows the representative assigned to the lead.

Activating, Deleting, and Discarding Leads for an Initiative in Bulk

Use the following procedure to activate, delete or discard leads for an initiative.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain.

   The selected initiative opens in the Initiative portal.
5. Scroll, as needed, to the Lead Processing zone.
6. To activate, delete or discard leads for the initiative, click **Lead Disposition**. This allows a batch job to be submitted for the activation, deletion or discarding of leads.
   - Select the **Pending Lead Disposition** to indicate the action to take on pending leads. The possible actions are: Activate (pending leads), Delete (in case of erroneous generation), Ignore.
   - Select the **Active Lead Disposition** to indicate to action to take on active leads. The possible actions Ignore and Discard.
   - Select the **Discard Reason** to be used for any active leads that get discarded. The values are maintained on the Status Reason control table.
   - Specify the **Email Address.** You may optionally specify an email address if you want to be informed when the batch job completes
7. Click **OK.**

Viewing Initiative Statistics

The Statistics page displays statistical information about the leads of the current initiative in context. It shows the following:

- Lead counts by sales representative and status
- List of leads

Copyright © 2026, Oracle and/or its affiliates.

- Graph of completed and discarded leads over a period of time
- Graph of leads by status
- Graph of active leads per sales representative
- Graph of open, completed and discarded leads over a period of time
- Graph of the top discard reasons over a period of time

The following sections describe these zones. You can also refer to each zone's embedded help for more details.

- Viewing Lead Summary by Representative
- Graphs and Statistics for an Initiative
- Displaying and Assigning Unassigned Leads

Viewing Lead Summary by Representative

The Lead Summary by Representative zone displays a summary of leads for the initiative (or initiative hierarchy group) by individual representative. If the Initiative is configured not have a sales representative, the zone will group those leads under 'Leads without Representative'.

To limit the summary of leads by sales representative:

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain.

   The selected initiative opens in the Initiative portal.
5. Go to the **Statistics** page and scroll, as necessary, to view the Lead Summary by Representative zone.
6. Specify **Start Date** and/or **End Date** to limit the summary to leads created within this date range. If specified, start date must be on or before the current date and end date must be on or after the start date. If left blank, end date is defaulted to the current date.
7. If **Include Inactive Leads** is set to No, leads for the initiative (or initiative hierarchy group) regardless of status are included.

The following is displayed in the result set:

- A summary line for the individual sale representative shows the total Lead Count.
- A breakdown is included beneath each sales representative summary line that shows a subtotal Lead Count by lead state.
- An icon is displayed that allows broadcasting of the representative, initiative, date range and lead state (if available) to other zones on the portal. Lead state broadcast is only available on the subtotal lines.

Graphs and Statistics for an Initiative

The following zones display statistics for the initiative's leads.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The system periodically refreshes the statistics for an initiative when the time specified on an initiative has lapsed. A user may also manually initiate the refreshing of statistics if required.

Lead Outcome Graph

This Lead Outcome Graph zone shows the number of Leads linked to the initiative (or initiative group hierarchy) that entered the final state in the last 24 months starting from the most recent completed Lead.

The date shown on the X-axis will be the first day of the month. Note that the information displayed here is based on the most recent completed Lead as of the time the Initiative Statistics is accumulated.

Lead Statistics

The Lead Statistics zone has two charts to provide statistics for leads for the initiative:

- **Leads by State:** The chart shows the number of leads for each state in the lead's lifecycle as of the initiative's Statistics Date/Time. Each state can be represented by a unique color. These colors are defined on the Program Management master configuration. Refer to Program Management Master Configuration for setting up program management options for more information.
- **Active Leads by Representative:** The chart shows the top five individual sales representatives for the initiative (i.e. sales representatives with the largest number of active leads assigned to them as of the Statistics Date/Time). The top five sales representative can be represented by a unique color. These colors are defined on the Program Management master configuration. Refer to Program Management Master Configuration for setting up program management options for more information.

Lead Monthly Statistics

The Lead Monthly Statistics zone has two charts to provide statistics for leads for the initiative:

- **Leads by State:** The chronological graph highlights the number of open, completed and discarded leads over time for the initiative in context. The date displayed on the x-axis represents the start date of the statistics period. For example, 01-01-2013 shows statistics for the period 01-01-2013 through 01-31-2013. Each State can be represented by a unique color. These colors, along with the number of months to display are defined on the Program Management master configuration. Refer to Program Management Master Configuration for setting up program management options for more information.
- **Active Leads by Representative:** The chronological graph highlights the top 4 discard reasons over time for the initiative in context. Statistics for all other discard reasons are accumulated in a catch-all bucket. The date displayed on the x-axis represents the start date of the statistics period. For example, 01-01-2013 shows statistics for the period 01-01-2013 through 01-31-2013.Each Discard Reason can be represented by a unique color. These colors, along with the number of months to display are defined on the Program Management master configuration. Refer to Program Management Master Configuration for setting up program management options for more information.

Displaying and Assigning Unassigned Leads

Use the following procedure to review and assign, as appropriate, non-final and unassigned leads that are associated to the initiative in context.

Copyright © 2026, Oracle and/or its affiliates.

1. Select **Menu**, and then **Sales & Marketing**, and then **Initiative**, and then **Search.**
2. Enter your search criteria in the Initiative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the initiative you want to maintain. The selected initiative opens in the Initiative portal.
5. Go to the **Statistics** page and scroll, as necessary, to view the Unassigned Leads Zone.
6. Enter a Start Date and End Date range that encompasses the desired period.
7. Click **Refresh** to view unassigned leads that were created within the period specified.
8. (optional) Assign one or more leads from the result set:
    a. Select the unassigned leads from the results set.
    b. Enter the sales representative that will be assigned the leads.
    c. Press **Assign to Sales Rep**.
    d. Select the Sales Representative from the dropdown menu and click **OK**.

Maintaining Leads

A lead is an instance of an initiative for a targeted customer. A lead may contain events or actions that are there to encourage a customer to participate in a program

The topics in this section describe the common functions associated with leads.

- My Leads
- Lead Portal
- How To Perform Common Lead Functions

My Leads

You can use the My Lead menu item to maintain leads.

- If your User ID is linked to a Sales Representative, you are navigated to the Sales Representative portal with the user in context.
- If your User ID is not linked to a Sales Representative, you are navigated to the Leads Query portal.

Lead Portal

You use the Leads portal to maintain leads. This portal contains the following zones:

- **Lead:** Displays the basic information about the lead.
- **Lead Events:** Displays the events associated with the lead in chronological order.
- **Related Leads:** Displays the leads that are related to the current lead's related objects (e.g. account).
- **Attachments:** Displays uploaded documents that are related to the lead.
- **Notes:** Displays all notes for the lead and its lead events.

Copyright © 2026, Oracle and/or its affiliates.

- **Third Party Event:** if a third party lead event exists, this zone displays the lead event's information, including the event steps

How To Perform Common Lead Functions

The topics in this section describe how to perform common lead maintenance functions from the Lead portal.

- Manually Creating a Lead
- Maintaining a Lead
- Adding New Lead Events to a Lead
- Adding Lead Notes
- Maintaining Lead Events
- Maintaining Third Party Lead Events

Manually Creating a Lead

Use this procedure to manually create a new lead.

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead** , and then **Add.**
2. Select the lead's Initiative and click **OK.**

   The Account Lead page displays so that you can complete the lead.
   Enter the following information as required by the Initiative:
3. Search for and select the **sales representative** for the lead.
4. Indicate how the customer requested information about the initiative in the **response channel**.
5. Indicate if the customer has **signed up** for the initiative.
6. Search for and select the **account** that will be associated to the lead.
7. Search for and select the **premise** that will be associated to the lead.
8. Click **Save**.

> **Note**
>
> Leads can also be created through the Initiative context menus and Account context menus.

Maintaining a Lead

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead**, and then **Search.**
2. Enter your search criteria in the Lead Query zone.
3. Click **Refresh.**
4. In the search results list (not applicable if Lead ID is specified in the search criteria), click the link for the lead you want to view maintain. The selected lead opens in the Lead portal.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You may also navigate to the Lead portal by clicking on the appropriate lead from various portals and zones. Examples include:
>
> - Sidebar Portal: Alerts zone
> - Initiative Portal: Initiative Leads zone
> - Sales Representative Portal: Initiative Leads zone or Initiative Lead Events

5. Click the appropriate record action button:

   Buttons that appear when the lead is Pending:

   - **Edit** – Click the **Edit** button to allow you to edit the main attributes of the lead. When you click **Save**, the system updates the record.
   - **Delete** – Click the **Delete** button and click **OK** to delete a pending lead from the database. Before you delete a lead, the system verifies that the lead is not referenced anywhere in the system. If it is, you cannot delete it.
   - **Activate** – Click the Activate button to transition a pending lead to the active state. When a lead is activated, lead events for the lead are created to encourage the customers to participate in the initiative.

   Buttons that appear when the lead is Active:

   - **Edit** – *see the description above.*
   - **Discard** – Select the **Discard** button to inactivate an active lead. Select a **Status Reason** indicating why the lead is being inactivated and click **Save.** The values are maintained on the Status Reason control table
   - **Complete** – Click the **Complete** button to indicate that an active lead has been successful.
   - **Sign Up** – Click the **Sign Up** button to initiate a script that is defined on the Initiative's business object. The base package script **C1-IntvSignU** reevaluates the customer to determine if they are still eligible to participate in a program, indicates the customer has signed up or enrolled on the lead and creates a third party lead event (if applicable) to notify the relevant third party of the actions to be performed. The Response Channel on the lead is also updated for analytical purposes

6. Click the **Add Notes** button, in the Notes zone, to capture additional information for the lead.
7. The **Third Party Event Zone** displays third party lead event (if any) details along with the steps involved in the lead event

Adding New Lead Events to a Lead

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead**, and then **Search.**
2. Enter your search criteria in the Lead Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the lead you want to view maintain. The selected lead opens in the Lead portal.
5. To add a lead event to the lead, click the **Add Lead Event** link in the Lead Events zone title bar.

Copyright © 2026, Oracle and/or its affiliates.

6. Search for and select the appropriate lead event type.
7. Specify the Trigger Date of when the lead event is to be executed.
8. Specify the Communication Channel to indicate what form the event will be sent.
9. Click **Save.**

Repeat steps 5 to 9 for add additional lead events.

Adding Lead Notes

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead**, and then **Search.**
2. Enter your search criteria in the Lead Query zone.

   The C1-owned base package search options include:

   - Name and address
   - Related object
   - Representative and initiative
   - Lead ID
3. Click **Refresh.**
4. In the search results list, click the link for the lead you want to view maintain. The selected lead opens in the Lead portal.
5. Scroll, as necessary, to view the Notes zone.
6. Click the **Add Notes** link in the zone's title bar.
7. Enter the desired information in the **Add Notes for Lead** pop-up window and click **Save**.

Maintaining Lead Events

A lead event is an event or action in a lead to take place to encourage a customer to participate in a program. A lead may contain many lead events.

The Lead Event portal is used to maintain lead events. This portal contains the following zones:

- **Lead Event:** Displays the basic information about the lead event.
- **Notes:** Displays user-defined comments about this lead event

To maintain a lead event:

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead Event**, and then **Search.**
2. Enter your search criteria in the Lead Event Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the lead event you want to maintain. The selected lead opens in the Lead Event portal.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You may also navigate to the Lead Event portal by clicking on the appropriate lead event from various portals and zones. Examples include:
>
> - Initiative Portal: Initiative Leads zone
> - Sales Representative Portal: Initiative Leads zone or Initiative Lead Events

5. Click the appropriate record action button:
   - **Edit** – Click the **Edit** button to allow you to edit the main attributes of the lead event. When you click Save, the system updates the record.
   - **Complete** – Click the **Complete** button to indicate a pending lead event is completed.
   - **Cancel** - Select the **Cancel** button to cancel a pending lead event.
6. Click the **Add Notes** icon, in the Notes zone, to capture additional information for the lead.

Maintaining Third Party Lead Events

A third party lead event represents an action within a lead to either encourage a customer to participate in a program, or perform an on-site service. A lead may contain one or more third party lead events.

The Lead Event portal is used to maintain third party lead events. This portal contains the following zones:

- **Lead Event:** Displays the basic information about the lead event.
- **Notes:** Displays user-defined comments about this lead event

To maintain a third party lead event:

1. Select **Menu**, and then **Sales & Marketing**, and then **Lead Event.**
2. Enter your search criteria in the Lead Event Query zone.

   The C1-owned base package search options include:
   - Name and address
   - Related object
   - Initiative and representative
   - Lead and lead event
3. Click **Refresh.**
4. In the search results list, click the link for the lead event you want to maintain. The selected lead opens in the Lead Event portal.

> **Note**
>
> You may also navigate to the Lead Event portal by clicking on the appropriate lead event from various portals and zones. Examples include:
>
> - Initiative Portal: Initiative Leads zone
> - Sales Representative Portal: Initiative Leads zone or Initiative Lead Events

Copyright © 2026, Oracle and/or its affiliates.

5. Click the appropriate record action button:
    - **Edit** – Click the **Edit** button to allow you to edit the main attributes of the lead event. When you click Save, the system updates the record.
    - **In Progress** – Click the **In Progress** button to initiate the notification process and have the third party lead event sent to the representative.
    - **Cancel** - Select the **Cancel** button to cancel a pending lead event.
6. Click the **Add Notes** icon, in the Notes zone, to capture additional information for the lead.

Maintaining Sales Representatives

A Sales Representative is a user or group of users who can be assigned to follow up on leads and responsible for selling the initiatives product or service to the customer.

The topics in this section describe the common functions associated with Sales Representatives.

- Sales Representatives Portal
- How To Perform Common Sales Representative Functions

Sales Representatives Portal

You use the Sales Representative portal to maintain sales representatives. This portal contains the following zones:

- **Sales Representative:** Defines the basic information about the representative.
- **Initiative Lead Summary:** Lists initiatives and information about leads for each initiative for the representative. Clicking the broadcast icon from the summary table in this zone displays the following additional zones:
    - **Initiative Leads:** Displays leads for the initiative and representative.
    - **Initiative Lead Event Summary:** Displays lead event statistics for the initiative lead and representative.
    - **Initiative Lead Events:** Displays lead events for the initiative lead and representative.
- **Group's Representatives:** This zone is displayed for a representative group and lists the individual representatives who are members of the group.
- **Representative's Groups:** This zone is displayed for an individual representative and lists the representative groups of which this representative is a member.

How To Perform Common Sales Representative Functions

The topics in this section describe how to perform common sales representative functions from the Sales Representative portal.

- Creating Sales Representatives
- Maintaining Sales Representatives
- Displaying Leads and Lead Event Information for a Sales Representative
- Reassigning Leads for a Sales Representative

Copyright © 2026, Oracle and/or its affiliates.

Creating Sales Representatives

Use this procedure to create a new sales representative.

1. Select **Menu**, and then **Sales & Marketing**, and then **Sales Representative**, and then **Add.**
2. Select the sales representative type that will define the representative and click **OK.**
3. Enter the appropriate information for the representative:
   - For a Group Sales Representative:
     - Specify the Sales Representative name and description.
     - Search for and select Individual Sales Representative who will be a member of the group.
   - For an Individual Sales Representative:
     - Specify the Sales Representative name and description.
     - Search for and select User for the individual sales representative.
     - Search for and select Group Sales Representative that the individual sales representative will be a member of.
4. Click **Save.**

Maintaining Sales Representatives

1. Select **Menu**, and then **Sales & Marketing**, and then **Sales Representative**, and then **Search.**
2. Enter your search criteria in the Sales Representative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the sales representative you want to maintain. The selected sales representative opens in the Sales Representative portal.

> **Note**
>
> You may also navigate to the Sales Representative portal by clicking on the appropriate Sales Representative from various portals and zones. Examples include:
>
> - Initiative Portal: The Initiative Zone or the Lead Summary By Representative.
> - Lead Portal: Lead zone

5. Click the appropriate record action button:
   - **Edit** – Click the **Edit** button to allow you to edit the main attributes of a sales representative. When you click **Save**, the system updates the record
   - **Duplicate** – Click the **Duplicate** button to create a duplicate of the sales representative being displayed. Before the new object is added to the database, a window appears asking you to enter the unique identifier of the new object. When you click **OK**, the system creates the record by copying the original record

Copyright © 2026, Oracle and/or its affiliates.

- **Deactivate** – Click the **Deactivate** button to transition an active sales representative to an inactive state
- **Delete** – Click the **Delete** button and click **OK** to delete the sales representative from the database. Before you delete a sales representative, the system verifies that the sales representative is not referenced anywhere in the system. If this is the case, you cannot delete the representative. For example, if you attempt to delete a sales representative, the system ensures the sales representative is not referenced on a lead, etc.

6. For an individual sales representative, you can add the sales representative to another group by clicking the **Add Rep to Another Group** link in the Representative's Group zones title bar. Select a group sales representative and click **Save**. You can remove the representative from a group (or groups) by selecting the group(s) and clicking the **Remove Rep from Group(s)** button.
7. For a representative group, you can add a representative to the group by clicking the **Add Rep To This Group** link in the Group's Representatives zone's title bar. Search for and select the individual sales representative to add to the group and click **Save**. You can remove representatives from a group by selecting the representatives and clicking the **Remove Rep(s) from Group** button.

Displaying Leads and Lead Event Information for a Sales Representative

From the Initiative Lead Summary zone on the Sales Representative portal, you can broadcast a representative, initiative, date range and lead state (if available) to other zones on the portal.

- Displaying Leads Associated Information for a Sales Representative
- Sales Representative Initiatives Leads Zone

Displaying Leads Associated Information for a Sales Representative

The Initiative Lead Summary zone on the Sales Representative portal displays the leads that have been assigned to the sales representative. If the representative is a group, leads are summarized for all individuals in the group.

To limit the summary of leads for the sales representative:

1. Select **Menu**, and then **Sales & Marketing**, and then **Sales Representative**, and then **Search.**
2. Enter your search criteria in the Sales Representative Query zone.
3. Click Refresh.
4. In the search results list, click the link for the sales representative you want to view or manage. The selected sales representative opens in the Sales Representative portal.
5. Specify **Start Date** and/or **End Date** to limit the summary to leads created within this date range. If specified, start date must be on or before the current date and end date must be on or after the start date. If left blank, end date is defaulted to the current date.
6. If **Only Show Active Initiatives** is set to **Yes**, only leads linked to initiatives in the active state are summarized. Otherwise, leads for all initiatives regardless of status are included.
7. Click **Refresh.**

The following is displayed in the result set:

- A summary line for each initiative shows the total **Lead Count** for the individual or group rep.

Copyright © 2026, Oracle and/or its affiliates.

- A breakdown is included beneath each initiative summary line that shows a subtotal **Lead Count** by lead state.
- An icon is displayed that allows broadcasting of the representative, initiative, date range and lead state (if available) to other zones on the portal. Lead state broadcast is only available on the subtotal lines.

Sales Representative Initiatives Leads Zone

By clicking on one of the broadcast items in the Initiative Lead Summary zone, the Initiative Leads zone on the Sales Representative portal is populated with relevant lead information for the selected item. If the sales representative is a group, leads for all individuals in the group are shown.

The following is displayed in the result set for each lead:

- Basic **Lead** information for the initiative's lead. You can navigate to the lead by clicking on the lead.
- **Customer Information** related to the lead. You can navigate to the account related to the lead by clicking on the customer information.
- The **Last Complete Event.** You can navigate to the lead event by clicking on the last complete event for a lead.
- For open leads, the **Next Pending Event.** You can navigate to the lead event by clicking on the next pending event for a lead.
- The **Sales Representative** shows the rep assigned to each lead.

For information about using this zone to reassign leads to another representative, see Reassigning Leads for a Sales Representative.

- Initiatives Lead Event Type Summary
- Initiatives Lead Events Summary

Initiatives Lead Event Type Summary

By clicking on one of the broadcast items in the Initiative Lead Summary zone, the Initiative Lead Event Type Summary zone is displayed. If the representative is a group, lead events are accumulated for all individuals in the group. The summary spans the 14 day period on/after the Start Date which defaults to current date if blank. If you want to specify the start date, click Refresh to update the zone.

The following is displayed in the result set:

- The **Event Count** shows the total number of events for a specific Event Type on each day within the 14 day period. You can click on the broadcast icon to see a list of events for the **Event Date** and **Event Type** on the Initiative Lead Events zone.
- The **Event Status** bar provides a breakdown of the total event count by status. Each color segment shown represents a different event status for pending, completed and canceled events. Colors used are defined on the program management master configuration.
  - Hover the mouse pointer over a segment of the bar to see the respective count and status represented by the segment.
  - Click on a segment of the bar to broadcast the event date, event type and event status. Broadcasting from the event status bar instead of the broadcast icon will limit the Initiative Lead Events zone to a particular status.

Initiatives Lead Events Summary

Copyright © 2026, Oracle and/or its affiliates.

By clicking on one of the broadcast items in the Initiative Lead Event Type Summary zone, the Initiative Lead Events zone is populated. This zone displays lead events for the initiative, representative, lead event type (if available) and status (if available). If the representative is a group, events for all individuals in the group are shown. Only lead events with a trigger date on or after the **Start Date** are displayed.

The following is displayed in the result set:

- **Event Date** and **Lead Event** information is shown for each event. You can navigate to a specific lead event by clicking on a lead event.
- **Lead** shows the lead information for the initiative's lead. You can navigate to the lead by clicking on a lead.
- **Customer Information** related to the lead. You can navigate to the account related to the lead by clicking customer information for a lead.
- The **Sales Representative** shows the representative assigned to the lead.

Reassigning Leads for a Sales Representative

To reassign leads from one sales representative to another sales representative:

1. On the Initiatives Leads zone, select the leads that you want to reassign and click **Reassign Rep.**
2. Select the Sales Representative to reassign the selected lead to and click **Save.**

Maintaining Third Party Representatives

A Third Party Representative is someone who can receive assignments to perform specific tasks within a lead; such as, undertaking a visit to a property to perform an inspection or an installation. Refer to The Big Picture of Third Party Representatives for more information

The topics in this section describe the common functions associated with Third Party Representatives.

- Third Party Representatives Portal
- How To Perform Common Third Party Representative Functions

Third Party Representatives Portal

The Third Party Representative portal is used to maintain third party representatives. This portal contains the following zones:

- **Third Party Representative:** Defines the basic information about the third party representative.
- **Initiative Lead Summary:** Displays a summary of initiatives that have leads assigned for the third party representative. Clicking the broadcast icon from the summary table in this zone displays the following additional zones:
  - **Initiative Leads:** Displays leads for the broadcasted initiative and third party representative.
  - **Unassigned Leads:** Displays unassigned leads for the broadcasted initiative that are not yet assigned to a third party representative.

Copyright © 2026, Oracle and/or its affiliates.

How To Perform Common Third Party Representative Functions

The topics in this section describe how to perform common representative functions from the Third Party Representative portal.

- Creating Third Party Representatives
- Maintaining Third Party Representatives
- Displaying Leads Associated Information for a Third Party Representative
- Displaying and Assigning Leads to a Third Party Representative

Creating Third Party Representatives

Use this procedure to create a new third party representative.

> **Note**
>
> Before adding the third party representative, they must be added as a Person without an account in the system. For more information about this process, see How To Add A New Customer From Control Central.

1. Select **Menu**, and then **Sales & Marketing**, and then **Third Party Representative**, and then **Add.**
2. Enter the appropriate information for the representative:
   - Specify the Third Party Representative name and description.
   - Search for and select the Person record for the third party representative.
   - Select the External System that will be used to communicate to the third party representative.
3. Click **Save.**

Maintaining Third Party Representatives

1. Select **Menu**, and then **Sales & Marketing**, and then **Third Party Representative.**
2. Enter your search criteria in the Third Party Representative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the third party representative you want to maintain. The selected third party representative opens in the Third Party Representative portal.
5. Click the appropriate record action button:
   - **Edit** – Click the **Edit** button to allow you to edit the main attributes of a third party representative. When you click **Save**, the system updates the record
   - **Duplicate** – Click the **Duplicate** button to create a duplicate of the third party representative being displayed. Before the new object is added to the database, a window appears asking you to enter the unique identifier of the new object. When you click **OK**, the system creates the record by copying the original record

Copyright © 2026, Oracle and/or its affiliates.

- **Deactivate** – Click the **Deactivate** button to transition an active third party representative to an inactive state
- **Delete** – Click the **Delete** button and click **OK** to delete the third party representative from the database. Before you delete a third party representative, the system verifies that the third party representative is not referenced anywhere in the system. If it is, you cannot delete it. For example, if you attempt to delete a third party representative, the system ensures the third party representative is not referenced on a lead, etc.

Displaying Leads Associated Information for a Third Party Representative

The Initiative Lead Summary zone on the Third Party Representative portal displays the leads that have been assigned to the Third Party representative.

To limit the summary of leads for the sales representative:

1. Select **Menu**, and then **Sales & Marketing**, and then **Third Party Representative**, and then **Search.**
2. Enter your search criteria in the Third Party Representative Query zone.
3. Click **Refresh**.
4. In the search results list, click the link for the sales representative you want to view or manage. The selected sales representative opens in the Sales Representative portal.
5. Specify **Start Date** and/or **End Date** to limit the summary to leads created within this date range. If specified, start date must be on or before the current date and end date must be on or after the start date. If left blank, end date is defaulted to the current date.
6. If **Only Show Active Initiatives** is set to **Yes**, only leads linked to initiatives in the active state are summarized; otherwise, leads for all initiatives regardless of status are included.
7. Click **Refresh.**

The following is displayed in the result set:

- A summary line for each initiative shows the total **Lead Count** for the third party representative.
- A breakdown is included beneath each initiative summary line that shows a subtotal **Lead Count** by lead state.
- An icon is displayed that allows broadcasting of the third party representative, initiative, date range and lead state (if available) to other zones on the portal. Lead state broadcast is only available on the subtotal lines.

The Initiative Leads zone on the Third Party Representative portal displays the leads that have been assigned to the third party representative. The following is displayed in the result set for each lead:

- Basic **Lead** information for the initiative's lead. You can navigate to the lead by clicking on the lead.
- **Customer Information** related to the lead. You can navigate to the account related to the lead by clicking on the customer information.
- The **Last Completed Step.** You can navigate to the lead event by clicking on the event/step information.
- For active third party lead events, the **Next Pending Step.** You can navigate to the lead event by clicking on the event/step information.
- The **Sales Representative** shows the representative assigned to the lead.

Copyright © 2026, Oracle and/or its affiliates.

The Unassigned Leads zone on the Third Party Representative portal displays active leads that are not yet assigned to a third party representative. The following is displayed in the result set for each lead:

- Basic **Lead** information for the initiative's lead. You can navigate to the lead by clicking on the lead.
- **Customer Information** related to the lead. You can navigate to the account related to the lead by clicking on the customer information.
- The **Sales Representative** shows the representative assigned to the lead.
- You can assign leads to the third party representative using the **Assign to Third Party** button.

See Displaying and Assigning Leads to a Third Party Representative for more information.

Displaying and Assigning Leads to a Third Party Representative

Use the following procedure to view leads that are not assigned to a third party representative, and assign one or more of those leads to the third party representative in context.

1. Select **Menu**, and then **Sales & Marketing**, and then **Third Party Representative**, and then **Search.**
2. Enter your search criteria in the Third Party Representative Query zone.
3. Click **Refresh.**
4. In the search results list, click the link for the third party representative you want to view. The selected third party representative opens in the Third Party Representative portal.
5. The **Initiative Lead Summary** zone shows the initiatives that have leads assigned to the third party representative in context. Click the broadcast button on an initiative to view its unassigned leads.
6. Scroll down to view the **Unassigned Leads** zone. This zone will display if there are active leads (for the broadcast initiative) that are not yet assigned to a third party representative.
7. To assign one or more leads to the third party representative in context:
   a. Select the unassigned lead(s) from the results set.
   b. Press **Assign to Third Party**.
   c. Click **OK**.

Rates

When you set up a customer in the system, you create a service agreement for every service that the customer uses. The most important information specified on a service agreement is the rate. The rate controls:

- How the service's charges are calculated.
- How the charges are described on the customer's bills.
- How the general ledger is affected by the charges.

It's important to understand that while *most* service agreements are associated to a rate, you may have service agreements that don't use a rate to calculate their charges; for example, you

Copyright © 2026, Oracle and/or its affiliates.

may have billable charge service agreements to bill a customer for charges that have been calculated in an external system. Whether or not a service agreement uses a rate and the list of rates that can be specified on a service agreement are controlled by the service agreement's SA Type.

In addition to calculating charges for service, your implementation can also use rates to calculate adjustments.

NOT_SUPPORTED

Refer to How Rates Affect The Information On A Bill for how billing uses the information in a rate to calculate a bill segment. Refer to Calculated Adjustments for how to use a rate to calculate an adjustment.

Note

Do not equate a rate in your legacy system with a rate in this system. You may find that the system's flexibility will result in fewer rates than you have today. For example, your current rate structure may have three rates for service because there are different prices based on the customer's geographic area. In this system, however, you can set up a single rate and vary the price based on where the customer lives.

This chapter describes the structure and process to configure rates.

Note

If you're using Oracle Utilities Rate Cloud Service, refer to the Oracle Utilities Rate Cloud Service chapter in the *Business User Guide*, and the Defining Oracle Utilities Rate Cloud Service Options chapter in the *Administrative User Guide* for information about how the rate functionality works in the context of that cloud service.

Rate schedules created using the new rate engine are referred to as **Calculation Rule-Based Rates** or, simply, **Rates**; and those created using the original rate engine are referred to as **Component-Based Rates**. This chapter focuses on Calculation Rule-Based Rates, unless stated otherwise.

Caution

Component-Based Rates, or classic rates, is not the recommended approach for new implementations. Component-Based Rates is intended for upgrading customers who currently use this rate structure and have not converted to Calculation Rule-Based Rates.

- Calculation Rule-Based Rates

Copyright © 2026, Oracle and/or its affiliates.

Calculation Rule-Based Rates

Calculation Rule-Based Rates (Rates) offer a flexible and reusable structure to calculate charges, centralize configuration, and reduce your development effort and time.

This chapter describes the structure and process to configure Rates.

- All Calculation Rule-Based Rates Share a Common Structure
- Control Tables That Must Be Set Up Before Creating a Rate Schedule
- Setting Up Bill Factors
- Rate Schedule Execution
- Understanding Calculation Groups and Rules
- Designing Calculation Groups and Rules
- The Big Picture of Calculation Rule Eligibility Rules
- Working with Calculation Groups and Rules
- Creating Rate Schedules
- Base Package Calculation Rule Descriptions
- Rate Check
- Effective Dates and Price Proration
- Setting Up Estimated Consumption for Items
- Setting Up Degree Days (and Average Temperatures)
- United Kingdom VAT and CCL

All Calculation Rule-Based Rates Share a Common Structure

The basic components used to calculate charges are **Calculation Rules**. Each calculation rule is configured to perform a specific type of calculation, such as calculating tax charges, the number of days in a billing period, or the amount of consumption used by a customer in a billing period.

Calculation rules are organized into **Calculation Groups**, which are logical collections of calculation rules that can be referenced by rate schedules. Calculation rules and groups are commonly "generic" and can be applied to multiple rates.

Furthermore, calculation groups can be nested within other calculation rules. When the parent rule is processed, it executes all the rules that are nested in its referenced group. This strategy allows a single rule to execute any number of nested rules. Nesting also allows common calculations to be encapsulated into a set of calculation rules that can be included in multiple rates.

Calculation groups are defined for a rate schedule in one of three ways:

- **Pre-Processing Calculation Groups** perform initial calculations in preparation for the rate version calculation group.

Copyright © 2026, Oracle and/or its affiliates.

- **Rate Version Calculation Groups** are effective-dated calculation groups that perform the main calculations of the rate schedule.
- **Post-Processing Calculation Groups** perform calculations after rate version calculation groups are executed.

  See About Calculation Groups for more information.

At the highest level, every rate has a single rate schedule that contains basic information that shouldn't change over time; such as, the billing frequency, currency, and the rate's proration strategy. This information combined with the calculations groups, and other high-level configuration, comprise the Rate Schedule

Control Tables That Must Be Set Up Before Creating a Rate Schedule

This section describes tables that must be set up before you can add a rate schedule.

- Master Configuration
- Defining Frequency Codes

Master Configuration

The **Rate Engine Configuration** master configuration can be used to define the behavior of the Rate Schedule portals used when creating rate schedules.

To edit this master configuration, select **Admin**, and then **Master Configuration,** and click the edit icon for the **Rate Engine Configuration.**

The **Rate Engine Implementation** dropdown list defines which rate engine(s) will be used by your organization.

- **Calculation-Rule Based** indicates that your implementation uses calculation rule-based rates, which require the configuration of rate schedules with calculation groups and calculation rules.
- **Component-Based** indicates that your implementation uses component-based rates, which require the configuration of rate versions with rate components, and rate schedules with SQ and register rules.
- **Both** indicates that your implementation uses both rate engines. If this applies to you, then the system will use this option to ease navigation as follows:
  - Select **Menu**, and then **Rates**, and then **Rate Schedule**, and then **Add** to launch the page to add a new rate schedule. *(A hyperlink is included to take the user to the component-based rate schedule maintenance page, if necessary.)*
  - Select **Menu**, and then **Rates**, and then **Rate Schedule**, and then **Search** to launch the **Rate Schedule Query** portal.
  - The query results attempt to highlight component-based rates by appending **"(Component-Based)"** to the rate schedule's description.
  - Once a rate schedule is selected from the result set, the system navigates the user to either the calculation rule-based **Rate Schedule portal** or the component-based **Rate Schedule page**.

The **External Communication** section is used when the rate engine needs to communicate with an external system. For example, service quantities are needed to test a rate on the **Rate Check** portal. These service quantities may be entered on the portal, retrieved from an existing

Copyright © 2026, Oracle and/or its affiliates.

bill segment, or (in the case of a service agreement that requires bill determinants) request service quantities from Oracle Utilities Meter Data Management.

- **Request Type** identifies the various data requests that can be made to an external system. Request types are defined in the RATE_EXT_REQ_TYPE_FLG lookup field. To introduce additional request types, add appropriate lookup values to this lookup field for each request type you wish to support.

  The base package includes the following business objects to be defined on the outbound message types created for such requests:

  – **Rate Check Usage**: C1-UsageRequestOutMsg (used to request bill determinants from MDM on the rate check portal)

- **External System** and **Outbound Message Type** are used when creating the outbound message for the data request.

Defining Frequency Codes

When you create a rate schedule, you must enter a code defining the frequency in which the prices are expressed, e.g., monthly, quarterly, biannual, etc. To set up frequency codes, open **Admin**, and then **Rates**, and then **Frequency.**

Description of Page

Enter a unique **Frequency** ID and **Description** for each frequency.

Before entering the remaining fields, you must analyze the "proration provisions" in your rates. For example, consider a rate used to calculate charges for a "month's worth of service". Assume this rate's charges will not be prorated when the customer's service period is anywhere between 27 and 33 days. This rate requires a frequency that indicates:

- 12 **Periods / Year** (because it's a monthly rate).
- **Minimum Days Offset** equal to 3.
- **Maximum Days Offset** equal to 3.

When the system creates a bill using a rate with this frequency, it compares the number of days of consumption on the bill to the minimum and maximum number of days inferred by the rate's frequency. In the example above, the minimum number of days would be 27 (365 days / 12 period per year - 3 offset for minimum days). The maximum number of days would be 33 (365 days / 12 periods per year + 3 offset for maximum days). Therefore, if a bill's consumption period is less than 27 days or more than 33 days, the bill's charges are prorated.

> **Note**
>
> **Overriding the system's standard proration logic.** If the standard proration logic does not satisfy your requirements, you may plug in an override proration algorithm to calculate the proration factors as required by your implementation. For example, we have encountered companies who treat a year has have 360 days rather than 365 days. Such a company would need to develop an override proration algorithm and plug it on the Installation Record.

Where Used

Copyright © 2026, Oracle and/or its affiliates.

Every Rate Schedule must have a frequency. Refer to Defining Rate Schedules for more information.

Setting Up Bill Factors

This section describes bill factors and how they are used to levy the many charges referenced in a rate.

> **Warning**
>
> Most bill factors exist to support calculations performed by calculation rules. This means that you must design your calculation rules before you can design your bill factors. We strongly recommend that you design "on paper" how each rate's calculation rules look before attempting to set up bill factors.

- An Overview of Bill Factors
- The Structure of a Bill Factor
- An Illustration of a Bill Factor and Its Characteristics
- Deriving/Passing In Characteristic Values
- Defining Bill Factors
- Defining Bill Factor Values

An Overview of Bill Factors

The primary reason that calculation rules exist is to define how much to charge for the services sold by your company. When you create calculation rules, you have four ways to indicate how much to charge:

- Specify the value directly in the calculation rule.
- Define that the value is defined in a *bill factor*.
- Define that the value is calculated by a *for calculation purposes only* calculation rule.
- Tell the calculation rule to call an algorithm to get the price.

The remainder of this section describes the bill factor approach. The other methods are discussed in Case Study: Three Simple Rates .

Bill factors are used to pass a value to a calculation rule; such as specifying a tax rate. To further illustrate this, you could specify the current tax rate, say 6%, directly in every rate's calculation rules OR you could indicate every rate should levy the current state tax rate and have the system look up the applicable rate when it calculates bills. The latter approach involves using a bill factor to indicate that the amount to charge should be retrieved from someplace other than the calculation rule when a bill is calculated.

You can use bill factors to define any of the following types of charges:

- A flat amount. This would typically be used to define service charges, minimum charges, and asset charges.

Copyright © 2026, Oracle and/or its affiliates.

- An amount for a given unit of measure. This would typically be used to define the price for measured consumption (such as, the price per therm).
- A percentage. This would typically be used for tax rates and discount percents.

As a rule of thumb, you would use a bill factor (rather than specify the value in the calculation rule) when any of the following situations exist:

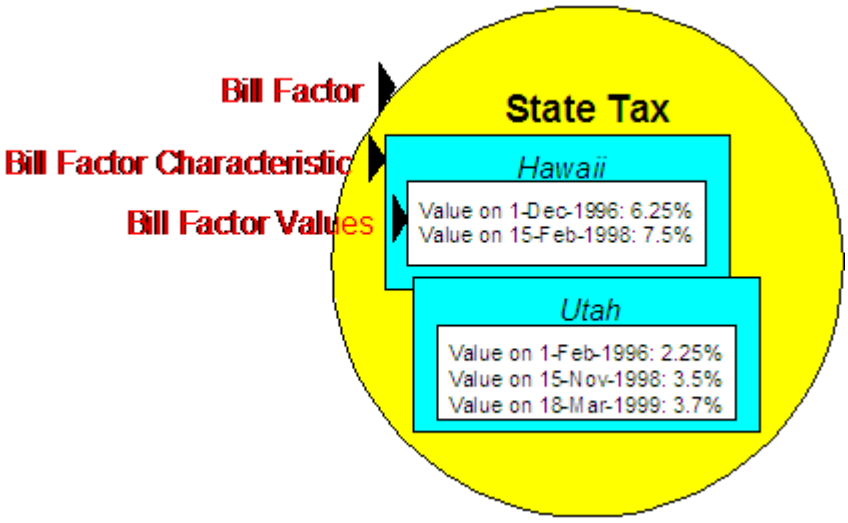
- The same charge exists in many rates. For example, if there are three rates that all contain the same service charge, it would make sense to use a bill factor to levy this charge rather than specify the same value on three calculation rules.
- The amount being charged is dependent upon some physical characteristic of the customer's equipment. For example, if the price differs depending on whether the customer owns the transformer, you would use a bill factor to specify the price.
- The amount being charged is dictated by some external organization and therefore can change independently from the rate. For example, tax percentages and use fees are set by taxing authorities and can be changed at any time (and you don't want to change the rate when these charges change).
- The amount being charged varies depending on WHERE the customer lives. If you have a rate where the price per gallon differs depending on where the customer lives, you should use a bill factor to levy this type of charge.
- A charge (or discount) is only levied for a subset of customers for defined periods of time. For example, if you have a discount that is only given to customers who have had service for more than two years, you would use a bill factor to specify the discount percent.
- The price differs per customer. For example, you may have a rate for large gas customers where the price per therm is different for every customer. In this situation, you would use a bill factor to specify the price.
- The price differs based on the number of weeks of consumption. This typically occurs in a deregulated environment when the price charge is based on the weighted average spot market price in effect during the bill period.

> **Warning**
>
> As you can deduce, bill factors are extremely flexible. While the flexibility makes for a very powerful rate engine, it also presents you a dizzying array of options. We have discovered that to take advantage of the flexibility, you need to gain an overall understanding of the pricing and taxation provisions in ALL of your rates. Only then can you make logical and pragmatic decisions in respect of the number and type of bill factors.

The Structure of a Bill Factor

All bill factors share a common structure as illustrated below:

Copyright © 2026, Oracle and/or its affiliates.



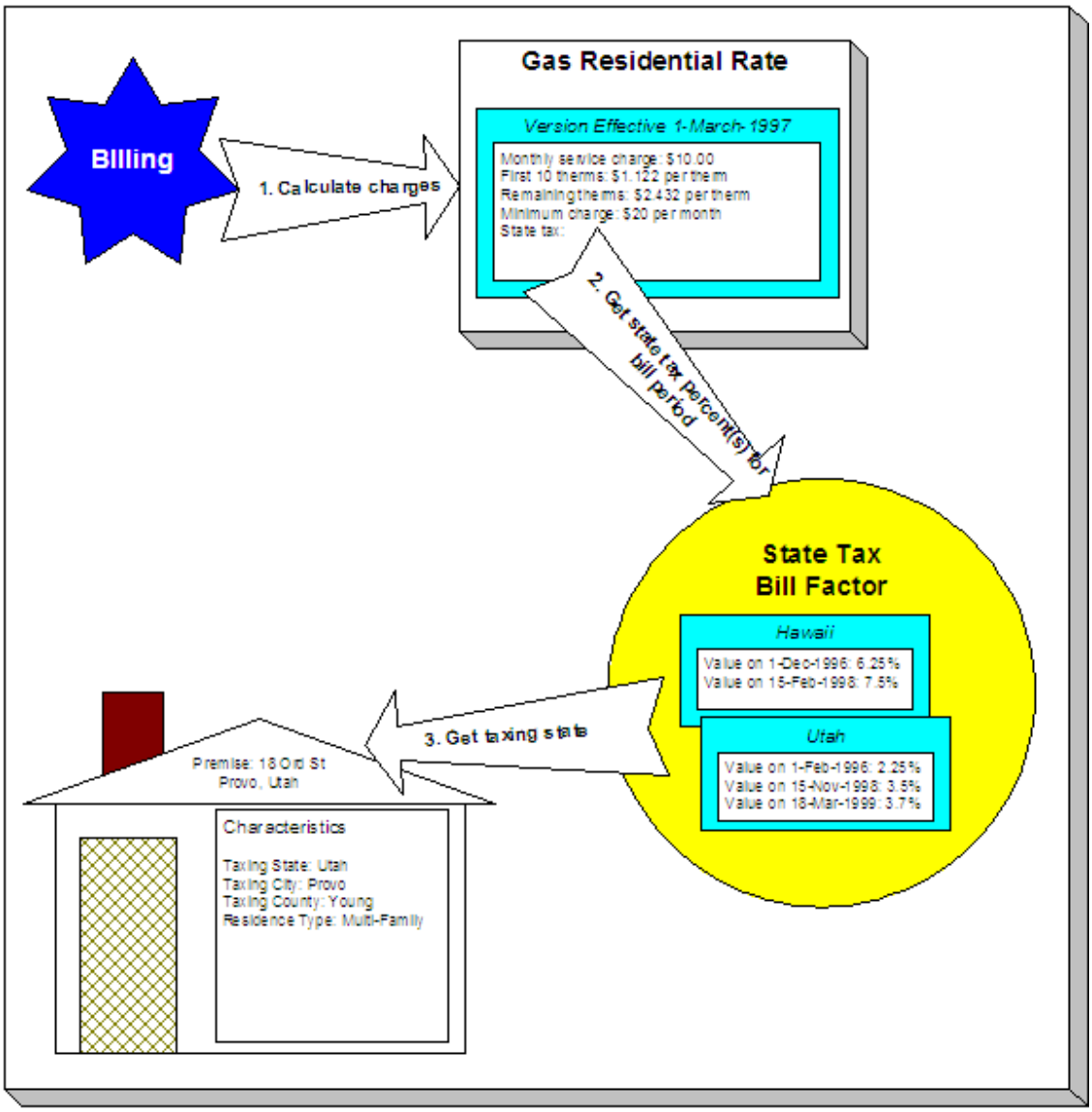
The above example shows a standard bill factor, which is made up of the following tables:

- **Bill Factor**. A **bill factor** contains descriptive information and attributes that control how the bill factor may be used in the system.
- **Bill Factor Characteristic**. All bill factors have the capability of having different values depending on some characteristic of the premise or service point. For example, a bill factor used to levy a state tax would have a different tax rate depending on the state in which the customer resides. A **bill factor characteristic** must exist for every state with a tax (states without a tax will not be linked to the state tax bill factor). At billing time, if the system cannot find a bill factor characteristic that corresponds with the customer's state of residence, the charge will be skipped.
- **Bill Factor Value**. Each bill factor characteristic may have many **bill factor values** over time. To continue with our state tax example, each state with a tax can have different tax rates over time. There are different ways in which a bill factor's values may be defined:
  - For bill factors whose values are applicable to many customers where only one value is effective on a given date (such as, a city tax) the values are defined using bill factor values. Refer to Defining Bill Factor Values for more information.
  - For bill factors whose value differs per customer, the value is defined in the customer's service agreement. Refer to Service Agreement - Contract Value for more information about bill factors in contract terms.
  - For bill factors whose values are applicable to many customers where more than one value is effective on a given date (such as, an interval price or a weighted average price) the values are defined on an interval value set. Refer to Defining Interval Values for more information about interval value sets.

An Illustration of a Bill Factor and Its Characteristics

The following picture illustrates how a bill factor and its characteristics are used to retrieve the relevant state tax at billing time:

Copyright © 2026, Oracle and/or its affiliates.



The following points describe the illustration:

- At billing time, the billing engine sends a request to a rate to calculate the charges for a customer's consumption.
- The rate calculates charges without bill factors until it encounters the calculation rule that levies state tax. This calculation rule references a bill factor. This means the rate must get the tax rate(s) in effect during the billing period from this bill factor.
- The state tax bill factor contains an attribute defining that the premise's taxing state characteristic controls the bill factor value. The bill factor therefore extracts the taxing state from the premise.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Deriving characteristic values.** Rather than have the system extract the characteristic value from an entity, you can setup the system to derive the characteristic value when the rate is calculated. For example, if all of your customers are located in a single state, you may not want to maintain the taxing state on every premise. To do this, you could setup the rate to "hard code" a taxing state of say Hawaii. This is an advanced topic, but it may prove useful for your implementation. Refer to Deriving / Passing In Characteristic Values for the details.

- The premise returns the taxing state and the bill factor extracts the tax rate(s) in effect during the bill period and returns them to the rate.
- The rate applies the tax percents and returns the charges to billing.

> **Note**
>
> **Some bill factors don't need a characteristic.** There are bill factors whose value does not differ based on a characteristic. For example, if your organization offers employee discounts, the discount percent is probably the same for all employees and therefore this bill factor doesn't need a characteristic to select the bill factor value at billing time. However, because of the relational design of the system, every bill factor value must reference both a bill factor and a characteristic value. Therefore, if you have bill factors whose value is not related to a characteristic, you must specify a characteristic type on the bill factor with a characteristic value that is exactly the same as the characteristic type.

> **NOT_SUPPORTED**
>
> For more information about setting up characteristics, see Setting Up Characteristic Types and Their Values .

Deriving/Passing In Characteristic Values

In An Illustration of a Bill Factor and Its Characteristics, we described how the system retrieves a characteristic value from a premise when a bill factor is encountered that uses a premise characteristic. This is a very common technique and is easy to understand. However, there are situations when an entity in Oracle Utilities Customer Care and Billing doesn't contain the appropriate characteristic value and therefore other techniques must be used. These techniques may not apply to your implementation, but they're worth understanding to help you form an intuitive understanding of rate application:

- There are implementations that only use the Oracle Utilities Customer Care and Billing rate engine. These implementations do not setup accounts, premises, etc. Rather, they simply setup their rate schedules, calculation rules, and bill factors. The rate engine is called from another application. This means the rate engine can't retrieve characteristic values from within Oracle Utilities Customer Care and Billing (because there are no objects where the characteristic values are defined). In this situation, the application that calls the rate engine

Copyright © 2026, Oracle and/or its affiliates.

must supply the characteristic types and values to be used when a bill factor requires this type of information.

- Assume you have a charge that is based on the account's customer class and the service agreement's revenue class. The characteristic value for this type of charge is not extractable through a single entity. Rather, the system must construct the characteristic value in real-time by concatenating information from the account and service agreement that is being billed. In this situation, you can use pre-processing calculation group to derive this characteristic type and value. This calculation group would create a characteristic type and value in real time by concatenating the account's customer class and the service agreement's revenue class. This characteristic type and value only exists in memory while the rate application executes.

When you setup a bill factor to use a characteristic whose value is derived or passed in, you need to define its characteristic source as Characteristics Collection. This source tells the system to extract the characteristic value from memory rather than extracting it from an object within the system.

> **Note**
>
> The plug-in spots on a calculation rule have access to the Characteristics Collection. You might find this useful if you have a calculation rule that needs to calculate characteristic values for subsequent calculation rules.

Defining Bill Factors

You create a bill factor for every type of charge whose value is not specified directly in a calculation rule.

It is recommended to use two windows to set up a bill factor. On the first, you define general information describing how the bill factor is used in the system. On the second, you define all the characteristic values for the bill factor.

- Defining General Bill Factor Information
- Setting Up Bill Factor Characteristics

Defining General Bill Factor Information

Begin to define a bill factor by selecting **Menu**, and then **Rates**, and then **Bill Factor**, and then **Add.**

Description of Page

Enter a unique **Bill Factor** code and a **Description** for the bill factor.

Use **Bill Factor Type** to indicate if this bill factor uses Regular, Interval (i.e., real-time), or Time of Use values.

- **Regular**. Indicates that this bill factor's values are specified on Bill Factor Value - Main or on Service Agreement - Contract Values.
- **Interval**. Indicates that this bill factor's values are specified on Defining Interval Values

Copyright © 2026, Oracle and/or its affiliates.

- **Time of Use**. Indicates that this bill factor's values are specified on Bill Factor TOU Values or on SA Interval Billing Info - TOU Contract Values (**Note:** This option is only applicable to Conponent-Based Rates).

Enter a **Currency Code** to define the bill factor's denomination, or currency. This field is disabled if the bill factor is referenced on a calculation rule.

> **Note**
>
> Bill factors referenced on a calculation rules must be denominated in the same currency as is the rate.

Select the **Value Type** to define the kind of value maintained by this bill factor. The Value Type options are:

- **Charge**. The value maintained in this bill factor is a fixed amount (e.g., a flat charge).
- **Percentage**. The value maintained in this bill factor is a percentage.
- **Unit Rate**. The value maintained in the bill factor is a price per service quantity.

This field is disabled if the bill factor is already referenced on a calculation rule or if the bill factor already has values

Turn on **Error If No Value** if the system should create a billing error when a bill factor value doesn't exist for the characteristic value linked to the customer. To clarify, understand that there are different types of bill factors:

- Those that should have a value for every customer whose rate references the bill factor. A bill factor used to levy the relevant state tax is a good example (assuming every state has a tax). For this type of bill factor, if the system cannot find a bill factor value for the taxing state, it must generate a billing error because data is missing from the system. You would turn on **Error if No Value** for this type of bill factor.
- Those that only have a value when the customer has a characteristic value that levies a charge. A bill factor used to levy the relevant city tax is a good example (because not all cities levy taxes). For this type of bill factor, if the system cannot find a bill factor value for the city in which the customer lives, the calculation rule will be skipped and no billing error will be generated. You would turn off **Error if No Value** for this type of bill factor

Bill factors used to levy taxes may or may not be eligible for tax exemptions; it's up to the respective taxing authority. Customers who are eligible for the tax exemption will have their tax obligation reduced (or nullified) at billing time. For bill factors that are eligible for tax exemptions, turn on **Tax Exemption**. If a service agreement is fully or partially exempt for the taxes defined by a given bill factor, the bill factor and the exemption information for the customer must be indicated on the Service Agreement page. This field will be disabled if the bill factor is referenced as being tax exempt in a service agreement.

> **NOT_SUPPORTED**
>
> For more information about how to make a customer tax exempt, refer to Service Agreement - Tax Exemptions.

Some types of bill factors do not have the same value for all customers. Rather, their value may be unique for every customer or it may be unique for some customers and common for

Copyright © 2026, Oracle and/or its affiliates.

others. For example, you may have rates where the price per kilowatt-hour is defined for every customer using the rate. You may have other rates where certain customers have their own price per kilowatt-hour defined and other customers who use the value defined on the bill factor. For any type of bill factor where a customer may have their own unique price, turn on **Value In Contract Term**. For any service agreement that has a unique value for a given bill factor, the bill factor and the service agreement's value must be indicated on the Service Agreement page. When applying the rate for a bill factor of this type, the system will first check for the existence of a contract value for the service agreement and if one is not found, it will use the value on the bill factor. This field will be disabled if a contract term referencing this bill factor exists.

> **Note**
>
> The **Error if No Value** switch also affects what happens if there is no contract term for the bill factor. If the error switch is on, a billing error will be generated. If the error switch is off, no billing error will be generated and the calculation rule will be skipped.

> **NOT_SUPPORTED**
>
> For more information about how to specify a bill factor value in a customer's contract, refer to Service Agreement - Rate Info.

Contract riders are used when the charge associated with the bill factor is only applicable to some customers for some period of time. For example, if a bill factor is used to provide a discount during the first 3 months of service, then a contract rider would be used. Turn on **Contract Rider Applicability** if the bill factor's applicability is contingent on the existence of a contract rider. This field will be gray if a contract rider referencing this bill factor exists on a service agreement.

> **NOT_SUPPORTED**
>
> For more information about how to establish a contract rider for a customer, refer to Service Agreement - Contract Riders.

The proration information tells the system how to handle changes to bill factor values and their optional contract riders that may occur during a customer's bill period. Turn on **Allow Proration** if changes to the bill factor's value and/or contract riders should be prorated. If the **Bill Factor Type** is not specified as Interval and if changes should **not** be prorated, indicate the **Rate Selection Date Algorithm** to use when the system retrieves bill factor values or determines if contract riders are applicable.

If you have sub service agreements, turn on **Use Sub SA Contract** terms if billing should access contract riders, values, tax exemptions from the sub SA rather than the master. If this switch is off, the system will use the master SA's riders, contract value, and tax exemptions when it creates bills for a sub service agreement. Refer to Sub Service Agreements for more information.

If the Contract Management module is not turned off, which includes umbrella agreement management, and you have checked either Tax Exemption, Value in Contract Term or Contract Rider Availability, indicate the **Terms of Service Usage**. Valid values are Check SA Only,

Copyright © 2026, Oracle and/or its affiliates.

Check TOS Only and Check TOS First, then SA. Refer to Overriding Rate Terms for details about how to set this value based on your business needs.

Bill Factor Tree

The right half of this page is dedicated to a tree that shows the bill factors characteristics and their respective values (for Regular bill factors) or interval value Sets (for Interval bill factors). You can use this tree to both view high level information about these objects and to transfer to the respective page in which an object is maintained by using the available Go To buttons.

Where Used

Many types of Rate Components may use a Bill Factor to define the price being charged. Refer to Designing Calculation Groups and Rules for more information.

The BILPC-TOTAL algorithm type that is used to levy late payment charges references the bill factor that defines the late payment charge percent. Refer to SA Type - Main for more information about this algorithm.

A Service Agreement references Bill Factors as follows:

- When a service agreement's Apply To calculation rule references tax exemptable bill factors AND the customer is tax exempt, the service agreement will define the bill factors from which the customer is exempt. Refer to Service Agreement - Tax Exemptions for more information.
- When a service agreement's rate references bill factors whose prices are customer-specific (i.e., specified on a contract term), the service agreement will contain prices for these bill factors. Refer to Service Agreement - Contract Values for more information.
- When a service agreement's rate references bill factors that are contingent on contract riders, the service agreement will define the rider period (if any) for these bill factors. Refer to Service Agreement - Contract Riders for more information.

Setting Up Bill Factor Characteristics

After defining general information about a bill factor, open **Menu**, and then **Rates**, and then **Bill Factor**, and then **Search** and navigate to the **Bill Factor Characteristics** page to define characteristic values where applicable. For example, a bill factor used to levy the applicable city tax would require the definition of every city that levies a tax.

> **Note**
>
> If all values for a bill factor are specified in contract terms, you do not need to set up Bill Factor Characteristics.

Description of Page

Enter a **Characteristic Type** to indicate the type of characteristic that controls how the bill factor's value is defined. This field will be disabled if Bill Factor Characteristics are specified in the following grid.

The **Characteristic Source** defines where the system retrieves the characteristic value at billing time. The Characteristic Source options are:

Copyright © 2026, Oracle and/or its affiliates.

Account	The characteristic type's value will be retrieved from the service agreement's account.
Characteristic Collection	Refer to Deriving / Passing In Characteristic Values for an explanation of this value.
Main Person	The characteristic type's value will be retrieved from the main person linked to the service agreement's account.
N/A	The characteristic type's value is the same for all customers using the rate. If this option is used, you must still choose a Characteristic Type. This Characteristic Type should have a single Characteristic Value. The ID of the characteristic value must be identical to the ID of the characteristic type.
Premise	The characteristic type's value will be retrieved from the service agreement's characteristic premise.
SA	The characteristic type's value will be retrieved from the service agreement.
SP	The characteristic type's value will be retrieved from the customer's service point. If multiple service points are linked to the service agreement, one will be chosen at random.

NOT_SUPPORTED

For more information about characteristics, see Setting Up Characteristic Types and Their Values and An Illustration Of A Bill Factor And Its Characteristics.

Use the add and remove buttons to define the **Characteristic Values** that are relevant for the bill factor's Characteristic Type. For example, if the bill factor is used to levy a city tax, you would define the city's that have a city tax in the above collection.

Use the **Minutes Per Interval** if your Bill Factor is of type Interval. This defines, for each characteristic value, the number of minutes expected between each record of interval data.

Use the **External ID** if the values for this characteristic are interfaced to the system from an external source and you want to record an identifier for the source.

Note

If the bill factor's value is the same for all customers, you still must link a "dummy" characteristic value to the bill factor. In other words, you'll have to create a characteristic type called "n/a" and define for it a characteristic value of "n/a". Then, on the bill factor, you reference the "n/a" characteristic type and on the above window, you indicate a characteristic value of "n/a".

Copyright © 2026, Oracle and/or its affiliates.

Defining Bill Factor Values

After the bill factor and its characteristic values are defined, you define the price / percent and GL distribution associated with each bill factor characteristic. The topics in this section describe the pages used to set up this information.

- Bill Factor Value - Main
- Bill Factor Value - GL Distribution

Bill Factor Value - Main

After the bill factor and its characteristic values are defined, select **Menu**, and then **Rates**, and then **Bill Factor Value** to define the price / percent applicable to each bill factor characteristic.

> **Important**
>
> You do not set up Bill Factor Values for bill factors whose values are specified in an interval value data set or if all the values are customer specific and are defined as contract terms. Refer to Defining Interval Values for more information about interval value sets. Refer to Service Agreement - Contract Value for more information about bill factors in contract terms.

Description of Page

Enter the **Effective Date** and respective **Value** for the bill factor and characteristic value. The first date should be the earliest date on which the bill factor's charges are effective. Subsequent effective dates and values are required whenever the value changes.

> **Note**
>
> **Default note.** A **Value** of 0, zero, will be defaulted.

Bill Factor Value - GL Distribution

Select **Menu**, and then **Rates**, and then **Bill Factor Value** and navigate to the **GL Distribution** page to define the GL account affected by the bill factor characteristic.

Description of Page

If the bill factor characteristic's charges are booked to a different GL account depending on the characteristic value, specify the appropriate **Distribution Code** for each **Revenue Class**.

Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> Specifying distribution codes on a bill factor is optional. You would only do this if the GL distribution code differs based on the characteristic value (such as, the city tax payable GL account differs depending on the city). If you do not enter the above information, the distribution code will be used.
>
> The GL distribution code on the bill factor's value will not be used in the following cases:
>
> - Value derived solely on contract value or contract value in combination with contract rider and/or tax exemption, where characteristic value is no longer used to determine the bill factor value.
> - Value derived in calculate service quantity and math calculation rules where multiple bill factors may be configured in the rule resulting to multiple values contributing to the calculation of the line amount.

Rate Schedule Execution

When a rate schedule is executed, the system executes the calculation rules contained in the calculation groups defined for the rate schedule as follows:

1. First, the system executes the calculation rules defined within Pre-Processing Calculation Groups.

- The rate engine may create a bill segment calculation header for pre-processing calculations, if bill calculation lines are to be created.
- Calculation groups are executed in order according to the Sequence defined on the rate schedule.
- Calculation rules within each group are executed according to the Sequence configured within the group.

2. Next, the system executes the calculation rules within the Rate Version Calculation Groups. The system uses the Effective Date on each calculation group to determine which group(s) to execute. If more than one version of a calculation group is in effect during the billing period, the system may prorate the calculations, based on the configuration of the rate schedule.

- The rate engine creates a separate bill segment calculation header for each rate version calculation group.
- Calculation rules within each group are executed according to the Sequence configured within the group.

3. Finally, the system executes the calculation rules defined within Post-Processing Calculation Groups.

- The rate engine creates a separate bill segment calculation header for post-processing calculations.
- Calculation groups are executed in order according to the Sequence defined on the rate schedule.
- Calculation rules within each group are executed according to the Sequence configured within the group.

Additional Information.

Copyright © 2026, Oracle and/or its affiliates.

- After every rate version calculation group is processed, the rate application returns the calculation details to the process that called it. This point is important as it means that rate application executes in memory (i.e., it does NOT insert the calculation details on the database, rather, it returns them to whatever process called rate application). Because rate application executes in memory, it can be used to perform billing calculations in many parts of the system. For example:
- Both online and batch billing use rate application to calculate the charges that are eventually saved on the bill's bill segments.
- The Rate Check transaction calls rate application when you want to check a rate real-time. When you use this transaction, you enter the consumption that's passed into rate application and then rate application shows you the calculation details that result.
- Plug-ins exist that call rate application to estimate bill segments when adequate billing history doesn't exist. For example, the BUDCALC-PH budget recommendation algorithm calls rate application to calculate a budget amount for a customer without adequate history. Another example is the DEPRECOM-MBT deposit recommendation algorithm. This plug-in calls rate application to calculate a recommended deposit amount for a customer without adequate history.

Understanding Calculation Groups and Rules

This section provides descriptions of calculation groups and calculation rules, and describes how to identify components of rates that can be defined using calculation rules and groups.

- About Calculation Groups and Rules

About Calculation Groups and Rules

This section includes descriptions of calculation groups and calculation rules.

- About Calculation Groups
- About Calculation Rules

About Calculation Groups

Calculation groups are collections of calculation rules that are applied to calculate quantities for billing and billing charges for service agreements.

As noted previously, calculation groups are associated with specific rate schedules in one of three ways:

- **Pre-Processing Calculation Groups**: These are generally used to derive service quantities or massage measured quantities. The following illustrate some use cases of when a pre-processing calculation group might be needed:
  - When a rate contains prices for things that aren't measured directly by the customers' meters. For example, if the rate calculates a charge based on a service agreement's contracted quantity.
  - When a meter measures something in a unit of measure that has to be converted into a different unit of measure before it can be rated. For example, a meter measures gas in cubic feet, but the rate contains prices in therms.
    During billing, a separate bill segment calculation header is produced for each Rate Version Calculation Group used to calculate charges. An additional calculation header

Copyright © 2026, Oracle and/or its affiliates.

is added if Pre-Processing Calculation Groups create bill calculation lines and by default, all these calculation lines are linked to a single calculation header. Before invoking the rate's Pre-Processing Calculation Groups, the rate engine processor builds the initial service quantity (SQ) collection based on the item collection and final values in the read collection.

- **Rate Version Calculation Groups**: These are the calculation groups used to create a bill segment for the service agreement. Calculation rules that support rate version proration such as Service Quantity, Stepped Service Quantity, Item Type, Flat Charge, Maximum and Minimum Charge, must be executed only in a rate version calculation group.
  When the system creates a bill segment for a service agreement, it checks if multiple Rate Version Calculation Groups are in effect during the bill period. If so, it uses the rate's **Allow RV Proration** and **RV Selection Date** values to determine if it should prorate the various Rate Version Calculation Groups, or if it should pick one of the Rate Version Calculation Groups effective during the bill period. Rate Version Calculation Groups are executed after the rate's pre-processing calculation groups.
  - **Effective Date**: defines the date on which each rate version calculation group becomes effective.
  - **Description on Bill**: Specifies the description of the rate version calculation group on the bill. This description is saved on bill calculation details created using this rate version calculation group. Substitution characters may be used to substitute variables into the description during billing. Refer to **How To Use Description on Bill** in the Defining Calculation Rules chapter for a discussion of the various substitution variables.
- **Post-Processing Calculation Groups**: These are similar to Pre-Processing Calculation Groups, except that these are executed after the Rate Version Calculation Groups. The intended types of functionality include applying taxes on calculated charges, or deriving more complex GL distribution.
  Calculation rules that support rate version proration such as Service Quantity, Stepped Service Quantity, Item Type, Flat Charge, Maximum and Minimum Charge, are not designed to be executed in post and pre-processing calculation groups.

  During billing, a separate bill segment calculation header is produced for each Rate Version Calculation Group used to calculate charges. An additional calculation header is added if Post-Processing Calculation Groups create bill calculation lines and by default, all these calculation lines are linked to a single calculation header.

Calculation groups can also be referenced by "Execute Calculation Group" calculation rules. See "Execute Calculation Group Rules - Nesting Calculation Rules" in About Calculation Rules for more information.

About Calculation Rules

Calculation rules are standard and custom rules that perform calculations used when calculating quantities for billing and billing charges for a customer's service agreements.

Calculation rules are created for a specific calculation group. For example, if you were configuring two calculation groups and both included a specific calculation rule, you would need to create two instances of the calculation rule, one for each group.

Attributes used to define calculation rules typically include the following:

- **Basic Information**: Basic information about the calculation rule, including its name and description, the calculation group to which the rule belongs, the sequence of the rule within the group, the description that appears on the bill, and other information. This information is standard for most calculation rules.

Copyright © 2026, Oracle and/or its affiliates.

- **Parameters**: The parameters used by the rule. Parameters are specific to each rule. See Base Package Calculation Rule Descriptions for details about the calculation rules included with the base package.

Execute Calculation Group Rules - Nesting Calculation Rules

Execute Calculation Group rules are used to define business logic to allow reference to a separated specified calculation group. This allows rules that are used frequently to be bunched under a single calculation group, which can be referenced / called by other rules as needed. For example, if a set of standard rules should be applied to all rate schedules, this set of rules can be configured as part of a calculation group, which is referenced by a single calculation rule of this type. Execute Calculation Group rules use the following options:

- **Referred Calculation Group**: defines the calculation group referenced by the rule.
- **Characteristics**: defines additional calculation rule attributes that may be used in rate processing or for reporting purposes. A calculation rule's characteristics are copied onto the resultant bill lines for those characteristic types that are allowed on bill segment calculation lines.

> **Note**
>
> There is a limit of 99 calculation groups that can be called from a calculation group defined directly on a rate schedule or via related child calculation groups when using Execute Calculation Group calculation rules.

Eligibility Criteria

Calculation rule eligibility criteria are user-definable conditions that could cause a given calculation rule to be applied or skipped. This can involve the evaluation of some attribute of the service agreement, or service point, or something else entirely.

A calculation rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence. Each eligibility criteria uses the following settings:

- **Criteria Comparison**: defines the specific comparison for the criteria, based on the following:
  - **Criteria Field Type**: the means used to retrieve the value of the criteria field (algorithm, bill factor, calculation line, characteristic, contract quantity, service quantity, or historic service quantity)
  - **Comparison Operator**: the operator used in the comparison
  - **Comparison Value**: a user-specified value to which the retrieved value is compared (using the comparison operator)
- **Comparison Results**: defines how the rule should behave, based on the results of the comparison defined under Criteria Comparison:
  - **If True**: the action to take if the comparison is true
  - **If False**: the action to take if the comparison is false
  - **If Insufficient Data**: the action to take if there is insufficient data to perform the comparison

Copyright © 2026, Oracle and/or its affiliates.

Calculation Line Category Types

Calculation line category types are used to establish connections between calculation rules. Calculation rules can specify a member category and value to which they belong. When this same category and value are referenced as a "Target Category" by other rules (the rule is applied to the results of rate version calculation rules (defined in the same rate schedule) that belong to the category and value.

For example, calculation rules whose results are subject to taxes might belong to the "Subject to Tax" (TAXABLE) category and "County Tax" value. Apply To calculation rules defined for the same rate schedule with a target category of "Subject to Tax" and "County Tax" values are applied to "Subject to Tax" member rules.

Calculation Line Category Types are defined by the following:

- **Calculation Line Category Type**: The unique identifier of a calculation line category type definition
- **Description**: The description of the calculation line category type.
- **Category Values**: The valid category values for the calculation line category type.

Base Package Calculation Rules

Oracle Utilities Customer Care and Billing includes the following base package calculation rule types that can be used when creating calculation rules:

- **Apply To:** Used to create bill lines that levy charges based on amounts calculated on other bill lines.
- **Calculate Final Reading Values:** Used to determine the final values for readings based on user-defined UOM/TOU/SQI.
- **Calculate Service Quantity:** Used to derive or calculate a service quantity that is eventually inserted into the bill segment SQ collection.
- **Copy Service Quantity From Master Bill Segment:** Used to copy all the entries from the master bill segment SQ collection to the sub bill segment SQ collection.
- **Execute Calculation Group:** Used to call and pass parameters to another calculation group.
- **Flat Charge:** Used to create bill lines that levy service charges and fees that aren't based on how much the customer uses.
- **Item Type:** Used to create bill lines that levy charges based on the number and type of items used by a customer. The system keeps track of items on the service point(s) linked to the customer's service agreement.
- **Math:** Used to derive interval data based on a formula, and apply TOU mappings and/or other operations to the derived data to calculate service quantities
- **Maximum Charge:** Used to create bill lines that levy charges only when the sum of previously calculated lines is more than the maximum charge amount.
- **Minimum Charge:** Used to create bill lines that levy charges only when the sum of previously calculated lines is less than the minimum charge amount.
- **Service Quantity:** Used to create bill lines that levy charges based on some type of consumption.
- **Simple Summary:** Used to create a "subtotal" on the bill, by summarizing bill lines produced by other calculation rules in the same calculation group.

Copyright © 2026, Oracle and/or its affiliates.

- **Stepped Service Quantity:** Used to create bill lines that levy charges based on some type of consumption when different prices are applied to different ranges of consumption (for example, the first 50 kWh are $0.43 per kWh, anything in excess of 50 kWh is charged at $0.71 per kWh)
- **Summary:** Used to create a "subtotal" on the bill. It exists purely for reporting and informational purposes.

See Base Package Calculation Rule Descriptions for details about each of these calculation rules.

Designing Calculation Groups and Rules

Before starting to design calculation groups and rules for your rates, please review the concepts outlines in the **Designing Rate Components** section in the Rates chapter. That section provides a good background for understanding how to break down rates into component pieces that can be modeled by calculation groups and rules.

This section provides guidelines for how to design and plan the calculation groups and rules you will need for your rates, and provides a simple case study of how to define the calculation rules and groups you might use for a set of rates for a small municipal electric company.

- Analyzing Your Rates and Identifying Calculation Rules and Groups
- Reusing Calculation Rules
- Leveraging Eligibility Criteria to Dynamically Apply Calculation Rules
- Case Study: Three Sample Rates

Analyzing Your Rates and Identifying Calculation Rules and Groups

One of the steps you will go through when implementing your rates is to look at the bill lines for your rates and ask yourself what are the variables that cause each line's charges to be calculated the way they are. After you've determined all of the potential bill lines you can start designing your calculation groups and rules. Typically, you create one calculation rule for every line that can appear on the customer's bill. These rules can be grouped in different configurations based on how you want your bills to appear and the needs of your implementation. Recall that bill segment calculation headers are created for Pre-Processing, Rate Version, and Post-Processing calculation groups. In addition, most rules can be configured generate a bill line.

Best Practices

When designing calculation groups and rules, the following general rules apply:

- **Pre-Processing Calculation Groups**. Perform any initial processing or conversions that should occur before the rate version calculation group. This can include:
  - Calculating the number of days in a billing period (Calculate Service Quantity)
  - Calculating service quantities that are not measured (demand, kVa, etc.) (Calculate Final Reading Values, Calculate Service Quantity, Math)
  - Validating that consumption is within tolerances set by the rate (residential limits, etc.) (Calculate Service Quantity, with eligibility criteria)
- Calculations processed by calculation rules should be performed by **Rate Version Calculation Groups**. This includes most charge calculations, including:

Copyright © 2026, Oracle and/or its affiliates.

  - Charges based on service quantities (Service Quantity, Stepped Service Quantity)
  - Customer charges (Flat Charge)
  - Minimum/maximum charges (Minimum Charge, Maximum Charge)
  - Discounts (Apply To)
  - Taxes (Apply To)
  - Summaries (Summary, Simple Summary)
- Additional processing should be performed by **Post-Processing Calculation Groups**. This can include:
  - Allocating consumption/revenue (energy revenue vs. demand revenue) (Service Quantity)

Example Configurations

Let's look at a couple of different possible configurations for a simple electric rate that includes the following:

- Customer Charge
- Service Charge (price per kWh)
- Minimum Charge
- Discount

This rate could be set up in the following ways:

- A single Rate Version calculation group that contains a calculation rule for each of the above
- A set of Rate Version calculation groups as follows:
  - A calculation group for "flat" charges, such as the Customer Charge
  - A calculation group for consumption-based charges (including the Service Charge and Minimum Charge).
  - A calculation group for discounts or other charges applied to the total of other charges (which would include the Discount)

There are no hard and fast rules concerning how calculation rules should be grouped. How this is done should be based on the specific needs of the implementation but it's a good idea to look for ways to keep your configuration simple for maintenance and upkeep purposes. This includes changes to existing rates as well as adding/removing rates over time.

Reusing Calculation Rules

One of the steps you will go through when implementing your rates is to look at the bill lines for your rates and ask yourself what are the variables that cause each line's charges to be calculated the way they are. After you've determined all of the potential bill lines you can start designing your calculation groups and rules. Typically, you create one calculation rule for every line that can appear on the customer's bill. These rules can be grouped in different configurations based on how you want your bills to appear and the needs of your implementation. Recall that bill segment calculation headers are created for Pre-Processing, Rate Version, and Post-Processing calculation groups. In addition, most rules can be configured generate a bill line.

Copyright © 2026, Oracle and/or its affiliates.

Identifying Calculation Rules that can be Shared/Reused

As you analyze the bill lines used by your rates, you should look for bill lines that utilize similar or identical variables or calculations. These calculations are candidates for reuse. For example, if all of your residential rates offer a discount for early payment, you would want to create a single rule to calculate that discount, and then use it each of the rates to which it applies.

How to Reuse Calculation Rules

There are two approaches that can be used to reuse calculation rules.

- **Encapsulating Rules in a Calculation Group**
  This approach involves grouping rules that can be applied to more than one rate into a single calculation group. You would use this approach when you identify charges and other components in your rates that apply to multiple rates, but that can be grouped together. Taxes for multiple types of jurisdictions (for example, State, County, and City) would be suitable for this approach.

  To use this approach, you simply create a calculation group that contains the rules to be applied to more than one rate. For the tax example, you might create the following group and rules:

  Calculation Group: Taxes, which includes:

  Calculation Rule: State Tax (Apply To)

  Calculation Rule: County Tax (Apply To)

  Calculation Rule: City Tax (Apply To)

  This "Taxes" calculation group can be included as a Rate Version Calculation Group in each of the rates to which the taxes apply. This calculation group could also be referenced by an "Execute Calculation Group" rule, as outlined below.

- **Using "Execute Calculation Group" Rules**
  This approach involves creating a calculation group that contains one or more rules, which is then referenced by another "Execute Calculation Group" rule. This approach can be used when you identify a charge or calculation that could be applicable within one or more calculation groups. For example, if multiple rates involve calculating peak demand (or some other quantity that is not measured by your customers' meters), you could create a "Calculate Demand" rule and include it in a group that is then referenced by an "Execute Calculation Group" rule.

  To use this approach, you create a calculation group that contains the calculation rules to be used within multiple calculation groups. You then create an "Execute Calculation Group" rule that references the group. Last, you include this "Execute Calculation Group" rule in the calculation groups to which it applies. For the calculate demand example, you might create the following:

  Calculation Rule: Calculate Demand (Execute Calculation Group), which references:

  Calculation Group: Calculate Demand, which includes:

  Calculation Rule: Calculate Demand (Calculate Service Quantity)

  The "Calculate Demand" Execute Calculation Group rule can be included in any calculation group used to calculate service quantities that includes calculating peak demand.

  "Execute Calculation Group" rules can be used with any calculation group. To return the taxes example from above, if you wanted to include your "Taxes" calculation group within other groups, you could use the following:

Copyright © 2026, Oracle and/or its affiliates.

Calculation Rule: Taxes (Execute Calculation Group), which references:

Calculation Group: Taxes, which includes:

Calculation Rule: State Tax (Apply To)

Calculation Rule: County Tax (Apply To)

Calculation Rule: City Tax (Apply To)

See "Execute Calculation Group Rules - Nesting Calculation Rules" in About Calculation Rules for more information about Execute Calculation Group rules.

Leveraging Eligibility Criteria to Dynamically Apply Calculation Rules

Another tool to consider when creating your calculation groups and rules is Eligibility Criteria. Eligibility criteria allow you to define rules that determine the conditions under which a calculation rule is applied or not.

For example, if you need to calculate demand for one or more of your rates, you might create separate calculation rules to calculate demand based on interval values of different units of measure (kWh vs kVarh for instance). You could create rules for each and include them in a single "Calculate Demand" calculation group, but assign eligibility criteria to each such that they are only executed when processing for the specified unit of measure.

See Eligibility Criteria in About Calculation Rules for more information about eligibility criteria and how they can be used with calculation rules. In addition, refer to **The Big Picture of Rate Component Eligibility Rules** in the Rates chapter for additional background on eligibility criteria.

Case Study: Three Sample Rates

Let's look at a sample set of rates, and how they might be implemented. We'll also look at which components of these rates can be shared.

Rates

The following rates are electric rates offered by a small municipal electric company.

Copyright © 2026, Oracle and/or its affiliates.

Rate	Domestic Rate A	Commercial Rate B	Power Rate C
Cost of Service	Stepped charges, as follows: $3.08 for the first 10 kWh or less, plus 19.23 cents per kWh for the next 40 kWh, plus 15.44 cents per kWh for the next 150 kWh, plus 14.93 cents per kWh for the next 300 kWh, plus 14.71 cents per kWh for all kWh in excess of 500 kWh	Stepped charges, as follows: $2.95 for the first 10 kWh or less, plus 19.23 cents per kWh for the next 40 kWh, plus 18.85 cents per kWh for the next 50 kWh, plus 16.86 cents per kWh for the next 400 kWh, plus 15,45 cents per kWh for the next 3,000 kWh, plus 14.87 cents per kWh for all kWh in excess of 3,500 kWh.	1. Customer Charge:$52.00 2. Demand: $11.44 per KW 3. Energy: $11.28 cents per KWH **Billing Demand** The Billing Demand for any month shall be the greater of the following: 1. The maximum 15 minute measured demand during the month as measured in kilowatts, or 2. 70% of the maximum 15 minute measured demand as so determined during the preceding 11 months.
Discount	Discount of ½ cent per kilowatt-hour after the first 10 kilowatt-hours, if paid on or before the discount expiration date. For qualifying elderly residents, an additional 10% discount is allowed on the basic rate if paid on or before the discount expiration date and no arrears exist on their accounts.	Discount of ½ cent per kilowatt-hour after the first 10 kilowatt-hours, if paid on or before the discount expiration date.	Discount of 10% exclusive of purchased power and fuel adjustment charge paid on or before the discount expiration date.
Purchased Power and Fuel Adjustment	An upward or downward adjustment applied to all kWh sold, based on the total cost of purchased power and fuel compared to the portion of that cost contained in the base rate (11.615 cents/kWh). The purchase power and fuel adjustment shall be rounded to the sixth decimal place and shall be applied to all kilowatt-hours sold in the following month.	See Domestic Rate A	See Domestic Rate A
Misc.	N/A	N/A	**Minimum Bill** The Customer Charge
Taxes	State, County, and City Taxes	State, County, and City Taxes	State, County, and City Taxes

Copyright © 2026, Oracle and/or its affiliates.

Calculation Rules

Now let's look at these same rates in terms of how we might implement them using calculation rules. The table below lists the type of calculation rule used for each item on the rate.

Rate	Domestic Rate A	Commercial Rate B	Power Rate C
Cost of Service	Stepped Charges: • Stepped Service Quantity calculation rule • Minimum Charge calculation rule (note that the first tier is a fixed price for the first 10 kWh or less)	Stepped Charges: • Stepped Service Quantity calculation rule • Minimum Charge calculation rule (note that the first tier is a fixed price for the first 10 kWh or less)	Customer Charge: • Flat Charge calculation rule Demand: • Service Quantity calculation rule • Math calculation rule (to calculate demand from interval values)* Energy: • Service Quantity calculation rule *Could be included in either a Pre-Processing or Rate Version Calculation Group
Discount	Discount: • Apply To calculation rule Elderly Discount: • Apply To calculation rule	Discount: • Apply To calculation rule	Discount: • Apply To calculation rule
Purchased Power and Fuel Adjustment	PPF Adjustment: • Service Quantity calculation rule **Note:** The PPF adjustment rate would be defined as a Bill Factor, updated monthly based on the prior month's total cost.	PPF Adjustment: • Service Quantity calculation rule	PPF Adjustment: • Service Quantity calculation rule
Misc.	N/A	N/A	Minimum Bill: • Minimum Charge calculation rule
Taxes	State, County, and City Taxes: • Apply To calculation rule (one for each type of tax)	State, County, and City Taxes: • Apply To calculation rule (one for each type of tax)	State, County, and City Taxes: • Apply To calculation rule (one for each type of tax)

Shared Components

Now, let's look at the components of these rates that can be shared.

Discounts: Rates A and B both use the same discount, so we could create a single "AB Discount" calculation rule that could be used with both rates. Note: Either this could be in its

Copyright © 2026, Oracle and/or its affiliates.

own calculation group that is listed among the Rate Version Calculation Groups for each rate, or the group could be referenced by an "Execute Calculation Group" rule which in turn would be included in one of the other Rate Version Calculation Groups used in each rate.

Purchased Power and Fuel Adjustment: Rates A, B, and C all use the same adjustment for purchased power and fuel, so we could create a single "PPF Adjustment" calculation rule that could be used with all three rates. Again, this calculation rule could be in its own calculation group listed on the rate, or in a group referenced by an "Execute Calculation Group" rule included in another group.

Taxes: Rates A, B, and C are all subject to the same taxes, so we could create calculation rules for each type of tax (State, County, and City) in a single group that could be used with all three rates. Again, these calculation rules could be in their own calculation group listed on the rate, or in a group referenced by an "Execute Calculation Group" rule included in another group.

The Big Picture of Calculation Rule Eligibility Rules

You can use eligibility rules to define the criteria when a calculation rule should be applied (or skipped); for example,

- You may have a calculation rule that generates an employee discount. This calculation rule may have eligibility criteria that requires the person to have a given characteristic type and value (this assumes you use a person characteristic to identify employees) before the calculation rule is applied.
- You may have a calculation rule that applies a surcharge if the customer uses 25% more than they used in the same period in prior year.

> **Note**
>
> **Eligibility rules are optional.** If you add eligibility rules to a calculation rule, the calculation rule will be skipped or applied per the instructions in the rules. If you don't add eligibility rules to a calculation rule, however, the calculation rule will be executed – this is because calculation rules are eligible by default

The topics in this section describe eligibility rules.

- A Rate Component is Eligible by Default
- Defining Logical Eligibility Criteria
- Examples of Calculation Rule Eligibility Rules

A Rate Component is Eligible by Default

If you don't want to setup eligibility rules, you don't have to. If you don't specify eligibility rules on a calculation rule, it's eligible and will be processed by rate application.

Defining Logical Eligibility Criteria

Copyright © 2026, Oracle and/or its affiliates.

When you setup an eligibility criterion, you must define three things:

- The field to be compared
- The comparison method
- The comparison results

Criteria Comparison

You have the following choices in respect of identifying the **Criteria Field Type**:

- **Algorithm.** You can execute an algorithm to retrieve a field value from someplace else in the system. This is a very powerful feature, but it's not terribly intuitive. We'll present an example later in this section to illustrate the power of this approach.
- **Bill Factor.** You can retrieve a bill factor value to use as a field value in the comparison.
- **Calc Line.** You can retrieve a calculation line amount to use as a field value. The value is derived by summing calculation lines defined as members of a target category.
- **Characteristic.** You can retrieve a characteristic value linked to any of the following:
  - The service agreement being billed
  - The service agreement's account
  - The main person linked to the service agreement's account
  - The characteristic premise linked to the service agreement
  - One of the service points linked to the service agreement
  - In addition, you can retrieve a characteristic value from the Characteristic Collection where values are derived while the rate is being calculated (characteristic values can be created by calculation rules)
- **Contract Quantity.** You can retrieve a contract quantity for the service agreement.
- **Service Quantity.** You can retrieve the value of a given service quantity from the SQ collection based on a specified UOM / TOU / SQI.
- **Historical Quantity.** You can derive a value based on the service agreement’s historical bill segment service quantities for the specified UOM / TOU / SQI.

You have the following choices in respect of identifying the **Comparison Method**:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value

Comparison Results

Once you have this information and have applied your comparison, you must define what action should be taken. You must specify what to do if the results are True, False, or Insufficient Data.

- **If True or False.** You have the option to apply the rule or not, or check the next condition.
- **If Insufficient Data.** You have the option apply the rule or not, check the next condition, or generate an error.

See Examples of Calculation Rule Eligibility Rules for illustrations of these rules.

Examples of Calculation Rule Eligibility Rules

The topics in this section provide examples about how to setup calculation rule eligibility rules.

Copyright © 2026, Oracle and/or its affiliates.

- A Calculation Rule with a Service Quantity Comparison
- A Calculation Rule with a Characteristic Lookup and Consumption Comparison
- A Calculation Rule with a Discount for Being on Autopay

A Calculation Rule with a Service Quantity Comparison

Imagine a calculation rule that is only eligible if the customer is a low-income customer and has used more than 1,000 kWh. This calculation rule would need the following eligibility rules:

- Customer is a low income customer
- The customer has used more than 1,000 kWh during the bill period

These rules will look as follows:

Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Main person characteristic: Characteristic type = Low income indicator	= YES	Check next condition	Do Not Apply Rule	Do Not Apply Rule
20	SQ value: UOM = KWH	>= 1,000	Apply Rule	Do Not Apply Rule	Error

The criteria are similar - they retrieve a value and compare it to something:

The first criterion is fairly simple:

- **Field to Compare.** We indicate that the value of the Low Income Indicator characteristic is kept on the main person linked to the account should be retrieved. We assumed that this characteristic type has values of YES and NO.
- **Comparison Method and Results.** We indicate that if the person has a characteristic value of YES, then we should check the next condition; otherwise, we indicate if the eligibility rule is false or there is insufficient data, the calculation rule is skipped.

TThe second criterion works similarly:

- **Field to Compare.** We indicate the service quantity (SQ) that holds the total amount of KWH that should be retrieved.
- **Comparison Method and Results.** We simply compare the value to 1,000. If the SQ is greater than or equal to 1000, the eligibility rule is True and the calculation rule is applied. If the SQ is less than 1000, the eligibility rule is false and the calculation rule is skipped. If there is insufficient data, there is an error generated.

A Calculation Rule with a Characteristic Lookup and Consumption Comparison

Imagine a calculation rule that is eligible for customers who have signed up for a consumption reduction program and have low consumption. This would require the following criteria:

- Check that the customer has signed up for the reduction program.
- Check to see that the customer’s consumption is lower than the threshold.

Copyright © 2026, Oracle and/or its affiliates.

These rules will look as follows:

Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Check if the customer has signed up for the reduction program	= Y	Check next condition	Do Not Apply Rule	Do Not Apply Rule
20	Check if consumption is less than the threshold	< = 100 Kilowatt	Apply Rule	Do Not Apply Rule	Do Not Apply Rule

The criteria are similar - they retrieve a value and compare it to something:

The first criterion works as follows:

- **Field to Compare.** We indicate the value of the reduction program characteristic type (CI_CLMSM) is found on the service agreement. We assumed that this characteristic type has values of Y and N.
- **Comparison Method and Results.** Comparison Method and Results. We indicate that if the service agreement has a characteristic value of Y, then we should check the next condition. If the condition is false or insufficient data, the calculation rule is skipped.

The second criterion works as follows:

- **Field to Compare.** We indicate the service quantity (SQ) that holds the total amount of KW that should be retrieved.
- **Comparison Method and Results.** We indicate that we want to compare the service quantity to the threshold (100 kW). If the service agreement's bill segment has a service quantity that is less than or equal to 100 kW, it is True and the calculation rule is applied. If the condition is false or insufficient data, the calculation rule is skipped.

A Calculation Rule with a Discount for Being on Autopay

Imagine a calculation rule that is only eligible if the current customer has signed up for Autopay. This calculation rule would need the following eligibility rule:

- Customer has Autopay

This rule will look as follows:

Seq	Field to Compare	Comparison Method	If TRUE	If FALSE	If Insufficient Data
10	Algorithm: check if customer has Autopay	= Y	Apply Rule	Do Not Apply Rule	Do Not Apply Rule

The criterion calls an algorithm that performs a logical comparison.

The criterion works as follows:

Copyright © 2026, Oracle and/or its affiliates.

- **Field to Compare.** We chose a "field to compare" algorithm that checks if the account associated with the bill segment is on Autopay. If an autopay setup exists, then it returns a value of Y. If not, however, it returns N
- **Comparison Method and Results.** We simply compare the value returned by the algorithm to Y and indicate the appropriate response. If the condition is true, the calculation rule is applied. If the condition is false or insufficient data, the calculation rule is skipped.

> **Note**
>
> Refer to CI_CHK_APAY for more information about this algorithm.

Working with Calculation Groups and Rules

This section outlines how users work with calculation groups and calculation rules.

- Defining Calculation Groups
- Searching and Maintaining Calculation Groups
- Defining Calculation Line Category Types
- Maintaining Calculation Line Category Types
- Defining Calculation Rules
- Searching and Maintaining Calculation Rules
- Defining Calculation Rule Eligibility Criteria

Defining Calculation Groups

Use this procedure to define calculation groups.

- Select **Menu**, and then **Rates**, and then **Calculation Group**, and then **Add.**
- Enter a **Name** for the calculation group.
- Enter a short **Description** and **Detailed Description** for the group.
- Click **Save**.

> **Note**
>
> You can also add a calculation group directly from the Rate Schedule portal. The **Add Calculation Group** action on the Pre-Processing Calculation Groups, Rate Version Calculation Groups and Post-Processing Calculation Groups sections allows you to create the calculation group and link the new group to the Rate Schedule.

To add a calculation rule to the calculation group, click the **Add Rule** link in the **Calculation Rules List** zone title bar. See Defining Calculation Rules for more information about creating calculation rules.

You can now create calculation rules for this group.

See About Calculation Groups for more information about calculation groups.

Copyright © 2026, Oracle and/or its affiliates.

Searching and Maintaining Calculation Groups

Use this procedure to search for and view a calculation group.

- Select **Menu**, and then **Rates**, and then **Calculation Group**, and then **Search.**
- Enter your search criteria.
  You can search by **Calculation Group** name or **Description**.
- Click **Refresh**.
- In the search results list, click the **Description** link for the calculation group you want to view or edit.
  The selected calculation group is displayed in the **Calculation Group** portal. You use this portal to maintain calculation groups. This portal includes the following zones:
  - **Calculation Group**: Defines basic information about calculation group
  - **Calculation Rules List**: lists the calculation rules belonging to the group. Eligibility criteria referenced by a calculation rule are listed indented beneath each rule. If a calculation rule in turn references a calculation group, the calculation group executed by the rule is displayed.
  - **Referencing Rates Schedules List**: lists the rates schedules that reference the group
  - **Referencing Calculation Rules List**: lists the calculation rules that reference the group

Use this procedure to maintain a calculation group:

- Select **Menu**, and then **Rates**, and then **Calculation Group**, and then **Search** to navigate to the Calculation Group Query portal.
- Search for and select the appropriate calculation group.
- Click **Edit**, **Delete**, or **Duplicate** as appropriate.
- To add a calculation rule to the calculation group, click the **Add Rule** link in the **Calculation Rules List** zone title bar. See Defining Calculation Rules for more information about creating calculation rules.
- To update the sequence values for all calculation rules in the group, click the **Resequence All Rules** link in the **Calculation Rules List** zone title bar.
  - Edit the sequence values for all rules by specifying new **Start Sequence** and **Increment By** values**.**
  - Click **OK** to update the calculation rules with new sequence numbers.
- To Edit or Delete a specific calculation rule in the group, select the **Edit** or **Delete** icon in the Calculation Rules List zone as appropriate.
- To Delete one or more of the calculation rules in the group:
  - Select one or more of the calculation rules you want to delete.
  - Click **Delete Calc Rules.**
  - Click **OK** to delete the selected calculation rules
- To update the bill print details for one or more of the calculation rules in the group:
  - Select one or more of the calculation rules for which you want to change the bill print details.

Copyright © 2026, Oracle and/or its affiliates.

- Click **Update Bill Print.**
- Edit the bill print details for the selected calculation rules.
- Click **Save** to update the selected calculation rules with the new bill print details.

- To change the sequence values for one or more of the calculation rules in the group:
  - Select one or more of the calculation rules for which you want to change the sequence values.
  - Click **Resequence Rules.**
  - Edit the sequence values for all the selected calculation rules.
  - Click **Save** to update the selected calculation rules with the new sequence values.

See About Calculation Groups for more information about calculation groups.

Defining Calculation Line Category Types

Use this procedure to define calculation line category types.

- Select **Menu**, and then **Rates**, and then **Calculation Line Category Type**, and then **Add.**
- Enter a **Category Type** name for the category type.
- Enter a short **Description** and **Detailed Description** for the category type.
- Define the **Category Value** and **Description** for each value for the category type.
- Click **Save**.

You can now reference this category and its values as either the Member Category or Target Category on calculation rules.

See "Calculation Line Category Types" in About Calculation Rules for more information about calculation line category types.

Maintaining Calculation Line Category Types

Use this procedure to search for and maintain calculation line category types.

- Select **Menu**, and then **Rates**, and then **Calculation Line Category Type**, and then **Search.**
  The **Calculation Line Category Type List** zone displays the calculation line category types defined in the system.
- Click the broadcast icon for the calculation line category type you wish to view and/or edit. The selected calculation line category type is displayed in the **Calculation Line Category Type** zone.
- Click **Edit**, **Delete**, or **Duplicate** as appropriate.

See "Calculation Line Category Types" in About Calculation Rules for more information about calculation line category types.

Defining Calculation Rules

Use this procedure to define calculation rules.

- Select **Menu**, and then **Rates**, and then **Calculation Rule**, and then **Add.**

Copyright © 2026, Oracle and/or its affiliates.

- Select the **Calculation Group** to which the rule will belong.
- Select the **Calculation Rule BO** for the calculation rule.
  Refer to Base Package Calculation Rule Descriptions for detailed descriptions of the available calculation rules and their associated business objects.
- Enter name for the **Calculation Rule**.
- Enter the **Sequence** for the rule. This designates the order in which the rule will be executed within its parent calculation group.
- Enter a short **Description** and **Detailed Description** for the rule.
- Enter or select other parameters for the rule as appropriate.
  Refer to Base Package Calculation Rule Descriptions for detailed descriptions of the available calculation rules and the parameters used by each.
- Click **Save**.

See About Calculation Rules for more information about calculation rules.

- How to Use Description on Bill

How to Use Description on Bill

Enter the verbiage to appear on the bill segment in the **Description On Bill**. Turn on **Print** if the verbiage should appear on the printed bill.

> **Note**
>
> **Suppressing calculation details on the printed bill.** Only calculation rules that are used to calculate intermediate values are typically suppressed on a bill. For these types of calculation rules, we recommend using **Description on Bill** to show users how the line was calculated, but select No from the **Print** dropdown menu. This way, users will see the calculation details on the Bill Segment - Calc Lines window, but the customer won't see it on their bill.

Many lines that appear on a bill do not contain static verbiage. Rather, the printed lines contain information from the calculation process. For example, the bill line for state tax would typically contain the tax percent.

You control exactly what is substituted in the bill line by entering one or more of the following substitution variables in the **Description on Bill** field:

> **Note**
>
> If none of the following substitution characters satisfy your need, there is a plug-in option on the calculation rule business object that allows customer modified variables. Algorithms can be attached to the business object as a Build Line Description system event.

Substitution Character	What It Does

Copyright © 2026, Oracle and/or its affiliates.

%A	Replaces the substitution character with the amount of the bill line. Please be aware of the following: • If the calculation rule isn't designated as being "for calculation purposes only" (FCPO), all trailing zeroes up to the decimal position of the rate's currency will be removed. If the calculation rule is designated as being "for calculation purposes only" (FCPO), the number of decimal places as defined in the calculation rule's precision will be shown. • All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place. • The Symbol on the Currency control table will prefix the value. • The position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process).
%B	Replaces the substitution character with the begin/start date covered by the bill calculation line. If the charge is proratable, %B may be used in conjunction with %E to display the period covered by the bill calculation line.
%C	Replaces the substitution character with the number of items being billed. Leading zeroes are suppressed.
%D	Replaces the substitution character with message 10101 from message category 11109. This message is maintained in the standard message table. We recommend this message be set to "(prorated for %1 of %2 days)" where %1 will be substituted with the number of days of prorated consumption charged on the bill line and %2 will be the "normal" days associated with the rate's prices. This message will only be substituted when the calculation rule is prorated. Note, neither %1 nor %2 are required. Either this character OR %F should be used when you desire to substitute the price / percent into the bill line. If the charge is proratable, %D should be used in conjunction with %R to print an appropriate message when the charges are prorated.
%E	Replaces the substitution character with the end date covered by the bill calculation line. If the charge is proratable, %E may be used in conjunction with %B to display the period covered by the bill calculation line.

Copyright © 2026, Oracle and/or its affiliates.

%F	Replaces the substitution character with message 10102 from message category 11109. This message is maintained in the standard message table. We recommend this message be set to "(prorated by %1)" where %1 will be substituted with the factor used to prorate the charge. This message will only be substituted when the calculation rule is prorated. Note, %1 is not required. If the charge is proratable, %F should be used in conjunction with %I to print an appropriate message when the charges are prorated.
%H	Replaces the substitution character with the high value defined on the calculation rule used to calculate the specific stepped charge on the bill line.
%I	Replaces the substitution character with the value specified in the calculation rule / bill factor. If the value is a percent, • all leading and trailing zeroes will be suppressed, • and a "%" will be suffixed to the resultant value. If the value is not a percent, • all trailing zeroes up to the decimal position of the rate's currency will be removed, • and all leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place, • and note that the position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process). Either this character OR %R should be used when you desire to substitute the price / percent into the bill line. Using %I will cause the value as specified in the calculation rule to be substituted regardless of whether the charges are prorated. If the charge is proratable, %I should be used in conjunction with %F to print an appropriate message when the charges are prorated.
%L	Replaces the substitution character with the low value defined on the calculation rule used to calculate the specific stepped charge on the bill line.

Copyright © 2026, Oracle and/or its affiliates.

%M	Replaces the substitution character with the item type defined on the calculation rule used to calculate the specific item type charge on the bill line. This is only applicable to calculation rule-based rates.
%O	Replaces the substitution character with the description of the time-of-use (TOU) code being billed on this line
%Q	Replaces the substitution character with the service quantity being billed. Leading zeroes are suppressed. The decimal places shown are specified in a field on the Unit Of Measure control table. If you wish to suppress trailing zeroes after the decimal, you must set up the Allow Consumption Zero Suppression In Bill Description option type on the Financial Transaction Options Feature Configuration.
%R	Replaces the substitution character with the value specified in the calculation rule / bill factor. Either this character OR %I should be used when you desire to substitute the price / percent into the bill line. Using %R causes the prorated price / percent to be printed in the bill line (%I causes the price / percent as specified in the rate to be printed during proration situations). If the charge is proratable, %R should be used in conjunction with %D to print an appropriate message when the charges are prorated.
%S	Replaces the substitution character with the summed amount. All trailing zeroes up to the decimal position of the rate's currency will be removed. All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place. The Symbol on the Currency control table will prefix the value. If the value is negative, a "-" will prefix the value. This character should only be used with Minimum, Apply To, and Summary calculation rules.
%T	Replaces the substitution character with message 10103 from message category 11109. This message is maintained in the standard message table. We recommend this message be set to "(%1% exempt)" where %1 will be substituted with the tax exemption percent as specified in the service agreement. This message will only be substituted when the calculation rule's bill factor is eligible for tax exemptions AND the customer is exempt from the tax. Note, %1 is not required.

Copyright © 2026, Oracle and/or its affiliates.

%V	Replaces the substitution character with the "C1-ADDSC" characteristic ad hoc value retrieved by the calculation rule used to calculate the charge on the bill line.
Lines that don't print	If the bill line is not printed (as defined by the Print option), message 10104 from message category 4 will be appended to the bill line. This message is maintained in the standard message table. We recommend this message be "NOT SEEN BY CUSTOMER!" This message appears in the bill line that appears in the billing windows but does not appear on the customer's bill.

The following are examples of the content of common **Description on Bills** along with how the line would look on the customer's bill:

Description on Bill On Calculation Rule	How It Looks On A Bill
%C lamp pole(s) at %I per pole %F	1,234 lamp pole(s) at $5.00 per pole (prorated by .3333)
%C lamp pole(s) at %R per pole %D	1,234 lamp pole(s) at $1.50 per pole (prorated for 10 of 30 days)
First %Q kWh at %I per kWh %F	First 100.32 kWh at $0.12349 per kWh (prorated by .3333)
First %Q kWh at %R per kWh %D	First 100.32 kWh at $0.12349 per kWh (prorated for 10 of 30 days)
%Q kWh at %I per kWh	100.32 kWh at $0.12349 per kWh
State sales tax %R %T	State sales tax 6.25% (50% exempt)
Minimum charge of %I %F	Minimum charge of $10.00 (prorated by .3333)
Subtotal of %S	Subtotal of $1,123.23
Charges for %O: %Q kW at %I per kW	Charges for Off Peak Winter: 234.43 kW at $0.05 per kW

The prorated references will only appear if the calculation rule's charges are prorated due to a short or long bill period.

Searching and Maintaining Calculation Rules

Use this procedure to search for and maintain calculation rules.

- Select **Menu**, and then **Rates**, and then **Calculation Rule**, and then **Search.**
- Enter your search criteria.
  Available query options include **Calculation Rule** and **Calculation Rule BO**.
  - When searching by **Calculation Rule**, you can search by **Calculation Group**, **Calculation Rule** name, or **Description**.
  - When searching by **Calculation Rule BO**, you can search by **Business Object**, or **Description**.
- Click **Refresh**.

Copyright © 2026, Oracle and/or its affiliates.

- In the search results list, click the **Description** link for the calculation rule you want to view or edit. To navigate to the rule's calculation group, click the **Calculation Group** link.
  The selected calculation rule is displayed in the **Calculation Rule** portal. You use this portal to maintain calculation rules. This portal includes the following zones:
  - **Calculation Rule**: Defines basic information about calculation rule
  - **Eligibility Criteria List**: Lists optional eligibility criteria for the current calculation rule. Eligibility criteria are used to determine whether a calculation rule should be applied (or skipped) during rate processing.

Use this procedure to maintain a calculation rule:

- Select **Menu**, and then **Rates**, and then **Calculation Rule**, and then **Search** to navigate to the Calculation Rule Query portal.
- Search for and select the appropriate calculation rule.
- Click **Edit**, **Delete**, or **Duplicate** as appropriate.
- To define eligibility criteria for the rule, click the **Add** link in the **Eligibility Criteria List** zone title bar. See "Eligibility Criteria" in About Calculation Rules for more information.

See About Calculation Rules for more information about calculation rules.

Defining Calculation Rule Eligibility Criteria

Use this procedure to define eligibility criteria for a calculation rule.

Prerequisites: You must create a calculation rule before you can define eligibility criteria.

- Click the **Add** link in the **Eligibility Criteria List** zone title bar.
- Enter a **Sequence** number, **Description**, and **Detailed Description** for the criteria.
- Complete the **Criteria Comparison** section as follows:
  - **Criteria Field**: the means used to retrieve the value of the criteria field (algorithm, bill factor, calculation line, characteristic, contract quantity, service quantity, or historic service quantity)
  - **Comparison Operator**: the operator used in the comparison
  - **Value**: a user-specified value to which the retrieved value is compared (using the comparison operator)
- Complete the **Comparison Results** section as follows:
  - **If True**: the action to take if the comparison is true
  - **If False**: the action to take if the comparison is false
  - **If Insufficient Data**: the action to take if there is insufficient data to perform the comparison
- Click **Save**.

See The Big Picture of Calculation Rule Eligibility Rules for more information.

Creating Rate Schedules

This section outlines how users work with rate schedules.

- Defining Rate Schedules

Copyright © 2026, Oracle and/or its affiliates.

- Searching and Maintaining Rate Schedules

Defining Rate Schedules

Use this procedure to define rate schedules.

Prerequisites: You must configure an instance of the **Rate Engine Configuration** master configuration in order to create and/or maintain rate schedules.

- Select **Menu**, and then **Rates**, and then **Rate Schedule**, and then **Add.**
  The Rate Schedule maintenance screen opens. To navigate to the Component-Based Rate Schedule page, click **Switch to Component-Based**.
- Enter a code for the **Rate Schedule**.
- Enter a short **Description** for the rate schedule.
- Select the **Service Type** (Electric Service, Gas Service, etc.) for the rate schedule.
- Select the **Frequency** (monthly, quarterly, etc.) for the rate schedule
- Select the **Currency Code** used with the rate schedule.
- Select whether or not the rate schedule will **Allow Estimates** when performing calculations.
- Select whether or not the rate schedule will **Allow RV Proration** when performing calculations.
- Select the **RV Proration Date**, used if the rate schedule does NOT allow RV proration. Valid options include Accounting Date, Bill End Date, and Bill Start Date.
- Define the **Pre-Processing Calculation Groups** used by the rate schedule. If the calculation groups do not exist yet, use the **Add Calculation Group** action to add each new calculation group and link it to the rate schedule. Refer to the embedded help for more information.

  For each pre-processing calculation group, specify:

  – **Sequence**: The sequence in which the group is executed when the pre-processing groups are processed.

  – **Description**: A description of the calculation group

  – **Calculation Group**: The calculation group.

  Use the **Add/Remove Links** action on each calculation group to maintain the cross-references between the rules in the group. This action includes an ability to maintain a rule's target categories.
- Define the **Rate Version Calculation Groups** used by the rate schedule. If the calculation groups do not exist yet, use the **Add Calculation Group** action to add each new calculation group and link it to the rate schedule. Refer to the embedded help for more information.

  For each rate version calculation group, specify:

  – **Effective Date**: The date on which each rate version calculation group becomes effective

  – **Description on Bill**: Specifies the description of the rate version calculation group on the bill. This description is saved on bill calculation details created using this rate version calculation group. Substitution characters may be used to substitute variables into the description during billing. Refer to How To Use Description on Bill for a discussion of the various substitution variables.

Copyright © 2026, Oracle and/or its affiliates.

- **Calculation Group**: The calculation group.

  Use the **Add/Remove Links** action on each calculation group to maintain the cross-references between the rules in the group. This action includes an ability to maintain a rule's target categories.

- Define the **Post-Processing Calculation Groups** used by the rate schedule. If the calculation groups do not exist yet, use the **Add Calculation Group** action to add each new calculation group and link it to the rate schedule. Refer to the embedded help for more information.

  For each post-processing calculation group, specify:

  - **Sequence**: The sequence in which the group is executed when the post-processing groups are processed.
  - **Description**: A description of the calculation group
  - **Calculation Group**: The calculation group.

  Use the **Add/Remove Links** action on each calculation group to maintain the cross-references between the rules in the group. This action includes an ability to maintain a rule's target categories.

- Define one more recommended **Meter Configuration Types** for meters linked to service agreements (by way of their service points) for the rate schedule.
- Define one or more **Bill Messages** that should appear on bills that use the rate schedule. For each message, specify
  - **Bill Message**: The message that should appear on the bill.
  - The **Start Date** and **End Date** when the bill message should appear on the bill. Leave **End Date** blank if the message should appear indefinitely,
- Click **Save**.

This rate schedule can now be referenced on service agreements.

Searching and Maintaining Rate Schedules

Use this procedure to search for and view a rate schedule.

Prerequisites: You must configure an instance of the **Rate Engine Configuration** master configuration in order to create and/or maintain rate schedules.

- Select **Menu**, and then **Rates**, and then **Rate Schedule**, and then **Search.**
- Enter your search criteria.
  You can search by **Rate Schedule**, **Description**, or **Rate Engine** (Calculation Rule-Based or Component-Based).
- Click **Search**.
- In the search results list, click the **Description** link for the rate schedule you want to view or edit.

> **Note**
>
> Component-based rate schedules will be highlighted by having "(Component-Based)" appended to their listing in the search results.

Copyright © 2026, Oracle and/or its affiliates.

- The selected rate schedule is displayed in the (Calculation Rule-Based) **Rate Schedule** portal. You use this portal to maintain rate schedule. This portal includes the following zones:
  - **Rate Schedule**: Defines basic information about the rate schedule.
  - **Rate Schedule Tree:** Displays the hierarchy of Calculation Groups and Calculation Rules for the Rate Schedule. Only the most recent rate version calculation group is initially displayed. In this zone you can:
    - Click a calculation group or calculation rule link to go to the Calculation Group or Calculation Rule portal for the associated calculation group or calculation rule.
    - Click the Broadcast icon for a calculation rule to open the Cross Reference Rule zone. This zone displays the rate schedule's calculation rules that are members of the broadcast calculation rule's target category.
    - Click the **Show All Rate Versions** button to display all rate versions for the rate schedule. This will also display the **Show Effective Rate Version** button, which you can use to display the information for the most recent rate version.

Use this procedure to maintain a rate schedule:

- Select **Menu**, and then **Rates**, and then **Rate Schedule**, and then **Search** to navigate to the Rate Schedule Query portal.
- Search for and select the appropriate rate schedule.
  - The selected rate schedule is displayed in the (new) **Rate Schedule** portal.
    - Click **Edit**, **Delete**, or **Duplicate** as appropriate.

> **Note**
>
> If you select a component-based rate schedule, the selected rate schedule is displayed in the (component-based) **Rate Schedule** page.

Base Package Calculation Rule Descriptions

This section provides detailed descriptions of each of the base package calculation rules provided with Oracle Utilities Customer Care and Billing. The descriptions of the calculation rules below contain the following information

- **Rule Name**: The name of the rule
- **Base Package Calculation Rule Business Object**: The base package business object used by the rule
- **Apply Calculation Rule Algorithm Type / Algorithm**: The base package algorithm type and algorithm specified in the "Apply Calc Rule" system event for the rule
- **Rule Parameters**: Parameters used to define the rule (if applicable). Does not include common parameters used by all rules (see Common Parameters).
- **Example**: One or more example configurations of the rule. Note that all examples use default values for all Apply Calculation Rule Algorithm Parameters.
- Common Parameters

Copyright © 2026, Oracle and/or its affiliates.

- Calculation Rule Validation
- Calculation Rules

Common Parameters

All calculation rules use the following common parameters:

- **Calculation Group**: The calculation group to which the rule belongs
- **Calculation Rule**: The user-defined name given to the calculation rule
- **Sequence**: The relative position of the calculation rule in respect of the other rules in the calculation group. This controls the order in which the system performs the calculation.
- **Description**: A short description of the rule. This is used as the Information String for the rule in the user interface.
- **Detailed Description**: A detailed description of the rule.

In addition, several of the base package calculation rules also use the following parameters:

- **Description On Bill**: The text that appears on the bill line generated for this calculation rule. Variables can be used to dynamically generate text (such as, price and quantity) when billing calculates the charges. Refer to How To Use Description on Bill for more information about available substitution variables.
- **Print** (Yes or No): A flag that specifies if the bill line created by this calculation rule should appear on the customer's bill. All bill lines are shown on Bill Segment - Calculation Details. However, only those lines marked as Print are presented to the customer.
- **Print If Zero** (Yes or No): A flag that specifies if a bill line should be printed if the bill line's value is zero.
- **Rounding Type**: This flag specifies if the calculated value should be rounded Down, Up, or Nearest. Used with Precision to control how the system rounds the rule's calculated value.
- **Precision**: This field defaults to a value consistent with the decimal positions defined on the rate schedule's currency code. Used with Rounding Type to control how the system rounds the rule's calculated value.
- **FCPO** (For Calculation Purposes Only): A flag that indicates if this calculation rule exists purely to calculate a value used by another calculation rule. When set to Yes, the system does not include the calculated amount in the bill total. The following are required when FCPO is set to Yes:
  - **Create Bill Line**: Determines if a bill line should be created for the rule (if there is no need to show the results to the customer or a user, there is no need to create a bill line).
  - **FCPO Retention Rule**: If a bill line created, this defines if the FCPO amount should be set to zero on the bill line or whether the bill line's amount should be set equal to the FCPO amount.
- **Override Proration**: A flag that specifies if an override proration algorithm should be used. This is used in situations when the standard proration logic does not satisfy your requirements. In this case, you may plug in an override proration algorithm to calculate the proration factors required by your implementation. This algorithm can be specified on the calculation rule business object or on the parent calculation rule business object if the override proration applies to all rates.

Copyright © 2026, Oracle and/or its affiliates.

- **Seasonal** (Yes or No): A flag that specifies if charges associated with the calculation rule are only levied during a specific season. If set to "Yes", the following must be specified:
  - **Season Start**: Defines the start of the season for the calculation rule.
  - **Season End**: Defines the end of the season for the calculation rule.
  - **Seasonal Prorate Method**: Defines how the system should prorate a seasonal charge when a bill segment's period is not entirely within the seasonal period. The following options are available:
    * **Accounting Date**, **Consumption End Date** and **Consumption Start Date** options will not prorate the charge calculated by the calculation rule. Rather, it will only calculate the charge if the accounting date associated with the bill segment is within the seasonal period, or if the consumption period's end date is within the seasonal period, or if the consumption period's start date is within the seasonal period respectively.
    * **Prorate** will prorate the calculation rule charge based on number of days in the season that is included in the bill period.
    * The **Prorate Seasonal SQ** option is only pertinent if you have:
      * Seasonal service quantity (SQ) charges, AND
      * Meters that measure seasonal consumption on separate registers, AND
      * You have a bill that crosses seasonal boundaries, AND
      * You have multiple rate version calculation groups effective across the bill period.
        If the Prorate Seasonal SQ option is used, the system will prorate the charge calculated by a SQ calculation rule based on the number of days in the season that is included in the rate version calculation group's period.
- **GL Distribution**: If the charge calculated by the calculation rule impacts the general ledger, you must define the distribution code used to construct the GL account.
  - **Distribution Code Source**: Specifies the type of GL distribution to use for this calculation rule.
    If the calculation rule's charges should be booked to a single GL account regardless of the service agreement's SA Type's revenue class, select the **Specific Distribution Code** source and enter a **Distribution Code**.

    If the calculation rule's charges are booked to a different GL account depending on the service agreement's revenue class, select the **Revenue Class** source and insert the appropriate **Revenue Class** and **Distribution Code** for each that uses the rate.
- **Cross Reference**: Defines the calculation line categories that the resultant calculation details belong to. For example, for a rate that calculates city and state taxes on consumption and demand based charges, the following categories could be used to calculate these taxes:
  - A calculation line category type called "Taxable" with a category value for City Tax and another for State Tax.
  - Each of the rate's calculation rules used for consumption and demand based charges should define the Member Category(s) that it belongs to ("Taxable - City Tax" and "Taxable - State Tax"). In other words, these rules would be defined as a member of the city tax and state tax categories.
  - The Apply To rules used to calculate city tax and state tax should define the respective Target Category(s) ("Taxable - City Tax" and "Taxable - State Tax") that they are applied to.
    There are two types of cross-reference categories:

Copyright © 2026, Oracle and/or its affiliates.

  - **Target Category**: Specifies the category that the calculation rule targets. Used with Apply To, Maximum Charge, and Minimum Charge calculation rules.
  - **Member Category**: Specifies the category to which the calculation rule belongs.
- **Characteristics**: Defines additional calculation rule attributes that may be used in rate processing or for reporting purposes. A calculation rule's characteristics are copied onto the resultant bill lines for those characteristic types that are allowed on bill segment calc lines. You can indicate that “C1-ADDSC” characteristic ad hoc value of a bill line should be substituted into the bill line's description by using the %V substitution variable in the calculation rule's **Description On Bill** field.

Calculation Rule Validation

All calculation rules are validated when created to ensure they contain proper data needed to perform their calculations. The algorithms used to perform these validations are performed by the algorithms specified in the “Validation” system event on the calculation rule’s business object.

Calculation Rules

This section provides descriptions for the base package calculation rules.

- Apply To
- Calculate Final Reading Values
- Calculate Service Quantity
- Copy Service Quantity from Master Bill Segment
- Execute Calc Group
- Flat Charge
- Item Type
- Math
- Maximum Charge
- Minimum Charge
- Service Quantity
- Simple Summary
- Stepped Service Quantity
- Summary

Apply To

Apply To calculation rules create bill lines that levy charges based on amounts calculated on other bill lines. Apply To calculation rules must define the respective target category that they are applied to. The calculation rule(s) that compute charges to which "Apply To" rules will be applied must be defined as members of the target category. See **Cross-Reference** under Common Parameters for more information about Member and Target categories.

- **Rule Name**: Apply To
- **Base Package Calculation Rule Business Object**: C1-ApplyTo

Copyright © 2026, Oracle and/or its affiliates.

- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-APPLYTO
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    * **Error If No Value**: Indicates what should happen if the system cannot find any calculation rules that are members of the target category at billing time.
    * **Seasonal**: See Common Parameters.
    * **Value Type**: Defines if the price is a Charge, Percentage or Unit Rate.
    * **Value Source**: specifies the source of the price. Depending on the value source selected, either a Bill Factor, Value, or Value Algorithm is required.
- **Example**: The following calculation rule applies a county tax other calculation rules that belong the "Taxable - County Tax" category.
  Calculation Group: Taxes

  Calculation Rule: CNTYTAX

  Sequence: 530

  Description: County Tax

  Description On Bill: County sales tax %R %T

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

  FCPO: No

  Override Proration: No

  Calculation Details:
  - **Error If No Value**: No
  - **Seasonal**: No
  - **Value Type**: Percentage
  - **Value Source**: Bill Factor
  - **Bill Factor**: County Tax
    GL Distribution:
  - **Distribution Source Code**: Specific Distribution Code
  - **Distribution Code**: Payable - County Tax
    Cross Reference:
  - **Target Category (Category Type / Value)**: Taxable / County Tax
  - **Member Category (Category Type/Value)**: Summary Item / Summary Line

Calculate Final Reading Values

Calculate Final Reading Values calculation rules determine the final values for readings based on user-defined UOM/TOU/SQI.

- **Rule Name**: Calculate Final Reading Values

Copyright © 2026, Oracle and/or its affiliates.

- **Base Package Calculation Rule Business Object**: C1-CalculateFinalValue
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-CALCFNVL
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Unit of Measure**: The unit of measure for the measured quantity.
    - **Time of Use**: The time of use for the measured quantity.
    - **Service Quantity Identifier**: The service quantity identifier for the measured quantity.
    - **Retain Measured SQ**: indicates whether to save the accumulated measured quantity entry in the SQ collection. If this value is set to No, it means the value will be removed from the SQ collection. Before any pre-processing rule is executed, the rate engine will place the sum / max value of the read details into the SQ collection. For example, if the read details contain entries for measured CCF, there will be an entry in the SQ collection for the total CCF from all read details. This might be what is needed, but it may be extraneous if a rule converts the measured quantity into therms. In this situation, the SQ collection only needs therms and the CCF entry can be removed.
    - **Formula**: Defines the formula used to calculate the final read quantity. Bill Factors (or scalar variables) to be used in the calculation are designated as Vn, where n matches the number assigned to the variable (i.e. V1 = Bill Factor 1, V2 = Bill Factor 2, etc.). Measured Quantity is referenced as MQ in the formula Mathematical operations supported include addition, subtraction, multiplication and division. For example: (MQ * V1) would be the formula to convert CCF to THERM.
  - **Scalar Variables**: Defines the scalar variables or Bill Factors to be used in the calculation. When referenced in the formula, scalar variables are designated as Vn, where n matches the number assigned to the variable (i.e. V1 = Bill Factor 1, V2 = Bill Factor 2, etc.). Note that only regular bill factors may be used in the formula. Interval and Time of Use bill factors are not supported.
    - **Sequence**: Designates the number of the variable (V1, V2, etc.)
    - **Bill Factor**: The Bill Factor used for the variable.
  - **Result**: Defines the type of data (**Unit of Measure**, **Time of Use**, and/or **Service Quantity Identifier**) to be used for the final quantity calculated by the rule.
- **Processing**: The following processing takes place:
  - The rule will process each read detail whose initial UOM matches the measured quantity UOM.
  - It substitutes the reference to MQ in the formula with each read's measured quantity and then calls the formula service passing the defined bill factors as variables 1 to n. The read's end date is passed as the effective date.
  - It then populates each read's final value with the results of the formula and it populates the final UOM / TOU / SQI with the values defined on the rule.
  - It populates the SQ collection with the results (this will either add a new value if the Final UOM / TOU / SQI is not present in the SQ collection, OR it will update the existing SQ quantity with the sum or max of the calculated value depending on the Final UOM's measure's peak value). Note, the initial and final values of such an SQ will be the calculated value.

Copyright © 2026, Oracle and/or its affiliates.

- It then removes the measured UOM / TOU / SQI from the SQ collection if the rule has been configured as such.

- **Example**: This calculation rule converts CCF (as recorded on the meter) to therms for billing purposes.
  Calculation Group: CCF to Therm Conversion (Calculate SQ - SQ and BF)

  Calculation Rule: CCF2TH

  Sequence: 10

  Description: Convert CCF to TH

  Calculation Details:

  - **Unit of Measure**: 100 cubic feet of gas
  - **Retain Measured SQ**: Yes
  - **Formula**: MQ*V1*V2
    Scalar Variables (Sequence / Bill Factor):
  - 1 / Pressure adjustment zone
  - 2 /Take point for therm conv.fac
    Result:
  - **Unit of Measure**: Thermal units

Calculate Service Quantity

Calculate Service Quantity calculation rules derive or calculate a service quantity that is eventually inserted into the bill segment SQ collection. Calculate Service Quantity calculation rules are generally used to either manipulate measured consumption, or to derive quantities that are not measured by a meter.

- **Rule Name**: Calculate Service Quantity
- **Base Package Calculation Rule Business Object**: C1-CalculateServiceQuantity
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-CALC-SQ
- **Rule Parameters**:
  - **Scalar Variable 1 (Scalar Variable 2, Scalar Variable 3, ... Scalar Variable 5)**: Defines one of the scalar variables to be used in the calculation. When used in formulas, scalar variables are designated as Vn, where n matches the number assigned to the variable (i.e. V1 = Scalar Variable 1, V2 = Scalar Variable 2, etc.).
    - **Type**: Defines the type of variable and can be one of the following:
      - **Algorithm**: an algorithm is responsible for retrieving the scalar variable value, e.g. calculate number of days in the bill period.
      - **Bill Factor**: the value for a specified bill factor that is in effect for the period.
      - **Calculated Line**: the value is derived by summing previously calculated lines defined as members of a target category.
      - **Characteristic Value**: the value is derived by summing characteristic values of a given characteristic type.
      - **Contract Quantity**: the value for a given contract quantity type specified on a service agreement.
      - **Historical Service Quantity**: the value is derived based on the SA's historical bill segment service quantities for the specified UOM / TOU / SQI. The

Copyright © 2026, Oracle and/or its affiliates.

Aggregate Function defines the calculation to perform on the set of historical service quantities retrieved to derive a single value. Average, Maximum, Minimum and Sum are supported.

        - **Service Quantity**: service quantities from the bill period's SQ collection, based on a specified UOM / TOU / SQI. This type of variable allows this rule to make use of values calculated by other rules.
        - **Previous Quantity**: the value is derived from the SA's most recent frozen bill segment service quantities for the specified UOM / TOU / SQI.
    - **Missing Value Action**: Defines what should take place if the system is unable to retrieve or derive a value for a particular scalar variable. Valid options are as follows:
        - Select **Error** to create an exception
        - Select **Skip** to exit the rule and process the next rule
        - Select **Use Default Value** if the scalar variable should be set to the default value specified
- **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Calculation Rule Processing**: Indicates if the rule is **Always** executed by rate application, even if the consumption details on the original bill segment are to be used when the system recalculates a bill segment. If the rule should not be executed when the consumption details on the original bill segment are used, indicate that the calculation rule should only be executed on the **Initial SQ Calculation**.
    - **Fail Action**: Indicates the action to take if an error is encountered when applying the rule's formula. Available options are to either create a bill segment **Error**, or to **Skip** the calculation rule.
    - **Calculation Rule Output**: Specifies whether calculation rule should add or update a **Service Quantity** entry in the bill period's SQ collection, or create a bill **Calculation Line**, or **Both**.
    - **Retain SQ**: Indicates whether to save the calculated value on the bill segment. If this value is set to **No**, it means the value will only exist in the SQ collection while the rate engine is calculating. A value of No should be used for SQ entries that do not have to be retained for audit or analytic purposes.
    - **Formula Source**: Specifies the type of formula to use for the rule. Valid options include "Simple Formula" and "Conditional Formula".
    - **Formula**: Specify the **Simple Formula** to apply. Scalar variables to be used in the calculation are designated as Vn, where n matches the number assigned to the variable (i.e. V1 = Scalar Variable 1, V2 = Scalar Variable 2, etc.). Mathematical operations supported include addition, subtraction, multiplication and division. For example: (V1*V2)/2.
        - **Conditional Formula**: this allows comparison between one or more pairs of operands to determine the specific formula to execute. A conditional formula can utilize one or more conditions. Each condition comprises the following:
            - **Operand 1**: the first operand in the condition. Can reference an expression with one or more scalar variables (designated as Vn), e.g. [V1+V2] or [(V1/V2)*V3].
            - **Operator**: the operator used to compare Operand 1 with Operand 2.

Copyright © 2026, Oracle and/or its affiliates.

        - **Operand 2**: the second operand in the condition. Can reference an expression with one or more scalar variables, or a constant (similar to Operand 1).
        - **True Action**: indicates how to proceed if the comparison between the operands is true. Can be one of the following:
          Apply True Formula: indicates that the True Formula be executed.

          Check Next Condition: indicates that the next condition should be checked.
        - **True Formula**: the formula to apply if True Action is set to Apply True Formula. Can reference an expression with one or more scalar variables (designated as Vn), or a constant.
        - **False Action**: indicates how to proceed if the comparison between the operands is false. Can be one of the following:
          Apply False Formula: indicates that the False Formula be executed.
        - **Check Next Condition**: indicates that the next condition should be checked.
        - **False Formula**: the formula to apply if the False Action is set to Apply False Formula. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).
    - **Result**: Define the **Unit of Measure**, **Time of Use** and/or **Service Quantity Identifier** to be used if this calculation rule inserts or updates a service quantity in the bill period's SQ collection.

- **Example**: This calculation rule calculates total kWh from On Peak and Off Peak service quantities.
  Calculation Group: Calculate KWH

  Calculation Rule: TOTKWH

  Sequence: 10

  Description: Total amount of KWH (On Peak + Off Peak)

  Scalar Variable 1:

  - **Type**: Service Quantity
  - **Missing Value Action**: Error
  - **Unit of Measure**: Kilowatt hour
  - **Time of Use**: On Peak
  - **SQ To Use**: Billable
    Scalar Variable 2:
  - **Type**: Service Quantity
  - **Missing Value Action**: Error
  - **Unit of Measure**: Kilowatt hour
  - **Time of Use**: Off Peak
  - **SQ To Use**: Billable
    Calculation Details:
  - **Calculation Rule Processing**: Always
  - **Fail Action**: Error

Copyright © 2026, Oracle and/or its affiliates.

- **Calculation Rule Output**: Service Quantity
- **Retain SQ**: No
- **Formula Source**: Simple Formula
- **Formula**: V1 + V2
  Result:
- **Unit of Measure**: Kilowatt hour

Copy Service Quantity from Master Bill Segment

Calculate Final Reading Values calculation rules determine the final values for readings based on user-defined UOM/TOU/SQI.

- **Rule Name**: Copy Service Quantity From Master Bill Segment
- **Base Package Calculation Rule Business Object**: C1-CopySQFromMasterBillSegment
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-CPY-MB-SQ
- **Rule Parameters**:
  - **Characteristics**: Defines characteristic(s) and corresponding value(s) that are to be passed to the referenced calculation group.
    - **Characteristic Type:** Defines the characteristic type that is to be associated to the calculation rule.

      > **Note**
      >
      > Characteristic Types for rates must be configured to be used on a calculation rule and/or in the characteristic collection. Refer to Defining Characteristic Types for information on how to set up characteristic types.

    - **Characteristic Value**: Defines the parameter value that will be associated to the characteristic type.

Execute Calc Group

Execute Calc Group calculation rules call another calculation group, and pass characteristics and their values to the referenced calculation group.

- **Rule Name**: Execute Calculation Group
- **Base Package Calculation Rule Business Object**: C1-CalcRuleReferredCalcGroup
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-ACALR-RFG
- **Rule Parameters**:
  - **Referenced Calculation Group**: Defines the calculation group that's being called.
    - **Calculation Group**: Defines the calculation group that's being called.
  - **Characteristics**: Defines characteristic(s) and corresponding value(s) that are to be passed to the referenced calculation group.
    - **Characteristic Type:** Defines the characteristic type that is to be passed to the referenced calculation group.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Characteristic Types for rates must be configured to be used on a calculation rule and/or in the characteristic collection. Refer to Defining Characteristic Types for information on how to set up characteristic types.

Characteristic Value: Defines the parameter value that will be associated to the characteristic type.

Flat Charge

Flat Charge calculation rules create bill lines that levy service charges and fees that aren't based on how much the customer uses. For example a flat charge calculation rule may be used to apply a monthly service charge of $10 to a customer's bill.

- **Rule Name**: Flat Charge
- **Base Package Calculation Rule Business Object**: C1-FlatCharge
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-FLAT-CHRG
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Seasonal**: See Common Parameters.
    - **Value Type**: Defines the type of value to use for the rule. For Flat Charge rules, this should be set to **Charge**.
    - **Value Source**: Specifies the source of the price used to calculate the charge. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.
- **Example**: This calculation rule levies a customer charge.
  Calculation Group: Customer Charge

  Calculation Rule: CSTCHG

  Sequence: 20

  Description: Customer Charge

  Description On Bill: Customer Charge

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

  FCPO: No

  Override Proration: No

  Calculation Details:
  - **Seasonal**: No
  - **Value Type**: Charge
  - **Value Source**: Value

Copyright © 2026, Oracle and/or its affiliates.

- **Value**: 5
  GL Distribution:
- **Distribution Source Code**: Specific Distribution Code
- **Distribution Code**: Revenue - Energy
  Cross Reference:
  - **Member Category (Category Type / Value)**: Subject to tax / City taxable

Item Type

Item Type calculation rules create bill lines that levy charges based on the number and type of items used by a customer. The system keeps track of items on the service point(s) linked to the customer's service agreement.

- **Rule Name**: Item Type
- **Base Package Calculation Rule Business Object**: C1-ItemType
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-ITEMTYPE
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Seasonal**: See Common Parameters.
    - **Item Type**: Defines the types of items to which charges are applied. When defining more than one item type on a single calculation rule, you can indicate that the item type of a bill line should be substituted into the bill line's description by using the %M substitution variable in the calculation rule's **Description On Bill** field.
      - **Error If No Value**: A flag that indicates (Yes or No) if a bill error should be generated if the item type specified on the calculation rule was not supplied at billing time.
      - **Value Type**: Defines the type of charge. Valid options include Charge, Percentage, or Unit Rate.
      - **Value Source**: Defines the source of the price used to calculate the charge. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.
      - **Value/Bill Factor/Algorithm**: The specific Value, Bill Factor, or Algorithm that serves as the source of the price used to calculate the charge.
- **Example**: This calculation rule calculates charges for light poles.
  Calculation Group: Lamp Charges

  Calculation Rule: POLE CHARGE

  Sequence: 10

  Description: Pole Charge

  Description On Bill: %C lamp poles at %R per pole %D %I

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

Copyright © 2026, Oracle and/or its affiliates.

FCPO: No

Override Proration: No

Calculation Details:

- **Seasonal**: No
- **Items**:
  - **Item Type**: Pole(s)
  - **Error If No Value**: No
  - **Value Type**: Unit Rate
  - **Value Source**: Value
  - **Value / Bill Factor /Algorithm**: 3
    GL Distribution:
- **Distribution Source Code**: Specific Distribution Code
- **Distribution Code**: Revenue - Area Lighting
  Cross Reference:
- **Member Category (Category Type / Value)**: Charges / Total Charges
- **Member Category (Category Type / Value)**: Charges / Lamp Charges

Math

Math calculation rules derive interval data based on a formula, and apply TOU mappings and/or other operations to the derived data to calculate service quantities.

- **Rule Name**: Math
- **Base Package Calculation Rule Business Object**: C1-Math
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-MATH
- **Rule Parameters**:
  - **Vector 1 (Vector 2, Vector 3, ..., Vector 5)**: Defines one of the vectors (Vector 1, 2, 3, ..., 5) to be used in the calculation. When used in formulas, interval values for vectors are designated as IV1, IV2, IV3, ..., IV5.
    - **Vector Type**: Defines the type of vector. Valid options include **Interval Bill Factor** and **Interval Service Quantity**.
    - **Bill Factor**: Defines the Interval Bill Factor from which interval values are derived for the vector.
    - **Unit of Measure/Time of Use/Service Quantity Identifier**: The UOM/TOU/SQI used to retrieve interval values from either the usage request service quantity entries given a usage request ID and usage period, or from the rate engine buffer populated prior to calling rate application.
    - **Missing Interval Data Action**: Defines the action to take (**Error** or **Skip**) if interval data is missing for the expected period
  - **Scalars**: Defines scalar variables to be used in the calculation. When used in formulas, scalar variables are designated as Vn, where n is the number of the variable (based on the sequence in which they are defined in the list).
    - **Scalar Type**: Defines the type of variable. Can be one of the following:

Copyright © 2026, Oracle and/or its affiliates.

        - **Bill Factor**: The value for a specified bill factor that is in effect for the billing period.
        - **Set Function**: The result of applying a function to a set of interval measurement values (defined as a vector).
        - **Service Quantity**: Service quantities from the current SQ list, based on a specified UOM, TOU, or SQI. This type of variable allows this rule to make use of values calculated by other rules.
    - **Bill Factor**: Defines the bill factor used to retrieve the variable value (applicable only if Scalar Type is Bill Factor)
    - **Unit of Measure**: Defines the UOM for service quantities to retrieve from the SQ list (applicable only if Scalar Type is Service Quantity).
    - **Time of Use**: Defines the TOU for service quantities to retrieve from the SQ list (applicable only if Scalar Type is Service Quantity).
    - **Service Quantity Identifier**: Defines the SQI for service quantities to retrieve from the SQ list (applicable only if Scalar Type is Service Quantity).
    - **SQ to Use**: Every service quantity supplied to a rate has two quantities; the quantity initially measured, and the quantity that should be priced by the rate. Most of the time these two amounts will be the same, unless another rule has executed and caused the billable quantity to change. Each rule needs to know if it should use the **Initial** or **Billable** quantity.
        - **Set Function**: Defines the function used to calculate a scalar value from a set of interval values (defined as a vector). This is applicable only if Scalar Type is Set Function). Can be one of the following:
            - **Average**: calculates the average of the vector's interval measurement values.
            - **Count**: returns the number of interval measurements.
            - **Max**: returns the maximum value from the vector's interval measurement values.
            - **Min**: returns the minimum value from the vector's interval measurement values.
            - **Total**: calculates the total of the vector's interval measurement values.
        - **Vector**: the vector to be used for this variable (applicable only if Scalar Type is Set Function). Can be one of the following:
            - **FV** (Final Vector Interval Value): the vector containing the results of the formula defined in the Calculation Details section.
            - **IV1** (Vector 1 Interval Value): the vector defined as Vector 1.
            - **IV2** (Vector 2 Interval Value): the vector defined as Vector 2.
            - **IV3** (Vector 3 Interval Value): the vector defined as Vector 3.
            - **IV4** (Vector 4 Interval Value): the vector defined as Vector 4.
            - **IV5** (Vector 5 Interval Value): the vector defined as Vector 5.
    - **Missing Value Action**: Defines the action to take (**Error**, **Skip**, or **Use Default Value**) if data is missing for the scalar variable for the expected period
    - **Default Value**: The default value to use if data is missing for the scalar variable. Applicable only if Missing Value Action is Use Default Value.
    - **Calculation Details**: Details regarding how the calculation rule performs its calculations:

Copyright © 2026, Oracle and/or its affiliates.

* **Calculation Rule Processing**: Indicates if the rule is **Always** executed by rate application, even if the consumption details on the original bill segment are to be used when the system recalculates a bill segment. If the rule should not be executed when the consumption details on the original bill segment are used, indicate that the calculation rule should only be executed on the **Initial SQ Calculation**.
* **Fail Action**: Indicates the action to take if an error is encountered when applying the rule's formula. Available options are to either create a bill segment **Error**, or to **Skip** the calculation rule.
* **Calculation Rule Output**: Specifies whether calculation rule should add or update a **Service Quantity** entry in the bill period's SQ collection, or create a bill **Calculation Line**, or **Both**.
* **Retain SQ**: Indicates whether to save the calculated value on the bill segment. If this value is set to **No**, it means the value will only exist in the SQ collection while the rate engine is calculating. A value of No should be used for SQ entries that do not have to be retained for audit or analytic purposes.
* **Formula Source**: Specifies the type of formula to use for the rule. Valid options include "Simple Formula" and "Conditional Formula".
* **Formula**: Specify the **Simple Formula** to apply. Scalar variables to be used in the calculation are designated as Vn, where n matches the number assigned to the variable (i.e. V1 = Scalar Variable 1, V2 = Scalar Variable 2, etc.). Mathematical operations supported include addition, subtraction, multiplication and division. For example: (V1*V2)/2.
    * **Conditional Formula**: this allows comparison between one or more pairs of operands to determine the specific formula to execute. A conditional formula can utilize one or more conditions. Each condition comprises the following:
        * **Operand 1**: the first operand in the condition. Can reference an expression with one or more scalar variables (designated as Vn), e.g. [V1+V2] or [(V1/V2)*V3].
        * **Operator**: the operator used to compare Operand 1 with Operand 2.
        * **Operand 2**: the second operand in the condition. Can reference an expression with one or more scalar variables, or a constant (similar to Operand 1).
        * **True Action**: indicates how to proceed if the comparison between the operands is true. Can be one of the following:
          Apply True Formula: indicates that the True Formula be executed.

          Check Next Condition: indicates that the next condition should be checked.
        * **True Formula**: the formula to apply if True Action is set to Apply True Formula. Can reference an expression with one or more scalar variables (designated as Vn), or a constant.
        * **False Action**: indicates how to proceed if the comparison between the operands is false. Can be one of the following:
          Apply False Formula: indicates that the False Formula be executed.
        * **Check Next Condition**: indicates that the next condition should be checked.
        * **False Formula**: the formula to apply if the False Action is set to Apply False Formula. Can reference a vector (designated as IVn, where n is the

Copyright © 2026, Oracle and/or its affiliates.

number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).

- **Result**: Defines how service quantities are inserted into the SQ list. Math calculation rules can insert one or more entries into the SQ list.
  - **Result Source**: Defines the method to use to calculate for the service quantity. Can be one of the following:
    - **Set Function Against Derived Vector**: Applies a function to the derived interval values. The function to be applied is specified in the Set Function field.
    - **Scalar Formula Result**: Applies a user-defined formula. The formula is specified in the Scalar Formula field.
  - **Set Function**: the function to apply to the derived interval values (applicable only if Result Source is set to Set Function Against Derived Vector). Can be one of the following:
    - **Average**: Calculates the average of the derived interval measurement values.
    - **Count**: Returns the number of derived interval measurements.
    - **Max**: Returns the maximum value from the derived interval measurement values.
    - **Min**: Returns the minimum value from the derived interval measurement values.
    - **Total**: Calculates the total of the derived interval measurement values.
  - **Scalar Formula**: the formula to apply (applicable only if Result Source is set to Scalar Formula). Variables used in this formula must be defined in the Scalars section. When referenced in formulas, scalar variables are designated as Vn (where n is the number of the variable).
  - **Unit of Measure**: the UOM to be used when inserting service quantity entries
  - **Time of Use**: the TOU to be used when inserting service quantity entries
  - **Service Quantity Identifier**: the SQI to be used when inserting service quantity entries
  - **Apply TOU Map**: Indicates if a TOU map should be applied to the derived interval values. If TOU periods and values are returned as a result of TOU mapping, then service quantity entries are inserted.
  - **TOU Map**: Defines the TOU Map to apply to the derived interval values (applicable only if Apply TOU Map is set to Yes).
  - **TOU Mapping Function**: Defines the function to apply to the derived interval values when calculating for the time of use values (applicable only of Apply TOU Map is set to Yes). Can be one of the following:
    - **Max**: Returns the maximum value from the derived interval values for each TOU period.
    - **Sum**: Returns the sum of the derived interval values for each TOU period.

- **Processing**:
  Math rules derive interval values based on a formula, and apply TOU mappings and/or other operations to the derived data to calculate billing quantities. Examples include the following:
  - Derive an interval data curve (vector) given a formula. For example, derive a power factor curve given a formula using kWh and kVarh curves.

Copyright © 2026, Oracle and/or its affiliates.

- Apply TOU mapping to a derived interval data curve. For example, after deriving the power factor curve, perform TOU mapping on the result.
- Perform mathematical operations on SQ entries. For example, get total kWh consumption by adding "kWh On Peak", "kWh Off Peak", and "kWh Shoulder Peak" where "kWh On Peak", "kWh Off Peak", and "kWh Shoulder Peak" were calculated by a previous calculation rule.
  Each interval data curve is defined as a vector parameter (the rule can define up to 5 vectors). Mathematical operations defined by the "Vector Processing" parameters can be performed between vectors (e.g. IV1 * IV2) and between vectors and scalar variables (e.g. IV1 * V1).

- **Example**: This calculation rule calculates prices for interval usage by multiplying an interval service quantity times an interval price bill factor.
  Calculation Group: Electric Commercial - Interval Pricing

  Calculation Rule: INTDPRICING

  Sequence: 10

  Description: KWH Interval Prices

  Description On Bill: kWh Interval Prices

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

  FCPO: No

  Vector 1:

  - **Vector Type**: Interval Service Quantity
  - **Unit of Measure**: Kilowatt hour
  - **Missing Interval Data Action**: Error
    Vector 2:
  - **Vector Type**: Interval Bill Factor
  - **Bill Factor**: Interval Prices
  - **Missing Interval Data Action**: Error
    Calculation Details:
  - **Calculation Rule Processing**: Always
  - **Fail Action**: Error
  - **Calculation Rule Output**: Calculation Line
  - **Formula Source**: Simple Formula
  - **Simple Formula**: IV1 * IV2
    Result:
  - **Result Source**: Set Function Against Derived Vector
  - **Set Function**: Total
    GL Distribution:
  - **Distribution Source Code**: Specific Distribution Code
  - **Distribution Code**: Revenue - Energy

Copyright © 2026, Oracle and/or its affiliates.

Maximum Charge

Maximum Charge calculation rules are used to create bill lines that levy charges only when the sum of previously calculated lines is more than the maximum charge amount. Maximum charge calculation rules must define the respective target category that they are applied to. The calculation rule(s) that compute charges that the maximum is levied on must be defined as members of the target category. See **Cross-Reference** under Common Parameters for more information about Member and Target categories.

Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a maximum discount of $ -2.00 and previous calculation rules have calculated a discount of $ -3.00. You want a calculation rule to create a bill line for $1.00 to reduce the discount. A maximum charge calculation rule will not work in this case because -3 is considered less than -2. For this business scenario you should use a Minimum Charge calculation rule.

Note: Maximum Charge calculation rules are rare. You tend to only see them in deregulated markets where customers have agreed to pay the market rate but want a cap on the maximum they'll pay each bill period.

- **Rule Name**: Maximum Charge
- **Base Package Calculation Rule Business Object**: C1-Maximum
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-MAXIMUM
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Error If No Value**: Indicates what should happen if the system cannot find any calculation rules that are members of the target category at billing time.
    - **Seasonal**: See Common Parameters.
    - **Value Type**: Defines the type of value to use for the maximum value for the rule. For Maximum Charge rules, this should be set to **Charge**.
    - **Value Source**: Specifies the source of the price used to calculate the charge. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.
- **Example**: This calculation rule sets a maximum charge for area lighting.
  Calculation Group: Customer-Owned Street & Highway Lighting

  Calculation Rule: MAXCHRGE

  Sequence: 20

  Description: Maximum Charge - Area Lighting

  Description On Bill: Maximum Charge

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

  FCPO: No

  Override Proration: No

Copyright © 2026, Oracle and/or its affiliates.

Calculation Details:

- **Error If No Value**: No
- **Seasonal**: No
- **Value Type**: Charge
- **Value Source**: Value
- **Value**: 100
  GL Distribution:
- **Distribution Source Code**: Specific Distribution Code
- **Distribution Code**: Revenue - Area Lighting
  Cross Reference:
- **Target Category (Category Type / Value)**: Revenue / Maximum Consumption Charge

Minimum Charge

Minimum Charge calculation rules create bill lines that levy charges only when the sum of previously calculated lines is less than the minimum charge amount. Minimum charge calculation rules must define the respective target category that they are applied to. The calculation rule(s) that compute charges that the minimum is levied on must be defined as members of the target category. See **Cross-Reference** under Common Parameters for more information about Member and Target categories.

Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a minimum discount of $ -2.00 and previous calculation rules have calculated a discount of $ -1.00. You want a calculation rule to create a bill line for $ -1.00 to apply a further discount. A minimum charge calculation rule will not work in this case because -1 is considered more than -2. For this business scenario you should use a maximum charge calculation rule.

- **Rule Name**: Minimum Charge
- **Base Package Calculation Rule Business Object**: C1-Minimum
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-MINIMUM
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Error If No Value**: Indicates what should happen if the system cannot find any calculation rules that are members of the target category at billing time.
    - **Seasonal**: See Common Parameters.
    - **Value Type**: Defines the type of value to use for the minimum value for the rule. For Minimum Charge rules, this should be set to **Charge**.
    - **Value Source**: Specifies the source of the price used to calculate the charge. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.
- **Example**: This calculation rule defines a minimum charge for use with residential electric rates.
  Calculation Group: Electric Residential

  Calculation Rule: MINCHRG

  Sequence: 50

Copyright © 2026, Oracle and/or its affiliates.

Description: Minimum Charge

Description On Bill: Minimum Charge

Print: Yes

Print if Zero: No

Rounding Type: Nearest

Precision: 0.01

FCPO: No

Override Proration: No

Calculation Details:

- **Error If No Value**: No
- **Seasonal**: No
- **Value Type**: Charge
- **Value Source**: Value
- **Value**: 75
  GL Distribution:
- **Distribution Source Code**: Specific Distribution Code
- **Distribution Code**: Revenue - Miscellaneous
  Cross Reference:
- **Target Category (Category Type / Value)**: Charges / Energy Charges
- **Member Category (Category Type / Value)**: Summary Item / Summary Line
- **Member Category (Category Type / Value)**: Taxable - State Tax
- **Member Category (Category Type / Value)**: Taxable - City Tab

Service Quantity

Service Quantity calculation rules create bill lines that levy charges based on some type of consumption.

Frequently the amount consumed is measured by a meter's registers and sometimes it's calculated using Pre-Processing calculation and sometimes it's derived based on the type of item (such as, lamps). Regardless of how the amount consumed is calculated, the charge for the consumption amount is levied using this type of calculation rule.

- **Rule Name**: Service Quantity
- **Base Package Calculation Rule Business Object**: C1-ServiceQuantity
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-SQ
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    * **Unit of Measure**: The UOM used as the service quantity that the price is applied to.
    * **Time of Use**: The TOU used as the service quantity that the price is applied to.

Copyright © 2026, Oracle and/or its affiliates.

    - **Service Quantity Identifier**: The SQI used as the service quantity that the price is applied to.
    - **Error If No Value**: A flag that indicates (Yes or No) if a bill error should be generated if the UOM/TOU/SQI specified on the rule was not supplied at billing time.
    - **Measures Peak**: Indicates (Yes or No) if the UOM/TOU/SQI specified for the rule is used to measure peak (demand) values.
    - **GL Statistical Qty**: A flag that specifies (Yes or No) if GL journal lines generated for this calculation rule should also contain the service quantity amount as a statistical quantity. You would use this option if you keep track of both dollar amounts and consumption units in your general ledger.
    - **Seasonal**: See Common Parameters.
    - **Value Type**: Defines the type of price to be used in the rule's calculations. Valid options include Charge, Percentage, or Unit Rate.
    - **Value Source**: Defines the source of the price used in the rule's calculations. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.

- **Example**: This calculation rule calculates charges based a customer's KWH consumption.
  Calculation Group: Electric Residential Charges

  Calculation Rule: KWHCHRG

  Sequence: 10

  Description: kWh Unit Rate

  Description On Bill: Usage %Q at %R per kWh

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01000

  FCPO: No

  Override Proration: No

  Calculation Details:

  - **Unit of Measure**: Kilowatt hour
  - **Error If No Value**: Yes
  - **Measures Peak**: No
  - **GL Statistical Qty**: Yes
  - **Seasonal**: No
  - **Value Type**: Unit Rate
  - **Value Source**: Value
  - **Value**: 0.0482000
    GL Distribution:
  - **Distribution Source Code**: Specific Distribution Code
  - **Distribution Code**: Revenue - Electric - Residential
    Cross Reference:
  - **Target Category**:

Copyright © 2026, Oracle and/or its affiliates.

        - **Category Type**:
        - **Category Value**:
    - **Member Category (Category Type / Value)**: Revenue / Consumption Charge

Simple Summary

Simple Summary calculation rules create a "subtotal" on the bill by summarizing bill lines produced by other calculation rules in the same calculation group.

- **Rule Name**: Simple Summary
- **Base Package Calculation Rule Business Object**: C1-SimpleSummary
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-SIMP-SUMM
- **Rule Parameters**:
    - **Referenced Calculation Rules**: Defines the calculation rules that should be included in the summary. Note that any calculation rules listed here must belong to the same calculation group as the Summary rule.
- **Example**: This calculation rule creates a summary of TOU-based charges (based on rules within the same calculation group as this rule).
  Calculation Group: Commercial TOU Electric Charges

  Calculation Rule: KWHTOT

  Sequence: 40

  Description: Consumption Subtotal

  Description On Bill: Consumption Subtotal

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01

  FCPO: No

  Referenced Calculation Rules:
    - Off Peak - kWh
    - On Peak - kWh
    - Shoulder Peak - kWh
      Cross Reference:
    - **Member Category (Category Type / Value)**: Taxable / State Tax

Stepped Service Quantity

Stepped Service Quantity calculation rules create bill lines that levy charges based on some type of consumption when different prices are applied to different ranges of consumption (for example, the first 50 kWh are $0.43 per kWh, anything in excess of 50 kWh is charged at $0.71 per kWh)

- **Rule Name**: Stepped Service Quantity
- **Base Package Calculation Rule Business Object**: C1-SteppedSQ

Copyright © 2026, Oracle and/or its affiliates.

- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-STPD-SQ
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Unit of Measure**: The UOM used as the service quantity that the price is applied to.
    - **Time of Use**: The TOU used as the service quantity that the price is applied to.
    - **Service Quantity Identifier**: The SQI used as the service quantity that the price is applied to.
    - **Error If No Value**: A flag that indicates (Yes or No) if a bill error should be generated if the UOM/TOU/SQI specified on the rule was not supplied at billing time.
    - **Measures Peak**: Indicates (Yes or No) if the UOM/TOU/SQI specified for the rule is used to measure peak (demand) values.
    - **GL Statistical Qty**: A flag that specifies (Yes or No) if GL journal lines generated for this calculation rule should also contain the service quantity amount as a statistical quantity. You would use this option if you keep track of both dollar amounts and consumption units in your general ledger.
    - **Create Step Summary**: Stepped service quantity calculation rules may be configured with either a single step or multiple steps. The Create Step Summary flag is used to produce a summary bill segment calculation line if the calculation rule has multiple steps. Valid options include:
      - **Summarize Step Lines, Print**: Creates a stepped charge summary bill line that will appear on the customer's bill.
      - **Summarize Step Lines, Do Not Print**: Creates a stepped charge summary bill line that will be shown on Bill Segment - Calculation Details, but not appear on the customer's bill.
  - **Step Calculation Details**:
    - **Seasonal**: See Common Parameters.
    - **Step Multiplier Algorithm**: Specifies the algorithm that will be used calculate the step boundaries. Used only if step boundaries are dynamic and are calculated based on something that's only known at billing time.
    - **Values**: Defines the values associated with each of the ranges (steps) of consumption used by the rule. For example, if the charge is applicable to the first 300 kilowatt-hours, you would enter **Step Start** 0 and **Step End** 300. When multiple tiers exist, the **Step End** of the first tier is the **Step Start** of the second tier (and so on). To extend our example, the second tier would have a **Step Start** of 300; the **Step End** may be left blank (the system assumes a blank **Step End** means "to infinity"). When defining negative steps, the **Step Start** value must be higher than the **Step End**. For example, if a step ranged from -200 to -400, -200 is a higher value than -400 so the **Step Start** would be -200 and the **Step End** would be -400.
      - **Sqeuence**: Used by the rule (along with the **Step Start** and **Step End**) to order the evaluation of the billable service quantity against the tier. This is particularly important and should be set properly for negative steps which must be configured from high to low.

Copyright © 2026, Oracle and/or its affiliates.

        * **Step Start**: Defines the start value for the consumption range. This field supports values of up to 14 digits and 4 decimal places (e.g. a maximum value of 99,999,999,999,999.9999).
          You can indicate that the **Step Start** of a bill line should be substituted into the bill line's description by using the %L substitution variable in the calculation rule's **Description On Bill** field.
        * **Step End**: Defines the end value for the consumption range. This field supports values of up to 14 digits and 4 decimal places (e.g. a maximum value of 99,999,999,999,999.9999).
          You can indicate that the **Step End** of a bill line should be substituted into the bill line's description by using the %H substitution variable in the calculation rule's **Description On Bill** field.
        * **Value Type**: Defines the type of price to be applied to the consumption range in the rule's calculations. Valid options include Charge, or Unit Rate.
        * **Value Source**: Defines the source of the price to be applied to the consumption range in the rule's calculations. Depending on the source selected, this will be a Bill Factor, Value, or Algorithm.
        * **Value/Bill Factor/Algorithm**: Specifies the Bill Factor, Value, or Algorithm used when calculating charges for the consumption range.

- **Example**: This calculation rule calculates charges for electric service based on stepped charges for kWh consumption during the Fall
  Calculation Group: Electric Residential - Stepped

  Calculation Rule: KWHSTFL

  Sequence: 10

  Description: Fall kWh Stepped Charges

  Description On Bill: Usage %Q kWh at %I per kWh %F

  Print: Yes

  Print if Zero: No

  Rounding Type: Nearest

  Precision: 0.01000

  FCPO: No

  Override Proration: No

  Calculation Details:

  – **Unit of Measure**: Kilowatt hour
  – **Error If No Value**: No
  – **Measures Peak**: No
  – **GL Statistical Qty**: Yes
  – **Create Step Summary**: Create Summary, Do Not Print
    Step Calculation Details:
  – **Seasonal**: Yes
  – **Seasonal Prorate Method**: Accounting Date
  – **Season Start**: 09-01
  – **Season End**: 11-30

Copyright © 2026, Oracle and/or its affiliates.

- **Value 1**:
  - **Low Value**: 0
  - **High Value**: 300
  - **Value Type**: Unit Rate
  - **Value Source**: Value
  - **Value / Bill Factor / Algorithm**: 0.0485
- **Value 2**:
  - **Low Value**: 300
  - **High Value**: [blank]
  - **Value Type**: Unit Rate
  - **Value Source**: Value
  - **Value / Bill Factor / Algorithm**: 0.0564
    GL Distribution:
- **Distribution Source Code**: Specific Distribution Code
- **Distribution Code**: Revenue - Electric - Residential
  Cross Reference:
- **Member Category (Category Type / Value)**: Revenue / Consumption Charge

Summary

Summary calculation rules create a "subtotal" on the bill. It exists purely for reporting and informational purposes. Summary calculation rules must define the respective target category that they are applied to. The calculation rule(s) that compute charges should be included in the summary must be defined as members of the target category. See **Cross-Reference** under Common Parameters for more information about Member and Target categories.

- **Rule Name**: Summary
- **Base Package Calculation Rule Business Object**: C1-Summary
- **Apply Calculation Rule Algorithm Type / Algorithm**: C1-SUMMARY
- **Rule Parameters**:
  - **Calculation Details**: Details regarding how the calculation rule performs its calculations:
    - **Error If No Value**: Indicates what should happen if the system cannot find any calculation rules that are members of the target category at billing time.
- **Example**: This calculation rule creates a consumption summary on the bill based on other calculation rules included in the rate that belong to the "Revenue / Consumption Charge" category.
  Calculation Group: Electric Residential Charges

  Calculation Rule: USGSUM

  Sequence: 30

  Description: Usage Summary

  Description On Bill: Summary

  Print: Yes

Copyright © 2026, Oracle and/or its affiliates.

Print if Zero: No

Rounding Type: Nearest

Precision: 0.01000

FCPO: No

Calculation Details:

- **Error If No Value**: No
  Cross Reference:
- **Target Category (Category Type / Value)**: Revenue / Consumption Charge
- **Member Category (Category Type / Value)**: Summary / Summary of Consumption Charge

Rate Check

This section describes the features and functionality of Rate Check.

- About Rate Check
- Performing a Rate Check
- Viewing the Results of a Rate Check

About Rate Check

The **Rate Check** feature allows you to test your rate schedules under different scenarios. For example, you can test your rates for an existing account using previous bill segments, or by providing user-specified bill period and service quantities.

Performing a Rate Check

> **Note**
>
> The Rate Check portal works with both calculation rule-based and component-based rate schedules.

Use this procedure to perform a rate check.

- Select **Menu**, and then **Rates**, and then **Rate Check.**
- Select the **Rate Schedule** for the rate check.
- Select the **Service Agreement** for the rate check.
- Select the **Bill Segment** for the rate check (if you wish to base the rate check on an existing bill segment).
- Enter the **Start Date**, **End Date**, and **Accounting Date** for the rate check.
- Select the **Revenue Class** for the rate check.
- If you are basing the rate check on user-specified **Service Quantities**, enter the following for each as applicable:

Copyright © 2026, Oracle and/or its affiliates.

  - **UOM**: The unit of measure applicable for the rate check (based on the specified rate schedule)
  - **TOU**: The time of use applicable for the rate check (based on the specified rate schedule)
  - **SQI**: The service quantity identifier applicable for the rate check (based on the specified rate schedule)
  - **Measured Quantity**: The amount of the service quantity specified.
  - **Service Point ID**: The service point which recorded the service quantities, if applicable Note: If you specified a Service Agreement with a Bill Segment and/or Start and End Dates, this section is populated based on the selected data, but can be manually updated.
- If you are basing the rate check on user-specified **Items**, enter the following for each as applicable:
  - **Item Type**
  - **Start Date**: The start date of the consumption period.
  - **End Date**: The end date of the consumption period
  - **Item Count**: The number of items for the specified Item Type.
  - Note: If you specified a Service Agreement with a Bill Segment and/or Start and End Dates, this section is populated based on the selected data, but can be manually updated.
- Specify characteristic types and values for any **Characteristics** are applicable to the rate check. Characteristic types and values are used to apply appropriate bill factor values during rate processing.
- If you are using a calculation rule-based rate schedule, specify if the system **Should Trace** the process. This produces logs that outline the calculation groups and rules used by the rate schedule.
- Perform the rate check:
  - If you wish to perform the rate check using a specified bill segment or service quantities or items, click **Check Rate**.
  - If you wish to execute the rate check using billing determinants retrieved from an external system (such as Oracle Utilities Meter Data Management), select the appropriate function from the **Select Function** dropdown list.

Refer to the Rate Check portal's embedded help, for more information.

Viewing the Results of a Rate Check

The outcome of the rate check process is displayed on the **Results** tab. This tab contains the following zones:

- **Main**: Displays results of the rate check, including:
  - **Rate Schedule**: The rate schedule used in the rate check
  - **Revenue Class**: The revenue class used for the rate check.
  - **Bill Period**: The start and end dates for the bill period used by the rate check.
  - **Consumption Period**: The start and dates for the consumption period used by the rate check.

Copyright © 2026, Oracle and/or its affiliates.

  - **Accounting Date**: The Accounting Date used by the rate check.
  - **Calculated Amount**: The calculated total for the rate check
- **Calculation Lines**: Displays the bill calculation headers and lines for the rate schedule.
- **Read Details**: Displays meter read details for the consumption used in the rate calculation.
- **SQ Details**: Displays a list of service quantities used in the rate calculation.
- **Characteristics**: Displays characteristics referenced by the rate calculation (if applicable).
- **Trace Log**: Displays logs that outline the calculation groups and rules used by the rate schedule. This zone is populated only if using a calculation rule-based rate schedule, and if the Should Trace dropdown list is set to "No".

Effective Dates and Price Proration

This section describes how proration applies to different types of calculation rules. Proration implies assessing charges proportionately.

The term "proration" describes two different issues:

- Prorating a charge whose value changes during a bill period. For example, if a tax rate changes during a bill period and you've indicated that such changes should be prorated, the system prorates the tax change (e.g., 20 days at 5% and 10 days at 6%).
- Prorating charges when the time period being billed is not in sync with the time period in which the charges were defined in the rate. For example, if a rate contains a flat monthly charge and the bill period spans two months, the flat charge must be prorated.

This is a complicated topic as it's possible for many proration issues to exist on a single bill. For example, on a single bill:

- The rate structure can change several times during the bill period (i.e., multiple rate versions are effective).
- The taxing authority changes the tax rate.
- The customer's tax exemption changes.
- The customer-specific price can change.
- etc.

- Types of Proration
- Proration Factors
- Overriding Proration Factors
- Service Quantity Proration
- Calculation Rule Value Proration
- Bill Factor Value Proration
- Example of Generating Prorated Bill Calculation Lines
- Seasonal Proration
- Price Changes and Interval Quantities

Copyright © 2026, Oracle and/or its affiliates.

Types of Proration

The following section describes the types of proration performed by applying a rate.

NOT_SUPPORTED

For more information on how prorated charges are presented on the printed bill, refer to How To Use Description On Bill.

- Consumption Period Proration
- Calculation Period Proration
- Bill Factor Proration
- A Service Agreement's Rate Changes Are Not Prorated

Consumption Period Proration

A rate schedule specifies its frequency. Bill segments generated using a rate are expected to have a consumption period that corresponds to this frequency. When a bill segment's consumption period falls outside of the rate frequency's normal period, proratable service quantities may be adjusted to match the normal period for calculation. The service quantities are then re-adjusted based on calculation period proration.

NOT_SUPPORTED

Refer to Service Quantity Proration for more information about these calculations.

Calculation Period Proration

There can be multiple versions of a given rate over time. Each version has an effective date. When the system detects that a rate schedule has multiple versions of a rate in effect during a bill period, it may or may not prorate the charges. You control exactly how the system handles this situation using the rate schedule's proration parameters.

If multiple rate versions are effective during the consumption period and the rate schedule is configured to prorate the rate versions, a given bill segment will contain a set of bill calculation lines for each rate version. You should consider quite carefully how to handle this situation on your printed bills as rate version proration can confuse a customer.

NOT_SUPPORTED

Refer to Defining Calculation Rules for more information.

Copyright © 2026, Oracle and/or its affiliates.

Bill Factor Proration

A rate contains calculation rules. The primary reason a calculation rule exists is to define how much to charge for the services sold by your company. When you create calculation rules, you have two ways to indicate how much to charge:

- You can specify the value directly in a calculation rule.
- You can have the calculation rule use a bill factor that contains the value.

Bill factors are effective-dated (meaning, their values may change over time). When the system detects that a calculation rule's bill factor has multiple values in effect during a bill period, it may or may not prorate the charges. You control exactly how the system handles this situation using the bill factor's proration parameters.

> **NOT_SUPPORTED**
>
> Refer to Defining Bill Factors and Bill Factor Value Proration for more information.

- Tax Exemption Proration
- Contract Value Proration
- Contract Rider Proration

Tax Exemption Proration

Bill factors that levy taxes may or may not be eligible for tax exemptions; it's up to the respective taxing authority. Customers who are eligible for tax exemptions will have their tax obligation reduced (or nullified) at billing time.

Whether or not a customer is tax exempt (and the percent exempt) is defined on the service agreement's tax exemption page. A customer's tax exemptability is effective-dated (it may change over time). If a customer's tax exemptability changes during a bill period, the system may or may not prorate the charges. You control exactly how the system handles this situation using the bill factor's proration parameters.

Contract Value Proration

Some types of bill factors do not have the same value for all customers. For example, you may have rates where the price per kilowatt-hour can be different for every customer using the rate.

For bill factors whose value is customer-specific, the value is defined on the service agreement's contract value page. The contract value is effective-dated (it may change over time). If a customer's contract value changes during a bill period, the system may or may not prorate the changes. You control exactly how the system handles this situation using the bill factor's proration parameters.

Copyright © 2026, Oracle and/or its affiliates.

Contract Rider Proration

Contract riders are used when the charge associated with a bill factor is only applicable to some customers for some period of time. For example, if a bill factor is used to provide a discount during the first three months of service, then a contract rider would be used.

A customer's contract rider applicability period is defined on the service agreement's contract rider page. A rider's applicability may change during a customer's bill period. You control exactly what the system does when a rider's applicability changes during a bill period using the bill factor's proration parameters.

A Service Agreement's Rate Changes Are Not Prorated

A service agreement's rate is effective-dated (it may change over time). When the system calculates a service agreement's bill segment when multiple rates are effective during the bill period, it will use a single rate from the service agreement's rate history. The system chooses the rate as per the service agreement's SA type's rate selection date.

NOT_SUPPORTED

Refer to SA Type - Rate for more information.

Proration Factors

The purpose of calculation rules is to generate bill calculation lines. When rate application builds bill calculation lines, it may also prorate the charges.

There are multiple types of proration performed by rate application. Service quantities, step boundaries, rate values, and bill factor values may all be prorated as part of applying a rate.

We need to establish some terms in order to understand the calculations.

- Normal Days
- Consumption Period Factor
- Calculation Period Factor
- BF Value Period Factor

Normal Days

As mentioned previously, the rate schedule references a frequency that specifies the number of times per year that service agreements referencing the rate are expected to bill. The following formula dictates a rate frequency's normal days: 365 days / number of periods per year.

Example:

Copyright © 2026, Oracle and/or its affiliates.

- For a monthly frequency (12 periods per year), 365 / 12 = 30 days.
- For a quarterly frequency (4 periods per year), 365 / 4 = 91 days.

NOT_SUPPORTED

Refer to Defining Frequency Codes for more information.

Consumption Period Factor

Before applying a rate, the bill segment's consumption period is determined. The ratio of normal days to consumption period days is the consumption period factor.

The system uses the Minimum and Maximum Days Offset on the rate's Frequency to determine whether or not the calculation rule should be prorated due to a short or long consumption period.

If the number of days in the consumption period is within the tolerance of days determined by the specified on the rate's frequency, the consumption period factor is 1 (not applicable).

Example:

If the consumption period spans 60 days and the rate's normal days equals 30 days then the consumption period factor is:

- 30 days / 60 days = 0.5

If the consumption period spans 31 days and the rate's normal days equals 30 days and the maximum offset is 3 days, the consumption period factor is 1 (not applicable) because the consumption period days falls within the tolerance specified on the frequency.

NOT_SUPPORTED

For more information on how a bill segment's consumption period is determined, refer to How the System Requests How Much Was Consumed. For more information on specifying minimum and maximum offset days on frequency, refer to Defining Frequency Codes.

Calculation Period Factor

The calculation period is the date range where a rate version is effective during the consumption period. For each rate version that is effective during the consumption period, a bill segment calculation period is created. A set of bill calculation lines is created for each calculation period.

A calculation period factor is determined for each calculation period. The calculation period factor is the ratio of calculation period days to normal days. If the consumption period factor is 1 (not applicable), then the calculation period factor is the ratio of calculation period days to consumption period days.

Example:

Copyright © 2026, Oracle and/or its affiliates.

If the consumption period covers April 1 to April 30 and there was a new effective rate version as of April 21, there are two calculation periods.

- Calculation period April 1 to April 20, calculation period factor = 20 days / 30 days = 0.6666667.
- Calculation period April 21 to April 30, calculation period factor = 10 days / 30 days = 0.3333333.

BF Value Period Factor

The bill factor value period is the date range where a proratable bill factor value is effective during the calculation period.

When a bill factor value changes during the calculation period, the results is that multiple bill calculation lines are generated for the same calculation rule.

The ratio of the number of days within the calculation period that a distinct value is effective to the number of days in the calculation period determines the BF value period factor. When the bill factor is set up not to prorate, the BF value period factor is 1 (not applicable).

Example:

If the state tax rate changes from 6% to 6.5% on April 16 and the calculation period is April 1 to April 30, there are two bill factor value periods.

- Bill factor value period April 1 to April 15, BF value period factor = 15 days / 30 days = 0.5.
- Bill factor value period April 16 to April 30, BF value period factor = 15 days / 30 days = 0.5.

NOT_SUPPORTED

Refer to Defining Bill Factors for more information on specifying whether the bill factor allows proration.

Overriding Proration Factors

If the system logic for determining proration factors does not satisfy your requirements, you may plug in an override proration algorithm to calculate the proration factors as required by your implementation.

- For Calculation Rule-Based Rates, the override proration algorithm is specified in the calculation rule's business object.
- For Component-Based Rates, the overriding proration algorithm is specified in the Installation record.

The system is delivered with an algorithm type that sets the proration factors to 1 (not applicable) when a predefined characteristic is specified on a calculation rule. Refer to OVPF-NOPROR for more information.

Copyright © 2026, Oracle and/or its affiliates.

Service Quantity Proration

When a bill calculation line is generated for a service quantity type calculation rule, the applicable service quantity is multiplied by the consumption period factor. That product is then multiplied by the calculation period factor to determine the prorated service quantity for the bill calculation line.

Example:

Customer's electricity bill segment's consumption period is from April 1 through May 15. A new rate version for the rate schedule is effective as of April 16. 100 kWh is consumed.

Assuming a monthly frequency, if the consumption period spans 45 days and the first calculation period spans 15 days, and the applicable service quantity is 100 kWh; then:

- Consumption period factor = 30 normal days / 45 days = 0.6666667.
- Calculation period factor = 15 days / 30 normal days = 0.5.
- Prorated service quantity = 100 kWh x 0.6666667 x 0.5 = 33 kWh.

The next calculation period extends to the end of the consumption period (30 days):

- Consumption period factor does not change = 0.6666667.
- Calculation period factor = 30 days / 30 days = 1.0.
- Prorated service quantity = 100 kWh x 0.6666667 x 1.0 = 67 kWh.

- Measures Peak
- Step Proration
- Step Multipliers

Measures Peak

If the calculation rule specifies that the service quantity measures peak, then service quantity proration is not performed. Although service quantities are not prorated if a calculation rule measures peak, calculation rule values (like price per kW) are prorated.

NOT_SUPPORTED

Refer to Calculation Rule Value Proration for more information.

Step Proration

If a calculation rule's charges are based on stepped pricing then the step boundaries are also prorated (as long as the calculation rule is not marked as measures peak). Step boundaries are multiplied by the calculation period factor to determine the adjusted steps boundaries.

Example:

Copyright © 2026, Oracle and/or its affiliates.

Assuming a monthly frequency, if the calculation rule's step boundaries are 100 kWh to 300 kWh, and the calculation period is 45 days; the step boundaries are prorated as:

- 45 days / 30 days x 100 kWh = 150 kWh.
- 45 days / 30 days x 300 kWh = 450 kWh.

Step Multipliers

If the calculation rule specifies that the steps be multiplied by the result value of another calculation rule, step proration is not performed. It is assumed that the result value from the referenced calculation rule has already been prorated, as the referenced step calculation rule is subject to Calculation Rule Value Proration.

Calculation Rule Value Proration

Except for summary type calculation rules (which are not prorated at all), a value is specified on the calculation rule. The value may be a unit rate, percentage or charge. The value may be specified directly on the calculation rule, derived from a bill factor, derived from another calculation rule, or calculated by an algorithm.

Regardless of where the calculation rule value comes from, it will be multiplied by the calculation period factor when determining the charge on a bill calculation line. Exceptions to this rule are:

- Unit rate values where the calculation rule's measures peak attribute is not checked.
- Percentage values.
- Calculation rule values derived from another calculation rule.

Even though the calculation rule values are not prorated for unit rate and percentage value types, if these rate values change during the calculation period due to a bill factor value change, bill factor value proration still applies.

Values derived from another calculation rule are not subject to rate value proration, as calculation rule value proration was already performed on the referenced calculation rule. Again, if the bill factor value changed on the referenced calculation rule, bill factor value proration still applies.

Example:

A calculation rule charges $2.00 per kW. The rate schedule's normal days equals 30, and the calculation period spans 15 days. The peak kW service quantity is 50 kW.

- The calculation period factor is 0.5 (15 days / 30 days).
- Calculation rule value proration yields $2.00 x 0.5 = $1.00 per kW.

NOT_SUPPORTED

For more information on bill factor value proration, refer to Bill Factor Value Proration.

Copyright © 2026, Oracle and/or its affiliates.

Bill Factor Value Proration

Bill factor value proration behaves the same way any time a bill factor value changes within a calculation period. This could be due to a change in a tax rate, a contract rider becoming effective, a tax exemption expiring, etc.

Multiple bill calculation lines are generated for a given calculation rule when a proratable bill factor value changes during the calculation period.

Example:

If the state tax rate changes from 6% to 6.5% on April 16, and the calculation period is April 1 to April 30.

- The first BF value factor period April 1 to April 15, so BF value period factor is 0.5.
- The value used in the first bill calculation line for state tax is 6% x 0.5 = 3%.
- The second bill factor value period is April 16 to April 30, so the BF value period factor is 0.5.
- The value used in the second bill calculation line for state tax is 6.5% x 0.5 = 3.25%.

Example of Generating Prorated Bill Calculation Lines

Let's say that 1200 kWh was consumed from March 17 to April 30. The peak kW for the consumption period is 20 kW.

A monthly rate schedule E-COM1 exists with two rate versions.

Rate version effective prior to April 1 has the following components:

- $0.05 per kWh.
- $1.00 per kW.

Rate version effective as of April 1 has the following calculation rules:

- $0.04 per kWh; as of April 16, $0.07 per kWh (new effective bill factor value).
- $1.50 per kW; as of April 16, $1.25 per kW (new effective bill factor value).

Calculation Period March 17 to March 31

The consumption period spans 45 days, so the consumption period factor is 30 normal days / 45 days = 0.6666667.

This calculation period spans 15 days, so the calculation period factor is 0.5.

Service quantity proration for the first calculation period yields:

- 1200 kWh x 0.6666667 (consumption period factor) x 0.5 (calculation period factor) = 400 kWh.
- 20 kW x 1 (not applicable) = 20 kW.

Calculation rule value proration for the first calculation period yields:

- $0.05 x 1 (not applicable) = $.05 per kWh.
- $1.00 x 0.5 (calculation period factor) = $0.50 per kW.

Copyright © 2026, Oracle and/or its affiliates.

First Set of Bill Calculation Lines

Bill calculation line 1: 400 kWh x $0.05 = $20.00.

Bill calculation line 2: 20 kW x $0.50 = $10.00.

Calculation Period April 1 to April 30

This calculation period spans 30 days, so the calculation period factor is 1.

Service quantity proration for the calculation period yields:

- 1200 kWh x 0.6666667 (consumption period factor) x 1 (calculation period factor) = 800 kWh.
- 20 kW x 1 (not applicable) = 20 kW.

Calculation rule value proration for the first calculation period yields:

- $0.04 x 1 (not applicable) x 0.5 (BF value period factor) = $0.02 per kWh.
- $0.07 x 1 (not applicable) x 0.5 (BF value period factor) = $0.035 per kWh.
- $1.50 x 1 (calculation period factor) x 0.5 (BF value period factor) = $0.75 per kW.
- $1.25 x 1 (calculation period factor) x 0.5 (BF value period factor) = $0.625 per kW.

Second Set of Bill Calculation Lines

Bill calculation line 1: 800 kWh x $0.02 per kWh = $16.00.

Bill calculation line 2: 800 kWh x $0.035 per kWh = $26.00.

Bill calculation line 3: 20 kW x $0.75 per kW = $15.00.

Bill calculation line 4: 20 kW x $0.625 per kW = $12.50.

Seasonal Proration

Seasonal proration adds another layer of complexity. If calculation rules are specified to prorate seasonally, the seasonal period specified on a given seasonal calculation rule dictates how a calculation rule's value or associated service quantity is prorated.

There are two proration methods, Prorate and Prorate Seasonal SQ, which may be specified on seasonal calculation rules. Both of these proration methods behave the same way with respect to calculation rule value proration. If a given seasonal calculation rule is eligible for Calculation Rule Value Proration, the ratio of days in the season that are applicable during the calculation period to the days in the calculation period determines the seasonal factor. When it comes to service quantity proration, the two proration methods are different. If a given seasonal calculation rule is eligible for Service Quantity Proration, it prorates the service quantities as follows:

- Prorate. A seasonal factor is calculated as the ratio of days in the season that are applicable during the calculation period to the days in the calculation period. This ratio is applied along with the consumption period factor and calculation period factor to determine the prorated service quantity.
- Prorate seasonal SQ. In this case, it is assumed that either the meter measures seasonal consumption on separate registers or the consumption has been split into separate service quantities by other means (like an SQ rule). To determine a prorated seasonal service quantity, the seasonal proration factor is calculated as for a Prorate seasonal calculation rule. The seasonal factor is then multiplied by another ratio of total consumption period

Copyright © 2026, Oracle and/or its affiliates.

days to days in the season that are applicable for the total consumption period. This modified seasonal factor is applied along with the consumption period factor and calculation period factor to determine the prorated service quantity.

> **Note**
>
> **Prorate seasonal SQ**. Specify this option only if you can determine the seasonal consumption as an independent service quantity for use by rate application. Because the service quantities themselves dictate the applicability of the calculation rule, a given service quantity need not appear in rate versions effective outside the season (i.e. a rate version effective on October 1 may not include calculation rules that charge for kWh/summer).
>
> **Override Seasonal Proration**. If the seasonal functionality provided with the system does not work for your organization, you may override the logic using the Override Seasonal Proration plug-in spot on the installation record. For example, perhaps the seasonal period is determined dynamically based on the scheduled meter read date of the bill cycle.

Example (Prorate):

Let's say that 600 kWh and 50 kW are the billable service quantities for a consumption period from April 1 to April 30. Prorate seasonal calculation rules are defined on a monthly rate that charge $0.05 per kWh and $0.75 per kW from January 1 to April 15. Prorate seasonal calculation rules are defined on the same rate version that charge $0.06 per kWh and $0.80 per kW between April 16 and December 31. Assume only one rate version is effective for the consumption period.

- The consumption period factor is 1 (not applicable).
- The calculation period factor is 1.0.
- The seasonal factor for the kWh calculation rule whose season is effective from January 1 to April 15 is 0.5 (15 season days in calculation period / 30 days in the calculation period).
- The resulting bill calculation line uses service quantity proration:
- 600 kWh x 0.5 (seasonal factor) x $0.05 = $15.00.
- The seasonal factor for the kW calculation rule whose season is effective from January 1 to April 15 is also 0.5.
- The resulting bill calculation line uses calculation rule value proration:
- 50 kW x $0.75 x 0.5 (seasonal factor) = $18.75.
- The seasonal factor for the kWh calculation rule whose season is effective from April 16 to December 31 is 0.5.
- The resulting bill calculation line uses service quantity proration:
- 600 kWh x 0.5 (seasonal factor) x $0.06 = $18.00.
- The seasonal factor for the kW calculation rule whose season is effective from April 16 to December 31 is also 0.5.
- The resulting bill calculation line uses calculation rule value proration:
- 50 kW x $0.80 x 0.5 (seasonal factor) = $20.00.

Example (Prorate seasonal SQ):

Copyright © 2026, Oracle and/or its affiliates.

Let's say that a given meter measures winter consumption separate from summer consumption. The meter itself measures kWh/summer on a register between June 21 and September 18, and measure kWh/winter on another register at other times of the year.

We are generating a bill segment with a consumption period from September 2 to October 30. A monthly rate schedule has two rate versions effective during the consumption period. The latest rate version is effective on October 1.

Each of the rate version calculation groups have Prorate seasonal SQ seasonal calculation rules defined. The first rate version's calculation rules charge $0.06 per kWh/summer and $0.05 per kWh/winter. The second rate version calculation group has a single calculation rule that charges $0.05 per kWh/winter.

Assume that the billable service quantities are 800 kWh/summer and 1600 kWh/winter.

First set of bill calculation lines (September 2 to September 30):

- The consumption period factor is 30 normal days / 60 consumption days = 0.5.
- The calculation period factor is 30 calculation days / 30 normal days = 1.0.
- calculation rule 1 ($0.06 per kWh/summer):
- The seasonal factor 17 seasonal days / 30 calculation days = 0.5666667.
- The modified seasonal factor is 0.5666667 x 60 consumption days / 17 seasonal days = 2.0.
- The resulting bill calculation line:
- 800 kWh/summer x 0.5 (consumption period factor) x 2.0 (modified seasonal factor) x $0.06 = $48.00.
- calculation rule 2 ($0.05 per kWh/winter):
- The seasonal factor is 12 seasonal days / 30 calculation days = 0.4.
- The modified seasonal factor is 0.4 x 60 consumption days / 42 seasonal days = 0.571428.
- The resulting bill calculation line:
- 1600 kWh/winter x 0.5 (consumption period factor) x 0.571428 (modified seasonal factor) x $0.05 = $22.86.

Second set of bill calculation lines (October 1 to October 30):

- The consumption period factor is 30 normal days / 60 consumption days = 0.5.
- The calculation period factor is 30 calculation days / 30 normal days = 1.0.
- Calculation rule 1 ($0.05 per kWh/winter):
- The seasonal factor 30 seasonal days / 30 calculation days = 1.0.
- The modified seasonal factor is 1.0 x 60 consumption days / 42 seasonal days = 1.428571.
- The resulting bill calculation line:
- 1600 kWh/winter x 0.5 (consumption period factor) x 1.428571 (modified seasonal factor) x $0.05 = $57.14.

NOT_SUPPORTED

For more information on specifying seasonal proration options, refer to Defining Calculation Rules.

Copyright © 2026, Oracle and/or its affiliates.

Price Changes and Interval Quantities

All of the above sections described situations where the service quantities will be prorated due to changes in the price during the billing period. The above sections assume that the bill contains a simple metering scenario, where reads will not exist on the date of the price break.

However, consider an interval billing example. There are two basic pricing scenarios for interval quantities:

- Interval prices are applied to interval quantities. In this case, proration is not applicable because the prices are changing at small intervals and the appropriate usage rules in Oracle Utilities Meter Data Management understand how to apply the prices to the interval quantities.
- Time of use prices. In this case, interval quantities are being mapped to time of use codes. Oracle Utilities Meter Data Management can compare interval quantities to a time of use map to create billing determinants for Oracle Utilities Customer Care and Billing. These will result in new entries in the bill segment's SQ collection when received by Oracle Utilities Customer Care and Billing. Calculation rules are then apply the appropriate price(s) for each SQ. In the time of use pricing scenario, it is possible for the prices to change during the usage or billing period. When usage requests are created by Oracle Utilities Customer Care and Billing, they will contain rate version date breaks to indicate to Oracle Utilities Meter Data Management to calculate the usage for each rate version period. Proration should not be necessary when the rate is applied for each usage (rate version) period. However, if prices are configured as bill factors and there are changes in bill factor values within the usage period, then proration may occur as the date breaks when bill factor values change are not included in the usage request from Oracle Utilities Customer Care and Billing.

Setting Up Estimated Consumption for Items

Some types of physical devices (such as, items) use power but a meter does NOT measure the power consumed. The most common example is a street light. If this situation exists for your company AND if the amount charged for the lamp varies based on the amount of estimated power used, then you will have to maintain estimated DAILY consumption for each type of item. The system uses this information at billing time to estimate how much power was used by the customer's items and then prices the estimated consumption as per the terms in the service quantity calculation rule.

Select **Admin**, and then **Rates**, and then **Item Type SQ Estimate** to define estimated consumption.

Description of Page

Select the **Item Type** using the search.

Enter the estimated daily consumption in the **Daily Service Quantity**. Each Service Quantity has an **Effective Date** so that changes in the estimated quantities may be recorded over time.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The unit of measure of the daily service quantity is displayed adjacent to the item type description. This value is defined on the item type table. Refer to Device Types for more information.

Where Used

This information is used when the system calculates a bill segment for a service agreement that references service points with items. If the items reference item types for which the system should estimate consumption, the system extracts the daily consumption amount from this table.

Setting Up Degree Days (and Average Temperatures)

Many organizations print degree-day and/or average temperature information on their customers' bills. Select **Admin**, and then **Billing**, and then **Degree Days**, and then **Search** to define these values for a given trend area.

Description of Page

The easiest way to define degree days is to select a **Trend Area** and **Month In Year** and tab out of the field. The system generates one row in the **Degree Days** collection for each day of the month. After the days exist in the collection, you should define the following for each day:

Degree Day This is a unit of measure used to express the extent to which temperatures vary from a specific reference temperature on a given date.

Maximum Temperature This is the maximum temperature on a given date.

Minimum Temperature This is the minimum temperature on a given date.

Average Temperature This is the average temperature on a given date.

Where Used

This information may be used by your bill print routines.

United Kingdom VAT and CCL

The topics in this section provide information about value added tax (VAT) and climate change levy (CCL) charges that are specific to non-domestic customers in the United Kingdom market.

> **Note**
>
> **Applicable for UK market**. This section is only relevant for the United Kingdom market. Other markets may disregard this section.

- UK VAT Overview
- UK CCL Overview

Copyright © 2026, Oracle and/or its affiliates.

- UK VAT and CCL Bill Examples
- Billing and UK VAT
- Excess Credits and UK VAT
- Setting Up The System For UK VAT and CCL

UK VAT Overview

Two rates of VAT, referred to as standard rate VAT and reduced rate VAT, are applicable to energy related charges in the UK. Domestic (i.e., residential) customers always pay VAT at the reduced rate. Non-domestic customers on the other hand normally pay standard rate VAT. However, part or all of the energy related charges for a non-domestic customer might be subject to reduced rate VAT. De minimis and VAT declarations affect the percentage of the energy related charges that is subject to each VAT rate:

- If average daily usage at a premise for a given service type does not exceed a certain threshold (the de minimis limit), all energy related charges at that premise / service type are taxed at the reduced rate.
- Some non-domestic customers, such as those with a mixed-use premise, may be eligible to pay reduced rate VAT on part or all of their energy related charges. Customers may file a VAT declaration specifying the percentage of their energy related charges that are eligible for reduced rate VAT - refer to Maintaining Declarations for more information. If the declared percentage exceeds a given threshold, the declared percentage is deemed to be 100% and the customer pays VAT at the reduced rate on all energy related charges at that premise. VAT declarations are non-transferable and must be filed for each account, premise, and service type combination.

In addition to the potential for different rates of VAT to be applicable on a bill, UK tax regulations require that excess credits are considered a prepayment of energy related charges together with VAT. Refer to Excess Credits and UK VAT for more information.

The system comes supplied with various algorithms types that can be used to perform the VAT calculations.

UK CCL Overview

The climate change levy (CCL) is based on the amount of energy used that is subject to standard rate VAT. Similarly to VAT declarations, customers may file for exemption from CCL. A CCL declaration specifies the percentage of the CCL charges that the customer is exempt from - refer to Maintaining Declarations for more information. CCL declarations are non-transferable and must be filed for each account, premise, and service type combination.

CCL charges themselves are subject to standard rate VAT.

UK VAT and CCL Bill Examples

The following examples show how a bill for a non-domestic customer is affected by de minimis, VAT declarations, and CCL declarations. In the examples, standard rate VAT of 17.5%, reduced rate VAT of 5%, a VAT declaration threshold of 60%, and a CCL charge of 0.43p per unit of energy are used.

Copyright © 2026, Oracle and/or its affiliates.

- Example 1 - Normal Account
- Example 2 - Account with Consumption Under the De Minimis Limit
- Example 3 - Account with VAT Declaration
- Example 4 - Account with VAT Declaration and CCL Declaration

Example 1 - Normal Account

This example shows the bill for a normal non-domestic account with no declarations and consumption above the de minimis limit. Standard rate VAT is applied to all energy related charges (the standing charge and the per unit charge) and to the CCL charge.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 2,000 units @ 0.43p	8.60
VAT @ 17.5% on 218.60	38.25
Total	256.85

Example 2 - Account with Consumption Under the De Minimis Limit

In this example, consumption does not exceed the de minimis limit and therefore reduced rate VAT is applied to all the energy related charges. There is no climate change levy because only units subject to standard rate VAT are subject to CCL.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
VAT @ 5% on £210.00	10.05
Total	220.05

Example 3 - Account with VAT Declaration

This example is for a non-domestic account with a VAT declaration of 20%. Consumption is above the de minimis limit. CCL applies to only 80% of the total units as 20% is subject to reduced rate VAT and therefore exempt. 80% of the energy related charges (the standing charge and the per unit charge) and all of the CCL charge are subject to standard rate VAT. 20% of the energy related charges are subject to reduced rate VAT.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 1,600 units @ 0.43p	6.88
VAT @ 17.5% on £174.88	30.60

Copyright © 2026, Oracle and/or its affiliates.

VAT @ 5% on £42.00	2.10
Total	249.58

Example 4 - Account with VAT Declaration and CCL Declaration

This example is for a non-domestic account with a VAT declaration of 20% and a CCL declaration of 10%. Consumption is above the de minimis limit. CCL applies to only 80% of the total units as 20% is subject to reduced rate VAT and therefore exempt. The customer gets a credit (CCL relief) for 10% of the CCL charges as a result of the CCL declaration. 80% of the energy related charges (the standing charge and the per unit charge) and all of the CCL charge less the CCL reliefs are subject to standard rate VAT. 20% of the energy related charges are subject to reduced rate VAT.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ 10p	200.00
CCL on 1,600 units @ 0.43p	6.88
CCL relief	<0.68>
VAT @ 17.5% on £174.20	30.48
VAT @ 5% on £42.00	2.10
Total	248.78

Billing and UK VAT

The following sections describe how VAT rules are implemented when a bill is produced.

- Application of De Minimis
- Calculation of VAT
- Calculation of CCL

Application of De Minimis

The de minimis rule specifies that all energy related charges at a premise be subject to reduced rate VAT if the total usage at a premise does not exceed a certain threshold. To determine if de minimis applies, the total billed consumption for a premise must therefore be known. However, energy usage at a premise may be measured with several meters and billed by multiple service agreements. To ensure that the final billed consumption for the premise is used to determine if de minimis applies, all service agreements for an account of a given service type at a given premise should be billed together and de minimis should be checked after the bill segments have been generated. In addition, all bill segments on one bill for consumption at a given premise must be for the same bill period.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bill segments from different periods may not appear on the same bill.** If a bill segment is canceled, all bill segments associated with consumption at a given premise for the same service type must be canceled and re-billed together. You cannot re-bill the bill segment on a bill for a different period.

An **Apply De Minimis Rule** Customer Class – Pre Bill Completion algorithm type CPBC-DMCH is supplied with the base package that checks for de minimis at bill completion time. The base bill completion algorithm type checks that for any service agreement that was billed, bill segments for all the account's service agreements with the same service type and characteristic premise exist on the bill. It then calculates the total consumption for the premise and determines if de minimis applies. If de minimis applies, it sets a bill characteristic whose value is the premise ID to indicate that de minimis applies and regenerates the bill segment. Refer to the **Apply De Minimis Rule** algorithm type CPBC-DMCH for more information on how this type of algorithm operates.

For calculation rule-based rates, the **Apply VAT to Cross-Referenced Charges** calculation rule business object (**C1–ApplyVatToCrossRefCharge**) calculates standard and reduced rate VAT and applies the de minimis rule if the bill characteristic for de minimis is found. Refer to Calculation of VAT for more information. Note that during the initial generation of each bill segment, the characteristic will not exist and standard rate VAT will be applied. This means that if you look at a bill before it is complete, VAT may not be accurately reflected.

> **Note**
>
> **Batch billing cannot regenerate re-billed bill segments.** If you cancel and rebill a bill segment, you should complete the bill on-line so that the re-billed bill segments can be regenerated if de minimis applies.

Calculation of VAT

To calculate VAT, the percentages of energy related charges subject to standard rate VAT and reduced rate VAT must be determined. As these percentages vary from customer to customer and even from one bill to the next for the same customer, their calculation must take place at billing time and is handled by the following:

For calculation rule-based rates, the **Apply VAT to Cross Referenced Charges** business object (**C1–ApplyVatToCrossRefCharge**) is provided to calculate the percentages of energy related charges subject to standard rate and reduced rate VAT, taking into account de minimis and any VAT declaration that is in effect for the service type, account, and premise. The percentages can then be applied to the appropriate charges that are cross-referenced.

Calculation of CCL

CCL is a charge per unit of energy subject to standard rate VAT. It is therefore dependent on the same percentage of energy related charges subject to standard rate VAT determined during the calculation of VAT.

Copyright © 2026, Oracle and/or its affiliates.

For calculation rule-based rates, a **Calculate Climate Levy and Relief Charges** calculation rule business object (**C1–ApplyCCLCharges**) is provided to calculate CCL charges and CCL relief, taking into account any CCL declarations that are in effect for the service type, account, and premise.

Excess Credits and UK VAT

When a financial transaction that results in a credit balance for a service agreement is frozen, the amount of the excess credit must be accounted for as a prepayment of energy related charges and VAT. VAT is calculated at the reduced rate for domestic customers and at the standard rate for non-domestic customers. When the excess credit is used, the VAT liability is reversed.

A **Create Excess Credit GL Details** Customer Class FT freeze algorithm type CFTZ-VAT-GL is provided to create additional GL detail entries for unbilled revenue and VAT liability when the freezing of an FT results in a credit balance or a change to a service agreement's credit balance.

- Excess Credit GL Accounting Example

Excess Credit GL Accounting Example

The following example shows the additional GL details that are created when a service agreement's balance changes and the starting or ending balance is a credit. For this example, a single VAT rate of 10% is used simply to illustrate the principle.

The service agreement has a zero starting balance. Note the following:

- When the first payment is received, the overpayment results in a credit balance and additional GL entries are created to recognize the unbilled energy revenue and unbilled VAT liability.
- When the second bill segment is created, the credit balance is reduced and additional GL entries are created to back out the unbilled GL entries, up to the amount of the credit balance.
- When the bill segment is canceled, the service agreement again has a credit balance and additional GL entries are created to recognize the unbilled energy revenue and unbilled VAT liability.

Event	Normal GL Accounting	Additional GL Accounting	SA Balance
Bill segment for £110 created	A/R 110 Revenue <100> VAT <10>		110
Payment of £330 is received	Cash 330 A/R <330>	A/R 220 Unbilled Energy <200> Unbilled VAT <20>	<220>
Bill segment for £275 created	A/R 275 Revenue <250> VAT <25>	A/R <220> Unbilled Energy 200 Unbilled VAT 20	55

Copyright © 2026, Oracle and/or its affiliates.

Payment of £55 is received	Cash 55 A/R <55>		0
Bill segment for £275 is canceled	A/R <275> Revenue 250 VAT 25	A/R 275 Unbilled Energy <250> Unbilled VAT <25>	<275>

Setting Up The System For UK VAT and CCL

The following sections provide an overview of how to configure the system to apply UK VAT and CCL charges for non-domestic customers.

- Bill Segment Freeze Installation Option
- Customer Class
- Service Agreements
- Bill Factors for UK VAT and CCL
- Characteristic Type for De Minimis Amounts
- Distribution Code
- UOM / TOU / SQI
- Service Quantities
- Algorithms for UK VAT and CCL
- Calculation Rules to Charge VAT and CCL

Bill Segment Freeze Installation Option

Select Freeze at Bill Completion as the bill segment freeze option. The system determines if de minimis applies at bill completion time and has to regenerate the bill segments as necessary.

Customer Class

Create a separate customer classes for non-domestic customers.

Service Agreements

You must configure the system so that each service agreement that must be checked for de minimis is only linked to service points at a single premise. The service agreement must reference that premise as its characteristic premise. Refer to Application of De Minimis for more information.

Bill Factors for UK VAT and CCL

Copyright © 2026, Oracle and/or its affiliates.

- Standard Rate VAT
- Reduced Rate VAT
- VAT Declaration Percentage Threshold. Set up a separate bill factor for each service type as the thresholds differ based on service type.
- De Minimis Average Daily Amount Threshold. Set up a separate bill factor for each service type as the de minimis amounts differ based on service type.
- Climate Change Levy (CCL) Per Unit Price. Set up a separate bill factor for each service type as the CCL per unit prices differ based on service type.

Characteristic Type for De Minimis Amounts

Set up a characteristic type to use for the bill characteristic that indicates when de minimis applies. Define the type of characteristic value as a Foreign Key Value and specify a foreign key reference that points to the premise table. Specify this characteristic type as parameters to the algorithms that apply the de minimis rule and that apply VAT.

You need to define a characteristic type for every service type that is subject to the de minimis rule, as the de minimis limit may apply for one service type but not another on the same bill.

> **Note**
>
> A **De Minimis Premise** Characteristic Type is provided that may be used for a specific service type. For additional service types, additional characteristic types will need to be defined.

Distribution Code

Define the following distribution codes:

- Unbilled Prepaid Energy Related Charges
- Unbilled Prepaid VAT

These distribution codes are required as parameters to the algorithms to create the GL details for excess credit.

UOM / TOU / SQI

If using calculation rule-based rates, the calculation rule that calculates CCL requires the consumption that is subject to standard rate VAT as a service quantity.

Define the UOM / TOU / SQI used to store these service quantities.

Service Quantities

If using calculation rules-based rates, the algorithm that checks if the de minimis applies, requires a service quantity containing the total consumption. Unless you have registers that measure the total consumption independent of time of use or interval, you will need to set up a

Copyright © 2026, Oracle and/or its affiliates.

pre-processing calculation group with a calculation rule to aggregate the usage into one service quantity.

Algorithms for UK VAT and CCL

Add the following algorithms:

- Apply De Minimis Rule. Define an algorithm for each service type for which de minimis should be checked. Plug the algorithm(s) in on the customer class for the Pre Bill Completion system event. Refer to the CPBC-DMCH algorithm type for more information about the base package algorithm.
- Create Excess Credit GL Details. Plug this algorithm in on the customer class for the FT Freeze system event. Refer to the CFTZ-VAT-GL algorithm type for more information about the base package algorithm.
- Highlight Effective Declarations for Account and Premise. Plug this algorithm in on the installation option for the Control Central Alert system event. Refer to the CCAL-DECL algorithm type for more information about the base package algorithm.

Calculation Rules to Charge VAT and CCL

Four additional calculation rules are required to charge for VAT and CCL on non-domestic rates:

- Calculate standard rate VAT
  - Calculation Rule Type= Apply VAT to Cross Referenced Charges
  - VAT Calculation Type = Standard Rate
  - Specify the De Minimis Characteristic Type
  - Specify Declaration Percentage Threshold Bill Factor
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Note that the calculation rule overrides description on bill
  - Cross-reference the calculation line category for energy related charges that are subject to VAT
- Calculate reduced rate VAT
  - Calculation Rule Type = Apply VAT to Cross Referenced Charges
  - VAT Calculation Type = Reduced Rate
  - Specify the De Minimis Characteristic Type
  - Specify Declaration Percentage Threshold Bill Factor
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Note that the calculation rule overrides description on bill
  - Cross-reference the calculation line category for energy related charges that are subject to VAT
- Calculate CCL and CCL Relief
  - Calculation Rule Type = Calculate Climate Change Levy and Relief Charges

Copyright © 2026, Oracle and/or its affiliates.

  - UOM/TOU/SQI = the identifier of the service quantity that the prices applies to
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Note that the calculation rule overrides description on bill
- VAT on CCL
  - Calculation Rule Type = Apply To
  - Value type = Percentage
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Cross-reference the calculation line category for the total CCL and CCL Relief related charges

Setting up the calculation rules as above produces separate lines for VAT on energy related charges and on CCL (i.e., lines will be created for standard rate VAT on energy related charges, reduced rate VAT on energy related charges, and standard rate VAT on CCL charges).

You can set up rates to calculate standard rate VAT on energy related charges and CCL together as follows:

- Calculate standard rate VAT charges
  - Calculation Rule Type = Apply VAT to Cross Referenced Charges
  - FCPO = Yes
  - VAT Calculation Type = Standard Rate
  - Specify the De Minimis Characteristic Type
  - Specify Declaration Percentage Threshold Bill Factor
  - Value Source = Value
  - Value = 100
  - Note that the calculation rule overrides description on bill
  - Cross-reference the calculation line category for energy related charges that are subject to VAT
- Calculate CCL and CCL Relief
  - Calculation Rule Type = Calculate Climate Change Levy and Relief Charges
  - UOM/TOU/SQI = the identifier of the service quantity that the prices applies to
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Note that the calculation rule overrides description on bill
- Calculate standard rate VAT
  - Calculation Rule Type = Apply To
  - Value type = Percentage
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Cross-reference the calculation line category for the total CCL and CCL Relief related charges

Copyright © 2026, Oracle and/or its affiliates.

- Calculate reduced rate VAT:
  - Calculation Rule Type = Apply VAT to Cross Referenced Charges
  - VAT Calculation Type = Reduced Rate
  - Specify the De Minimis Characteristic Type
  - Specify Declaration Percentage Threshold Bill Factor
  - Specify Value Source. It is recommended that you use a value source of Bill Factor and reference the bill factor you set up for standard rate VAT.
  - Note that the calculation rule overrides description on bill
  - Cross-reference the calculation line category for energy related charges that are subject to VAT

Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of consumption. If the customer accepts a quote, one or more real service agreements are created.

NOT_SUPPORTED

Refer to Defining Quotation Options for how to setup the system to enable quotation functionality.

Note

Separate module. Please note that the quoting functionality is part of the Contract Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

The topics in this section describe the quotation functionality.

- The Big Picture Of Quotations
- Maintaining Quotes

The Big Picture Of Quotations

The topics in this section provide background information about a variety of quotation features and functions.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> If you are not familiar with the concepts described under Sales and Marketing, Billing, Rates the quotation functionality will be difficult to understand.

- Basic Quotation Concepts
- Printing Quotes

Basic Quotation Concepts

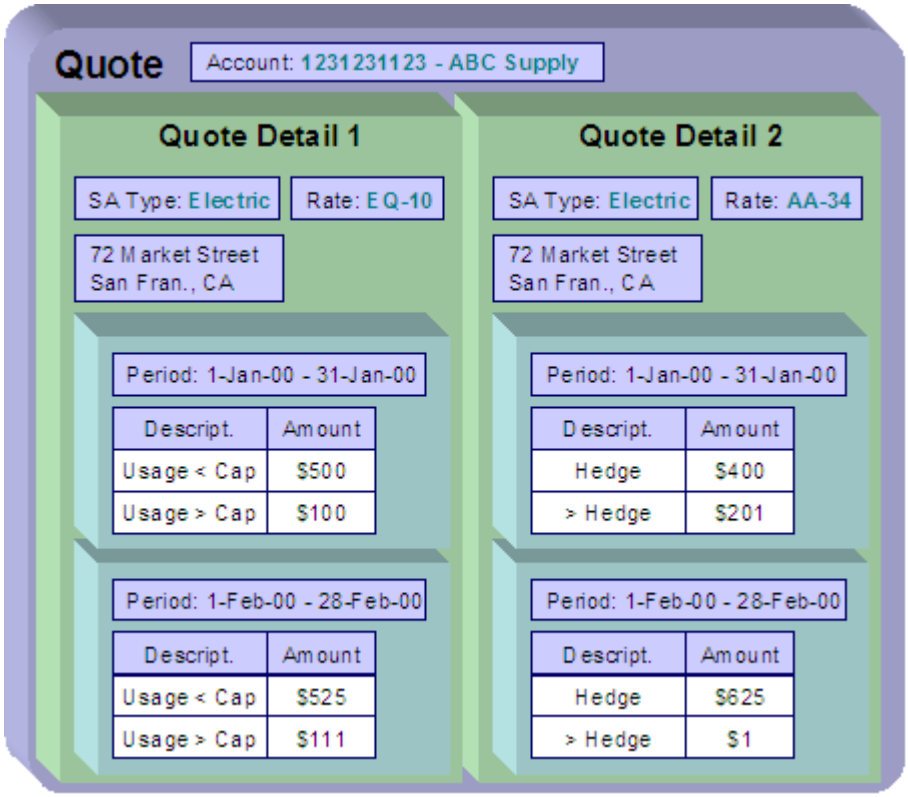
The section describes basic quotation concepts.

- A Quote Shows How Bill Segments Might Look
- Proposal Service Agreements Must Exist
- Creating Quotes And Quote Details
- Completing A Quote Creates Quote Routings (And More)
- Accepting / Declining Quote Details
- Tinkering With Proposal SAs After Quote Generation
- Closing Quotes
- Alerts
- Trees

A Quote Shows How Bill Segments Might Look

A quote shows how a customer's bill segments will look if a given set of pricing terms is applied to a given amount of consumption. The following example illustrates how a quote might look that provides a customer with two different options for their electric service. If the customer elects either option, a new service agreement would be created.

Copyright © 2026, Oracle and/or its affiliates.



Please note the following about a quote:

- A quote is related to an account.
- A quote can contain many quote details. A quote detail shows how bill segments will look if a given set of pricing terms (e.g., rates, riders, contract quantities, contract options, etc.) is applied to a given amount of consumption.
- A quote detail may contain many simulated bill segments. For example, a quote detail could show the customer what their monthly bills would be if a given set of pricing terms were used over the course of several months (where each month has different consumption levels and different prices).
- A quote may contain multiple quote details for the same service. For example, a quote can contain four different pricing options for the same service (where each option is shown using a separate quote detail). The customer can then compare them and choose the optimum option.
- A quote may contain quote details for multiple premises and multiple services at each premise. For example, a quote can contain quote details for electric and gas service at multiple premises.

Proposal Service Agreements Must Exist

You cannot create a quote unless its account has at least one proposal service agreement (SA). A proposal SA contains the pricing terms, legal terms, and template consumption used to generate the simulated bill segments that appear on a quote. It might be helpful to view a proposal SA as special type of service agreement that is ignored when a bill is produced for an account.

The topics in this section provide detailed information about proposal SAs.

Copyright © 2026, Oracle and/or its affiliates.

- Proposal SAs Contain Pricing Terms
- Legal Terms and Conditions May Be Specified On Both Proposal And Real SAs
- Proposal SAs Contain Billing Scenarios And Template Consumption
- Proposal SAs Differ From Real SAs
- Proposal SA State Transition
- Orders Are Used To Create Proposal SAs
- Proposal SAs May Be Grouped Under An Umbrella Agreement

Proposal SAs Contain Pricing Terms

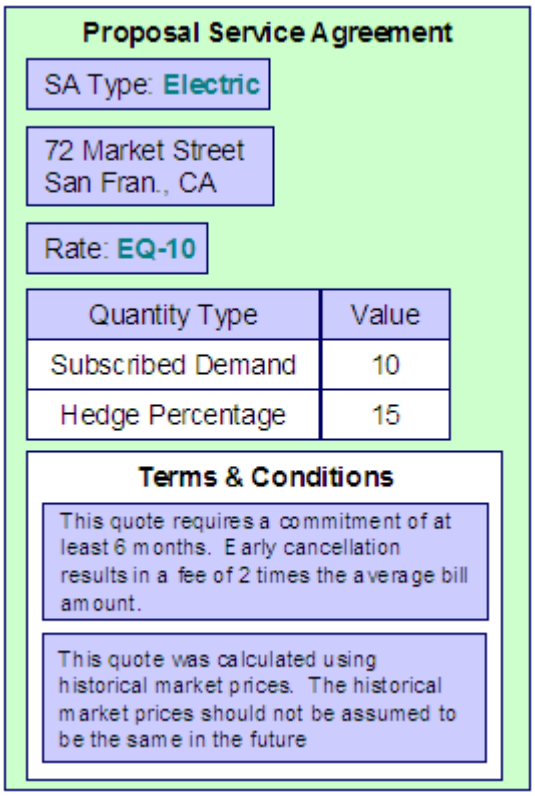
The pricing terms (e.g., rates, contract quantities, riders, bill factor values, etc.) used to generate a quote detail's simulated bill segments reside on a special type of service agreement (SA) referred to as a "proposal SA". Proposal SAs are like real SAs in many respects. In fact, you can view and change a proposal SAs pricing terms using the same transaction used to maintain real service agreements (i.e., service agreement maintenance).

> **Note**
>
> **Quote details are rather empty.** A quote detail doesn't contain the pricing terms or template consumption. Rather, a quote detail references a proposal SA and the proposal SA contains the pricing terms and template consumption.

Legal Terms and Conditions May Be Specified On Both Proposal And Real SAs

In addition to containing pricing terms, proposal SAs also contain the legal terms and conditions (T&Cs) that govern the proposal.

Copyright © 2026, Oracle and/or its affiliates.



Your master list of terms and conditions are defined when you setup your control tables. The system defaults a service agreement's T&Cs from the start option used to initially create it. You can change a SA's T&Cs at will.

> **Note**
>
> **Real SAs and proposal SAs can reference T&Cs.** Both proposal and real SAs can reference T&Cs. The T&Cs on a proposal SA are those that will apply if the customer accepts the related quote detail. The T&Cs on a real SA are those that apply while the SA is active.

Proposal SAs Contain Billing Scenarios And Template Consumption

In addition to the pricing terms and the legal terms and conditions, a proposal SA also contains billing scenarios. A billing scenario defines:

- The number of simulated bill segments that will be generated when a quote detail is produced for the proposal SA.
- The bill period of each simulated bill segment.
- The amount of consumption to be priced under each simulated bill segment. Note this statement is only true for non-interval consumption. *Note: Proposal SAs that use interval consumption are not supported in C2M. Load curves and TOU mapping are handled in Oracle Utilities Meter Data Management.*

A proposal SA's billing scenarios can be created as follows:

Copyright © 2026, Oracle and/or its affiliates.

- A plug-in spot exists on SA type called Proposal SA Creation. Algorithms plugged into this spot can setup billing scenarios. Refer to Enabling The Automatic Generation Of Billing Scenarios for more information about how to do this.
- A user can add / change / delete billing scenarios on a proposal SA using Service Agreement - Billing Scenario.

Proposal SAs Differ From Real SAs

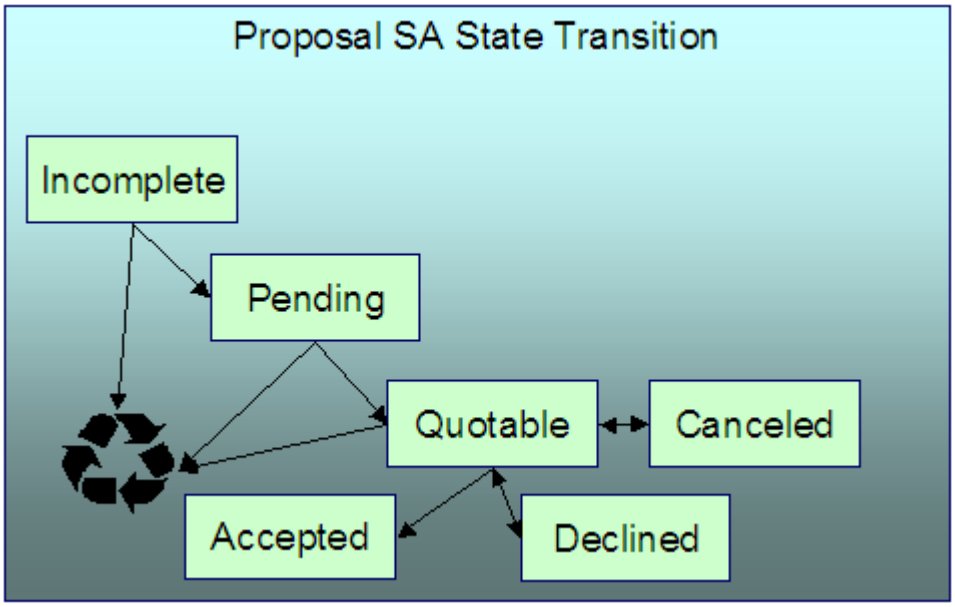
A proposal SA is just like any other SA in that it contains the pricing terms that are used to generate how much the customer owes for service. In fact, the service agreement maintenance transaction is used to maintain both types of service agreements (however, you cannot use this transaction to create a proposal SA; proposal SAs can only be created using the order transaction).

However, there are differences between proposal SAs and real SAs:

- Billing ignores proposal SAs.
- Proposal SAs cannot have financial transactions.
- Proposal SAs are not subject to credit and collections.
- Proposal SAs do not appear on lists of "active" SAs in the system's various zones, grids and trees.
- Fieldwork is not performed for proposal SAs.
- A proposal SA's state transition is very different from a real SA. Refer to The Lifecycle Of A Service Agreement for an illustration of a real SAs state transition. Refer to Proposal SA State Transition for an illustration of a proposal SAs state transition.

Proposal SA State Transition

The following diagram shows the possible lifecycle of a proposal SA.



Copyright © 2026, Oracle and/or its affiliates.

You should never see an Incomplete proposal SA as this is a transitory state used when a proposal is first initiated using the Order transaction. The system always transitions such proposal SAs to the Pending state. Please note that when a proposal SA becomes Pending, the system executes any Proposal SA Creation algorithms plugged in on the proposal SA's SA type. You might use such algorithms to generate billing scenarios or initiate a workflow process that requests template consumption from the prospect.

A proposal SA exists in the Pending state while you are working on the proposal. For example, you might want to leave the proposal SA in this state while you wait for the prospect to send you their historical consumption.

A proposal SA is Quotable when a quote detail can be generated for it (at least one billing scenario must be linked to the proposal SA before it can become Quotable). Note, proposal SAs can be modified when they are Quotable.

> **Note**
>
> **Automatically transitioning a proposal SA to the Quotable state when it's first created.** As described earlier, you can setup a proposal SA creation algorithm to automatically create billing scenarios when a proposal SA is created. The base package algorithm can also automatically transition the proposal SA to the Quotable state (thus making it ready for quotation). Refer to Enabling The Automatic Generation Of Billing Scenarios for more information.

A proposal SA becomes Accepted when a customer accepts one of its quote details.

A proposal SA becomes Declined when a customer declines one of its quote details (or its quote expires). A Declined proposal SA can be "reopened" (by making it Quotable). You may need to do this if you want to change a proposal SA's terms in preparation of generating a new quote detail.

A proposal SA becomes Canceled when a quote detail is withdrawn (or its quote expires). You may need to do this if you want to change a proposal SA's terms in preparation of generating a new quote detail.

Incomplete, Pending and Quotable proposal SAs may be deleted.

Orders Are Used To Create Proposal SAs

Proposal SAs can only be created using the order transaction. This transaction allows a user to pose eligibility-oriented questions to a customer or prospect. The system determines which packages of goods and services for which they are eligible based on the answers to the questions. If the customer is interested in receiving a package, one or more SAs (real and/or proposal) are created using the SAs To Create information defined on the selected package.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You must setup campaigns and packages before you can create an order.** If you are not familiar with the order transaction, you should take the time to form an intuitive understanding of the concepts described under Sales & Marketing. With this knowledge in place, you can design and set up the campaigns and packages necessary to allow an order to create proposal SAs. Keep in mind that a single switch controls whether real SAs or a proposal SAs are created when a package is selected on the order transaction. This switch resides on Package - SAs To Create.
>
> **A package could create a combination of real and proposal SAs.** When a customer elects to take a package of goods and services (via the order transaction), the system creates a service agreement for each SA type / start options defined on the package's SAs To Create. It is possible to setup a package to create both real and proposal SAs. For example, the customer may be calling you to start electric service, but they may also be interested in seeing a proposal for gas service.
>
> **You don't create special SA types for proposal SAs.** The SA type on a proposal SA is the same as the SA type on the real SA that's created when the proposal SA is accepted. This is because the business rules on your SA types govern both real and proposal SAs.
>
> **You don't need special start options for proposal SAs.** The start options that you setup to default information onto new, real service agreements are also referenced on the SA To Create information used to create proposal SAs.

Proposal SAs May Be Grouped Under An Umbrella Agreement

Each proposal SA may be part of a larger contract for multiple services that you are negotiating with a customer or group of customers. You may use an umbrella agreement and its collection of terms of service records to group together multiple proposal service agreements.

Creating Quotes And Quote Details

Once an account has quotable proposal SAs, you can generate a quote for the account using the quote maintenance transaction. When you generate a quote for an account, you can include quote details for any / all of the account's quotable proposal SAs.

The logic used to create a quote detail's simulated bill segments is not hard-coded. Rather, the system calls the algorithm plugged in on the proposal SA's SA type in the Proposal SA - Bill Segment Generation plug-in spot. Refer to Enabling The Generation Of Simulated Bill Segments for information about this plug-in spot.

The major difference between a bill segment linked to a quote detail and one linked to a bill is that a quote detail's bill segments are calculated using simulated consumption. This means that the financial information on quote details is not a guess - it's what the bill would be if a given amount of consumption was priced using the pricing terms (as long as you opt to use rate application to calculate the simulated bill segments).

In addition to this major difference, there are a few minor differences between a real bill segment and a simulated bill segment:

Copyright © 2026, Oracle and/or its affiliates.

- A quote detail bill segment does not have a foreign key to a bill. Rather, it references a quote detail.
- A quote detail bill segment's prime key is inherited from its quote detail (rather than its bill).
- A quote detail bill segment can never have a financial transaction linked to it.
- A quote detail bill segment doesn't have read details.
- A quote detail bill segment's state transition is simpler (it's either in the Error state or OK).

Completing A Quote Creates Quote Routings (And More)

When a quote's quote details are all OK, you change the status of the quote to complete to indicate the quote is ready to be sent to the customer or prospect.

When a quote is completed, the system creates a quote routing for each person who receives a copy of the account's quotes. When you indicate a person should receive a copy of the account's quotes, you define the routing method (e.g., email, fax, post) and the appropriate address.

> **Note**
>
> **Recipients of quotes.** You define who receives a copy of an account's quotes on the Account - Persons page. The system assumes (by default) that the main person linked to the account should receive the account's quotes. You can subsequently indicate any person linked to the account should receive its quotes (and turn off the creation of quotes for the main person). In addition, you can manually add additional quote routings to a quote if you need to send the quote to individuals not currently linked to the account. Refer to How To Reprint A Quote for more information.

The information that appears on the quote routing record is controlled by the quote route type specified on the respective account / person information:

- If the quote route type's routing method is email or fax and the system has been configured to use notification preferences for quote routing, a quote routing is created for each active contact preference whose notification type is associated with the **quote route type** specified on the respective account / person information. Refer to Bill and Quote Routing in *Designing Notification Types* for more information. Otherwise, quote routings are created as follows:
  - If the quote route type indicates the person's quotes are routed via fax, the person's fax number is used as the routing information. The system knows which of a person's person contacts is a fax number is by the contact routing or which of a person's phone numbers is a fax number by the phone type.
  - If the quote route type indicates the person's quotes are routed via email, the person's Email address is used as the routing information. The system knows which of a person's person contacts is an email or knows where to find the person email.
- If the quote route type indicates the person's quotes are routed via the postal service, the address constituents are populated with the address specified on the **bill** address source that is also specified on the account / person information. Note, if the person has a seasonal address effective on the business date, the seasonal address will be used regardless of the value of **bill address source**.

Copyright © 2026, Oracle and/or its affiliates.

Regardless of whether the quote is routed via fax, email or post; the routing record is also stamped with the batch process and run number that performs the download of the quote information. The batch process is defined on the quote route type. Refer to Printing Quotes for more information about this batch process.

In addition to creating routing information at completion time, the system executes any quote completion algorithms plugged in on the account's customer class. For example, you may want to create a workflow process to manage the processing of the quote. To do this, you plug in a Quote Completion Algorithm that creates a workflow process on the respective customer class .

Accepting / Declining Quote Details

A customer can accept or decline each quote detail on a quote.

When a quote detail is accepted, the system:

- Changes the state of the related proposal SA to accepted.
- Executes the Proposal SA - Acceptance algorithms plugged in on the proposal SA's SA type. The base package algorithm creates a real SA by copying the information from the proposal SA. The real SA is created in the pending start state. This new SA behaves similarly to a pending start SA that's created via the Order or Start/Stop transactions (e.g., field activities are initiated if the SA type has been setup accordingly).

When a quote detail is declined, the user must supply a decline reason. The system then changes the state of the related proposal SA to declined.

NOT_SUPPORTED

Refer to Proposal SA State Transition for more information about a proposal SA's states.

Note

Reopening a declined proposal SA. If a customer changes their mind about a declined proposal SA, you can return it to the quotable state using the service agreement maintenance transaction.

Rejecting an entire quote. If all details on a quote are being declined, you can simply change the state of the quote to closed rather than reject each quote detail. Refer to Closing Quotes for more information.

Tinkering With Proposal SAs After Quote Generation

The system allows a quotable proposal SA to be changed, EVEN IF QUOTE DETAILS HAVE BEEN GENERATED. This is intentional as otherwise you would have to create an entirely new proposal SA if you wanted to send another quote for slightly different terms. Your technical team could always introduce a user exit to prevent this from happening if you really didn't want this to take place.

Copyright © 2026, Oracle and/or its affiliates.

Because the system allows this, a potential issue is that the customer may end up accepting a proposal SA that differs from the terms on their quote (e.g., you could change the SA after the quote detail is generated). For this reason, we snapshot the version number of the proposal SA on the quote detail at the time it's generated. If a proposal SA is accepted and its version number differs from the value on its quote details, a warning is produced.

A warning is also issued if you change a proposal SA and a quote detail exists (reminding you to generate a new quote / quote detail).

In addition to the above, it's also important to understand that the system intentionally allows multiple quotes to reference a given proposal SA. This allows you to leave multiple quotes open that reference the same proposal SA(s).

Closing Quotes

Every quote has an expiration date. The CLOSEQTE background process closes quotes on their expiration date. This process changes the state of all related proposal SAs to canceled or declined (an input parameter of this process controls which state is used).

You can also close a quote real time using the quote maintenance transaction. When you do this, the system asks if the related proposal SAs should be left as quotable or if they should be changed to canceled or declined.

> **Note**
>
> **Reopening a declined proposal SA.** If a customer changes their mind about a declined proposal SA, you can return it to the quotable state using the service agreement maintenance transaction.

Alerts

The alert zone highlights if:

- The selected account has pending or complete quotes. Refer to Quote Lifecycle for more information.
- The selected account has pending or quotable proposal SAs. Refer to Proposal SA State Transition for more information.

Trees

The account tree highlights:

- The existence of non-closed quotes
- The existence of proposal SAs and their related quote details

Printing Quotes

Copyright © 2026, Oracle and/or its affiliates.

The contents of this section describe the technical implementation of printing quotes.

- Quote Routings Are Created For Each Recipient Of A Quote
- Quote Route Types Control The Information Merged Onto Quotes
- Technical Implementation Of Online Quote Display
- Technical Implementation Of Printing Quotes In Batch
- Reproducing The Quote Print Flat File
- How To Reprint A Quote (For The Original Recipients or For Someone New)
- The Quote Extract Records

Quote Routings Are Created For Each Recipient Of A Quote

Refer to Completing A Quote Creates Quote Routings for a description of how the system constructs the information used to route a quote to one or more recipients.

Quote Route Types Control The Information Merged Onto Quotes

Every quote routing references a quote route type.

The quote route type controls the following quote functions:

- It contains an algorithm that is responsible for extracting the information merged onto your quotes. Specifically, algorithms of this type create the "flat file records" that are passed to your quote-print software. Algorithms of this type are called under the following scenarios:
  - The background process that builds the flat file that's passed to your quote-print software calls these algorithms to construct the "flat file records" for each quote. Refer to Technical Implementation of Printing Quotes In Batch for the details.
  - If your quote-print software has the ability to construct a real-time image of a quote (in a PDF), you can plug in an Online Quote Display algorithm on the Installation Record. This type of algorithm calls the quote's dispatch group extract algorithm to extract the information that is merged onto the quote. Refer to Technical Implementation Of Online Quote Display for the details.
- Refer to The Quote Extract Records for examples of the record layouts passed to the document composition application.
- It contains the ID of the background process that builds the flat file that's passed to your quote-print software. The base package example of this background process (QUOTROUT) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file.

> **Note**
>
> CIS Division can override the extract algorithm and/or the background process found on the quote route type. The system processes first check the quote's account's CIS Division for an override specific for the quote route type.

Copyright © 2026, Oracle and/or its affiliates.

Technical Implementation Of Online Quote Display

Users can view an image of any quote if you set up the following:

- Plug in an Online Quote Display construction algorithm on the Installation Record. Refer to ONQD-DOC1 for an example of such an algorithm. Note if your quote-print software is not capable of producing a PDF containing an image of a quote, users will not be able to view images of quotes.
- Plug in the appropriate extract algorithm on each quote route type . Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to QTEX-EX for an example of such an algorithm.

When you plug in these algorithms, a button appears on Quote - Main. When a user clicks this button, the following takes place:

- The installation record's Online Quote Display algorithm is executed.
- This algorithm calls the quote's route type's extract algorithm. This algorithm constructs the information that's merged onto the quote and returns it to the Online Quote Display algorithm. This algorithm, in turn, passes it to the quote-print software.
- The quote-print software renders an image of the quote in a PDF and returns it to the Online Quote Display algorithm.
- And finally, the Online Quote Display algorithm displays the PDF in a browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

Technical Implementation Of Printing Quotes In Batch

The batch process that extracts quote information reads all quote routing records in a given run number that are marked with its batch control ID. For each quote routing record, it creates numerous records on a flat file. These records contain the information that is merged onto your quotes.

The base package example of this process (QUOTROUT) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file. Refer to Quote Route Types Control The Information On Quotes for more information.

> **Note**
>
> **If your software doesn't support online quote display.** The algorithm that formats quote extract records that's plugged in on quote route type serves two purposes: 1) it formats the records used to construct online images of a quote, and 2) it formats the records downloaded to your quote print software in batch. If your quote printing software does not support the rendering of quote images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that both formats the extract records and downloads them (and then specify this batch process on your quote route type).

Copyright © 2026, Oracle and/or its affiliates.

Reproducing The Quote Print Flat File

You can reproduce the flat file at any time. Simply request the quote production background process and specify the run number associated with the historic run.

How To Reprint A Quote (For The Original Recipients or For Someone New)

A copy of a quote is sent to every individual / business specified in the quote's routing details. The system creates the routing details when a quote is completed. Refer to Completing A Quote Creates Quote Routings for more information about how this information is compiled during quote completion.

If you want to reprint a quote OR if you want to send a quote to someone other than the original recipients:

- Navigate to the Quote - Main page.
- Transfer to the Quote - Quote Routings tab.
- Insert a new row (click the + button). Note that you can only insert information on a quote that is Complete.
- Specify the name and address to which the quote should be sent. If you want to send the quote to someone other than the original recipient, specify the individual's name and address.
- Save the quote.

If a quote is Complete but the print process hasn't yet executed, you may change the name and address on the original routing details; you don't have to insert a new routing row.

The Quote Extract Records

Numerous different types of records are interfaced to the document composition application. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the document composition application quote template.
- The next X bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed quote appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted quotes are printed.
- The remaining bytes contain the **Quote Information** that appears on the printed quote. The type of information differs for each type of record.

The topics in this section describe each component.

- Printer Key (Record Types)
- Sort Key
- Mailing Key
- Quote Information

Copyright © 2026, Oracle and/or its affiliates.

Printer Key (Record Types)

The following table defines each printer key (i.e., record type) on the quote print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file
0100	Quote record	Required. 1 per Quote.
0200	Quote Detail record	1 per quote detail.
0300	Terms and Conditions record	1 per terms and conditions linked to the proposal SA.
0400	Bill segment record	1 per bill segment linked to the quote detail.
0500	Bill calculation header	1 per bill calculation header linked to the bill segment.
0600	Bill calculation line	1 per bill calculation line linked to the bill calculation header.
0700	End of record type 0500	1 for each bill calculation header
0800	Read detail	1 for each meter read detail linked to the bill segment.
0900	SQ detail	1 for each service quantity detail linked to the bill segment.
1000	End of record type 0400	1 for each bill segment.
1100	End of record type 0200	1 for each quote detail.
9999	End of record type 0100	1 for each quote.

Sort Key

The following table defines the structure of the sort key that appears on each quote print record. Please note that different components are filled in for each printer key.

Field Name	Format	Source/Value/Description
Quote ID	A12	
Sequence Number	N2	The sequence number of the account / person information associated with this quote routing information.
Quote record group	A2	10 Quote 20 Quote Detail 30 Quote end
Quote Detail ID	A12	

Copyright © 2026, Oracle and/or its affiliates.

Quote Detail record group	A2	10 Quote Detail record 20 SA Terms and Conditions 30 Bill Segment 40 Quote Detail record end
SA T&C Short Description	A60	
Bill Segment End Date	A10	
Bill Segment Id	A12	
Bill Segment Group	A2	10 Bill segment 20 Bill segment calculation group 30 Bill segment read detail 40 Bill segment SQ detail 50 Bill segment end
Bill Calc Header Sequence	N2	
Bill Calc Header Group	A2	10 Bill calculation header 20 Bill calculation line 30 Bill calculation end
Line Seq	N4	Defines order of rows for similar record types

Mailing Key

The following table defines the structure of the mailing key that appears on each quote print record.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on the quote routing record.

Quote Information

The topics in this section describe the information that appears on each on each quote print record. The following links are provided so that you can move easily to specific topic within this section:

- Global Extract Information (0010) Record Definition
- Quote (0100) Record Definition
- Quote Detail (0200) Record Definition
- SA Terms and Conditions (0300) Record Definition
- Bill Segment (0400) Record Definition
- Bill Calculation Header (0500) Record Definition
- Bill Calculation Line (0600) Record Definition
- End of Bill Calculation Header (0700) Record Definition

Copyright © 2026, Oracle and/or its affiliates.

- Bill Segment Read (0800) Record Definition
- Bill Segment SQ (0900) Record Definition
- End of Bill Segment (1000) Record Definition
- End of Quote Detail (1100) Record Definition
- End of Quote (9999) Record Definition
- Address Sub-Record Definition

Global Extract Information (0010) Record Definition

This is a "header" record that contains information about the extract run as a whole.

> **Note**
>
> Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file.

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
QTE_CNT	N10	Count of quotes on this extract.
EXTRACT_DTTM	A26	System time of extraction.
QUOTE_ID_START	A12	This is only used when a quote is produced as a result of an online request (from Quote - Main). It contains the Quote Id to extract.
QUOTE_ID_END	A12	This is only used when a quote is produced as a result of an online request (from Quote - Main). It contains the Quote Id to extract.
THREAD_NBR	N10	This field is only used if quotes are extracted in batch. It contains the thread number in which the quote was extracted.
THREAD_CNT	N10	This field is only used if quotes are extracted in batch. It contains the total number of threads that were extracted.

Top of Page

Quote (0100) Record Definition

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
QUOTE_ID	A12	CI_QUOTE
ACCT_ID	A10	CI_QUOTE
EXPIRE_DTTM	A26	CI_QUOTE
QUOTE_STAT_FLG	A2	CI_QUOTE
CURRENCY_CD	A3	CI_ACCT
CUR_SYMBOL	A4	CI_CURRENCY_CD
ENTITY_NAME1	A64	CI_QUOTE_RTG. Mailing Entity Name1.
ENTITY_NAME2	A64	CI_QUOTE_RTG. Mailing Entity Name2.
ENTITY_NAME3	A64	CI_QUOTE_RTG. Mailing Entity Name3.
ADDRESS_SBR	A361	CI_QUOTE_RTG. Refer to its components in a later section.
EMAILID	A254	CI_QUOTE_RTG
PHONE-FAX	A64	CI_QUOTE_RTG

Top of Page

Quote Detail (0200) Record Definition

Field Name	Format	Source/Value/Description
QUOTE_DTL_ID	A12	CI_QUOTE_DTL
SA_ID	A10	CI_QUOTE_DTL
TOTAL_AMT	N13.2	CI_BSEG_CALC. Sum of all calculation amounts (quote detail's bill segments).
PREMISE_ID	A10	CI_SA (Characteristic Premise)
ADDRESS_SBR	A361	CI_PREM. Refer to its components in a later section.
SA_TYPE_DESCR	A60	CI_SA_TYPE_L
SVC_TYPE_CD	A30	CI_SVC_TYPE_L
SA_VERSION	N5	CI_QUOTE_DTL
QTE_DTL_SA_VERSION	N5	CI_QUOTE_DTL
RS_DESCR	A60	CI_RS_L. Description of Rate effective on SA start date.

Top of Page

SA Terms and Conditions (0300) Record Definition

Field Name	Format	Source/Value/Description
START_DT	A10	CI_SA_TC
END_DT	A10	CI_SA_TC
TC_TYPE_CD	A12	CI_TC

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
DESCR	A60	CI_TC_L
TC_TYPE_TXT	A2000	CI_TC_L

Top of Page

Bill Segment (0400) Record Definition

Field Name	Format	Source/Value/Description
START_DT	A10	CI_BSEG
END_DT	A10	CI_BSEG
BSEG_STAT_FLG	A2	CI_BSEG
BSEG_AMT	S13.2	CI_BSEG_CALC_LN

Top of Page

Bill Calculation Header (0500) Record Definition

Field Name	Format	Source/Value/Description
START_DT	A10	CI_BSEG_CALC
END_DT	A10	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
CALC_AMT	S13.2	CI_BSEG_CALC
DESCR_ON_BILL	A254	CI_BSEG_CALC

Top of Page

Bill Calculation Line (0600) Record Definition

Field Name	Format	Source/Value/Description
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_AMT	S13.2	CI_BSEG_CALC_LN
EXEMPT_AMT	S13.2	CI_BSEG_CALC_LN
UOM_CD	A30	CI_BSEG_CALC_LN
TOU_CD	A30	CI_BSEG_CALC_LN
SQI_CD	A30	CI_BSEG_CALC_LN
BILL_SQ	S12.6	CI_BSEG_CALC_LN
DESCR_ON_BILL	A254	CI_BSEG_CALC_LN

Top of Page

End of Bill Calculation Header (0700) Record Definition

Field Name	Format	Source/Value/Description
Dummy field		

Copyright © 2026, Oracle and/or its affiliates.

Top of Page

Bill Segment Read (0800) Record Definition

> **Note**
>
> The Bill Segment Read (0800) Record is not applicable for Customer to Meter (C2M). A quote detail bill segment doesn't have read details.

Top of Page

Bill Segment SQ (0900) Record Definition

Field Name	Format	Source/Value/Description
UOM_CD	A30	CI_BSEG_SQ
TOU_CD	A30	CI_BSEG_SQ
SQI_CD	A30	CI_BSEG_SQ
SQ_DEC_POS	N1	CI_BSEG_SQ
BILL_SQ	S12.6	CI_BSEG_SQ

Top of Page

End of Bill Segment (1000) Record Definition

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End of Quote Detail (1100) Record Definition

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

End of Quote (9999) Record Definition

Field Name	Format	Source/Value/Description
Dummy field		

Top of Page

Address Sub-Record Definition

The address that appears in both the quote record (0100) and the quote detail record (0200) is composed of the following:

Field Name	Format	Description
COUNTRY	A3	

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Format	Description
ADDRESS1	A64	Refer to Completing A Quote Creates Quote Routings for a description of conditions when this field will contain a fax number or an Email address rather than the first line of a postal address (this only occurs for a bill record (0100)).
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Top of Page

Maintaining Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of consumption. If the customer accepts the quote, real service agreements are created.

NOT_SUPPORTED

Refer to The Big Picture Of Quotations for background information about quotes.

The topics in this section describe how to maintain quotes.

- The Lifecycle Of A Quote And Quote Detail
- Quote - Main

The Lifecycle Of A Quote And Quote Detail

The topics in this section describe the lifecycle of the various quote objects.

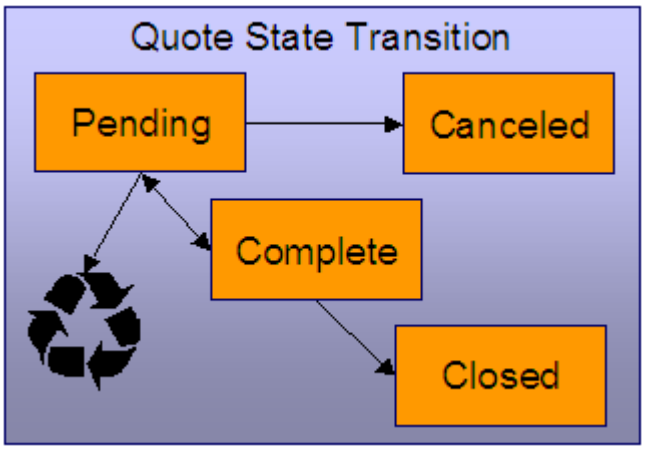
- Quote Lifecycle

Copyright © 2026, Oracle and/or its affiliates.

- Quote Detail Lifecycle

Quote Lifecycle

The following diagram shows the possible lifecycle of a quote.



Pending A quote starts its life in the Pending state. Refer to Creating Quotes And Quote Details for more information.

A quote remains Pending until you Cancel or Complete it.

You may delete a Pending quote. Deleting a pending quote also deletes its quote details.

Canceled You cancel a quote if you want a record of a Pending quote that was never sent to a customer.

Complete A quote becomes Complete when it is ready to be sent to a customer. No changes may be made to a Complete quote. Refer to Completing A Quote for a description of what happens when a quote is completed.

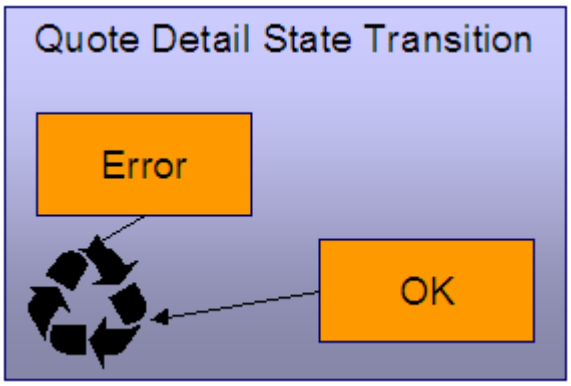
A Complete quote may be reopened. You would only do this if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote details, and then recomplete the quote.

Closed A Complete quote becomes Closed when its terms are no longer valid. Refer to Closing Quotes for more information.

Quote Detail Lifecycle

The following diagram shows the possible lifecycle of a quote detail.

Copyright © 2026, Oracle and/or its affiliates.



When a quote detail is generated, it is placed in the Error or Ok state depending on whether the system successfully generated simulated bill segments for each billing scenario linked to the related proposal SA. A quote detail may be deleted if its parent quote is Pending. Refer to Creating Quotes And Quote Details for more information.

Quote - Main

This page is used to maintain quotes. This page provides many diverse functions:

- You can generate, complete, close, reopen, and cancel a quote.
- You can regenerate and delete quote details.
- You can accept, decline and cancel the proposal SAs that contribute a quote detail to the quote.

You should not attempt to use this transaction until you are comfortable with the concepts described under The Big Picture of Quotations.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Quote**, and then **Search.**

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

Account ID identifies the customer or prospect associated with the quote. The name of the account's main customer appears adjacent.

Quote Status contains the state of the quote. Refer to Quote Lifecycle for information about the valid values and the state transition rules.

If your system has been configured to display an online image of a quote, a **Display Quote** button appears adjacent to the quote's status. When clicked, an online image of the quote is displayed. Refer to Technical Implementation Of Online Quote Production for how to enable this feature.

Expiration Date / Time identifies when the quote's terms are no longer valid. Refer to Closing Quotes for how the system uses this information to automatically close a quote.

Refer to Quote Actions for a description of each **Quote Action** button.

This transaction has sophisticated logic that can be used to perform mass updates to the quote's quote details and their related proposal SAs. These mass update actions are executed by selecting quoted details (in the grid) and then clicking the desired action button at the

Copyright © 2026, Oracle and/or its affiliates.

bottom the page. The remainder of this section describes how to use this mass update functionality.

The **Select All** / **Clear All** buttons are used to select / unselect the quote details. These buttons are disabled when the quote is Closed.

> **Note**
>
> **100 quote details at a time.** Clicking **Select All** selects the first 100 quote details in the grid. If more than 100 quote details exist, you must select them in batches.

The grid that follows contains the quote details linked to the quote. The following information appears in the grid:

- Select box. Before you can perform any of the mass update actions at the bottom of the page, at least one quote detail must be selected. This field is disabled when the quote is Closed. You can use the **Select All** and **Clear All** buttons to change the value of the switch for all quote details in the grid.
- Click the go to button to transfer to the Quote - Quote Detail tab with the corresponding quote detail positioned in the scroll. On this page, you can see the details of the quote detail's simulated bill segments (and, from there, drill down into the details of each bill segment).
- The **Quote Detail Info** column contains a concatenation of a quote detail's division, SA type, start option, status, start date, and premise information (this information comes from the quote detail's proposal SA).
- The **Total Amount** column contains the sum of the quote detail's simulated bill segments.
- The **Remarks** column highlights special circumstances associated with the quote detail.
  - Proposal SA has been changed since the quote was created appears if the proposal SA was modified after the quote detail was generated. If you'd like to recalculate this quote detail, select this row and click the **Regenerate** button at the bottom of the page to regenerate the quote detail's simulated bill segments.
  - Quote detail contains bill segments in error appears if any of the simulated bill segments linked to the quote detail are in error . Click the adjacent go to button to transfer to the Quote - Quote Detail tab where the state of each simulated bill segment is shown. You can correct simulated bill segments that are in error by **Regenerating** the quote detail (and you do this by selecting the quote detail and clicking the Regenerate button at the bottom of the page).

The **Quote Detail Actions** and **Proposal SA Actions** can be used to perform mass updates to the selected quote details in the grid. Refer to Quote Detail Actions and Proposal SA Actions for the details.

- Action Buttons On Quote Main
- Quote Actions
- Quote Detail Actions
- Proposal SA Actions
- Quote - Quote Details
- Quote - Quote Routings

Copyright © 2026, Oracle and/or its affiliates.

Action Buttons On Quote Main

The topics in this section describe the various action buttons on the Quote - Main page.

Quote Actions

The topics in this section describe the **Quote Action** buttons on the Quote - Main page. The following links are provided so that you can move easily to specific topic within this section:

- Generate
- Complete
- Close
- Reopen
- Cancel (Quote)

Generate

The **Generate** button causes one or more quote details to be created for the account's Quotable proposal SAs. This button is enabled when there is at least one Quotable proposal SA linked to the selected **Account ID**.

When clicked, the **Generate Quote** window opens.

You must specify the following parameters to generate quote details:

- Indicate if quote details should be created for All Proposal SAs or for a Specific Proposal SA.
- If you indicated a quote detail should be generated for a Specific Proposal SA, also specify the **Proposal SA ID**.

After specifying the parameters, click **Generate** to create the quote details for quotable proposal SAs.

> **NOT_SUPPORTED**
>
> Refer to Creating Quotes And Quote Details for more information.

> **Warning**
>
> If the quote has existing quote details for the designated proposal SAs, they are deleted when you generate quote details (i.e., the system regenerates existing quote details). You can also regenerate quote details by selecting them on the Main page and clicking the **Regenerate** button.

Top of Page

Complete

Copyright © 2026, Oracle and/or its affiliates.

The **Complete** button causes the quote to become complete. This button is enabled when the quote is pending and all quote details linked to the quote are OK.

> **NOT_SUPPORTED**
>
> Refer to Completing A Quote Creates Quote Routings for a description of what happens during quote completion.

Top of Page

Close

A quote is closed when its terms are no longer valid (i.e., when the related proposal SAs can no longer be accepted). Most quotes are closed automatically by the system (refer to Closing Quotes for how this takes place).

If you want to close a quote real time, click the **Close** button. When this button is clicked, the system opens the **Close Quote** window and asks if anything should be done with the related proposal SAs.

In the resulting window, you must specify one of the following **Proposal SA Actions**:

- Choose No Action if you don't want to change the status of the related proposal SAs.
- Choose Decline if you want the status of the proposal SAs that have contributed quote details to the quote to be changed to declined . You must also define a **Decline Reason**.
- Choose Cancel if you want the status of the proposal SAs that have contributed quote details to the quote to be changed to canceled .

Top of Page

Reopen

The **Reopen** button causes a quote to return to the pending state. You would only reopen a quote if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote detail, and then recomplete the quote.

This button is enabled if the quote is complete.

Top of Page

Cancel (Quote)

The **Cancel** button causes the quote to become canceled. You would cancel a quote if you want a record of a pending quote that was never sent to a customer. This button is enabled when the quote is pending.

Canceling the quote does NOT cancel the related proposal SAs. If you want to cancel the related proposal SAs, you have the following options:

- You can click the Close button rather than the **Cancel** button. When you close a quote, you are given the option of also canceling the proposal SAs that contributed a quote detail to the quote.
- Before canceling the quote, you can select the quote details and click the Cancel button at the bottom of the page (in the Proposal SA Actions section). When you click this button, the system cancels the proposal SAs linked to the quote details.

Copyright © 2026, Oracle and/or its affiliates.

- You can transfer to the Quote - Quote Details tab and then drill down to the each proposal SA (using the SA context menu button). This transfers you to Service Agreement - Main where you can cancel the proposal SA.

Quote Detail Actions

The topics in this section describe the **Quote Detail Action** buttons on the Quote - Main page. These buttons are used to perform a given action to the quote details that are selected in the quote details grid.

- The **Regenerate** button is used to delete and recreate the selected quote details. This button is enabled if the quote is pending and at least one row in the quote details grid is selected.
- The **Delete** button is used to delete the selected quote details. This button is enabled if the quote is pending and at least one row in the quote details grid is selected.

Proposal SA Actions

The topics in this section describe the various **Proposal SA Action** buttons on the Quote - Main page. These buttons are used to perform a given action to the proposal SAs associated with the selected quote details.

The **Accept** button is used to accept the proposal SA associated with the quote detail. This button is enabled if the quote is pending or complete and at least one row in the quote details grid is selected.

When you accept a proposal SA, you are asked to supply the **Start Date** of the real service agreement that will be created.

Refer to Accepting / Declining Quote Details for more information about how accepting a quote detail causes a real service agreement to be created.

Decline

The **Decline** button is used to decline the proposal SA associated with the quote detail. This button is enabled if the quote is pending or complete and at least one row in the quote details grid is selected. When you decline a proposal SA, you are asked to supply a **Decline Reason**.

Refer to Accepting / Declining Quote Details for more information about declining a quote detail.

Cancel

The **Cancel** button is used to cancel the proposal SA associated with the quote detail. This button is enabled if the quote is pending or complete and at least one row in the quote details grid is selected.

Quote - Quote Details

This page is used to view a summary of the simulated bill segments linked to a quote detail.

Copyright © 2026, Oracle and/or its affiliates.

Open this page using **Menu**, and then **Sales & Marketing**, and then **Quote**, and then **Search** and navigate to the **Quote Details** page to define this information. You can also display a quote detail by clicking the respective go to button in the quote details grid on the main tab.

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Details** scroll contains one entry for every quote detail linked to the quote (a summary of these quote details also appears on the Main tab). The remainder of this section describes the information displayed for each quote detail.

> **Note**
>
> **You don't have to use the scroll.** If the quote has many quote details, you don't have to use the scroll to work through them. Rather, you can use the go to button in the quote details grid on the main tab to position the respective quote detail in the following scroll. Also note, you can use Alt - and Alt + to scroll to the next and previous quote details in the scroll.
>
> **You cannot maintain quote details on this page.** All maintenance functions (generate, delete, accept, decline and cancel) are performed on the **Main** tab.

Premise Info is the address of the main premise associated with the service agreement (as defined on the service agreement's characteristic premise). It's important to note that the service agreement may cover service at multiple premises, but only one may be defined as the main premise.

Quote Detail ID is the system-assigned unique identifier of the quote detail.

SA Info is a concatenation of the proposal SA's division, SA type, start option (if any), status and start date.

SA Version is a snapshot of the proposal SA's version number at the time the quote detail was generated. This information is used by the system to detect if the proposal SA was modified after the quote detail was created.

Total Amount is the sum of the quote detail's simulated bill segments.

Create Date / Time is the date and time the quote detail was generated.

Remarks highlight special circumstances associated with the quote detail. Please see the description of this field on the main tab for more information.

The grid contains a row for each simulated bill segment linked to the quote detail. The following information appears in the grid:

- The first go to button transfers you to the bill segment page where the details of the simulated bill segment are shown.
- **Bill Segment Period** defines the start and end dates of the simulated bill segment.
- **Amount** contains the amount of the bill segment.
- **Remarks** highlight special circumstances associated with the bill segment.

Copyright © 2026, Oracle and/or its affiliates.

- **Billing Scenario ID** allows you to go to the proposal SA's billing scenario where you can tinker with consumption and the bill segment period used to generate this bill segment. Refer to Proposal SAs Contain Billing Scenarios for more information.

Quote - Quote Routings

This page is used to view the recipients of a quote. Refer to Printing Quotes for a discussion of how this information is used to produce quote images.

This page is also used to request a new copy of a quote.

> **Note**
>
> **No routing information until completion.** A quote has no routing information until it is complete. At completion time, the system creates a routing detail for every person linked to the account that receives a copy of its quotes. Refer to Printing Quotes for more information about how this information is constructed.

Use **Menu**, and then **Sales & Marketing**, and then **Quote**, and then **Search** and navigate to the **Quote Routings** page.

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Routing Information** scroll contains an entry for each recipient of the quote. This information is disabled after the quote has been routed to the recipient. The remainder of this discussion describes the fields that can be defined for each recipient of the quote.

Person ID identifies the person who was originally associated with the routing record. You may change the **Person ID** to any person in the system as this information is only used to default the recipient's name and address information, below.

The **Quote Routing Parameters** area contains information explaining how and when the quote is routed to the recipient. This information is gray when the quote routing information has been extracted or if a quote is reopened:

- **Quote Route Type** is the method used to transmit the quote to the customer. This information defaults from Account - Person Information.
- **Batch Control** is the identifier of the batch of quotes in which the recipient's copy was downloaded. Embedded in this identifier is the batch control ID and the batch number. Note, this batch process is defined on the Quote Route Type or CIS Division.
- **Sequence** is the system-assigned identifier assigned to this quote routing.
- Turn on **Do Not Extract** if the quote should not be routed to the individual for whatever reason.
- **Reprint** is checked if an operator created this routing. This switch is populated by the system and is always protected.
- **Extract Date/Time** is the date and time the quote was downloaded. This information only appears after the download has commenced.

Copyright © 2026, Oracle and/or its affiliates.

The **Mailing Address Information** area contains the recipient's name and address.

- If the recipient has an override mailing name , **Name1**, **Name2** and **Name3** contain the person's override mailing name. Otherwise, **Name1** contains the recipient's primary name (this name might also include a prefix or suffix if the related Account - Person relationship has these fields defined).
- The remaining fields are the recipient's address. Refer to Completing A Quote Creates Quote Routings (And More) for information about how the system constructs the recipient's address
- If the recipient's address is a mailing address (as opposed to fax or email), the **Country** associated with the address controls the fields that appear in the address. Refer to Defining Countries for more information on the address constituents.
- If a user manually adds a routing record, the mailing address information defaults from the information on the current routing record.
- If a user specifies a **Person ID** that is linked to the quote's account, the **Quote Route Type** defined for this person on the account is defaulted.
- If a user specifies a **Person ID** that is not linked to the quote's account, the **Quote Route Type** on the installation record is defaulted. Note, if the quote route indicates the quote is routed via the postal service, the system will only be able to default the person's address if the person has an override mailing address on their correspondence information.
- If the user changes the **Quote Route Type** after the initial default, the system populates the address information as described above.

Service Credits

"Service credits" is a term used to describe a special program designed to reward a customer for service along with benefits related to such a program. For each program that a customer participates in, there is a service credit membership record. Service credit events are used to record the "points" earned and redeemed for the membership. The topics in this section provide further details related to the service credit functionality.

NOT_SUPPORTED

For information about designing and setting up your control tables for configuring service credit functionality, refer to Designing Your Service Credit Options.

Note

Service credits are optional. Service credits may not be set up in your system; they are only used if your company offers rewards programs to customers.

Penalty Points. The service credits functionality is described in the documentation with the assumption that it is used for accumulating points to reward your customers. However, if your company has a business need to record penalty points for a customer, the service credits functionality may be used for that purpose as well.

Copyright © 2026, Oracle and/or its affiliates.

- The Big Picture Of Service Credit Membership
- The Big Picture of SC Events
- Maintaining Service Credit Memberships and Their Events

The Big Picture Of Service Credit Membership

If a customer is participating in a rewards program, the customer has a service credit membership. The topics in this section provide background information about a variety of service credit membership features and functions.

- Who are the Members?
- Membership Application
- Creation of a Membership
- Membership Fees
- Stopping a Membership vs Inactivating a Membership

Who are the Members?

Each membership has one or more "members". The members are defined by the accounts linked to the membership and by the financially responsible person(s) linked to each account.

A typical membership is linked to only one account. However, in some cases more than one account is linked to the membership. For example, a customer may have one account for residential service and one for commercial service, but both accounts are part of the same membership. In this example, one person with multiple accounts is linked to the membership, so there is only one member.

For each account linked to the membership, the service related to the account may be used to calculate the accumulated credits for the membership. Refer to How Are Service Credits Earned? for more information.

> **Note**
>
> An account may not be linked to more than one membership of the SAME membership type at a given point in time. In other words, an account may not have two United Airlines frequent flier memberships (otherwise, the account would accumulate double miles). However, an account may be linked to more than one membership if they are of different membership types. For example, an account may have a capital credits membership and a United Airlines frequent flier membership.

When you find a customer's record on Control Central - Search, you are transferred to the Account Information portal. One of the zones on this page contains information about memberships for the customer.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Portal Preferences.** The service credit membership zone only appears if you have modified your preferences to display this zone. Refer to Portal Preferences for more information.

- Joint Membership
- Changing the Membership Accounts

Joint Membership

It is possible for more than one person to belong to the same membership. These persons may be linked to the same account or to different accounts. A typical scenario would be a married couple that shares in a membership. This is considered a "joint" membership.

The number of "members" for a membership is mainly for information purposes and typically has no bearing on the service credit amounts calculated for the membership. There may be legal restrictions or business practice restrictions controlling how you may set up your members. For example:

- In an airline's frequent flier program, a separate frequent flier account would exist for each customer. It is possible for you to create a frequent flier membership with more than one "member", but for the system, the important information is the external frequent flier miles number linked to the membership. The customer linked to this number accumulates the miles. For this case, your business practice may be that the person with the frequent flier account must be the only financially responsible person.
- For a capital credits membership, you may be required to distinguish between "single" memberships and "joint" memberships. If a single person gets married and would like the spouse to be added as a member, your practice may be to expire the single membership and to create a new "joint" membership to include both spouses.

> **Note**
>
> **Co-op Members**. Do not confuse the "members" of a co-op with the members linked to a service credit membership. At a cooperative, most customers are considered "members" of the co-op. However, in the system, separate membership records are created for each single or joint membership.

Changing the Membership Accounts

The link between the accounts and the membership is NOT effective dated. This means that if the list of accounts for a membership changes, you need to stop the membership and start a new one with the new list of accounts.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Stopping Memberships**. Stopping a membership does not necessarily prevent new service credits from accumulating. Refer to Stopping a Membership vs. Inactivating a Membership for more information.

Membership Application

Your business practice may dictate that the customer complete an enrollment form. Depending on your practice, the form may need to be received prior to starting membership or it may simply be a mechanism to gather supplemental information once the membership has begun.

Either way, you may consider setting up the system to handle forms as follows:

- Design your membership type to include a creation algorithm that generates a letter when the membership is created. The letter template could include the form to be filled out.
- Create the new membership in pending status. The letter and form are sent to the customer and the pending status indicates that you are waiting for the completed form. Refer to the Lifecycle of a Membership for more information.
- When the paperwork is received, change the status of the membership to active.

> **Note**
>
> **Membership Events for Pending Memberships**. The system allows events to be created for pending memberships assuming that paperwork is supplemental information that should not prevent service credits from accumulating. If your business practice dictates that the paperwork must be received prior to accumulating service credits, the algorithm or background process that creates events for memberships of this type should be coded to only create events for active memberships. Refer to How Events Are Created for more information.

Creation of a Membership

Memberships may be created using the membership page when a customer has signed up for a special program. In addition, it may be your business practice to automatically create a membership under certain circumstances when a customer has signed up for service.

For example, if you are a co-op, perhaps every new customer is automatically enrolled in a capital credits membership. Or perhaps signing up for cable service automatically entitles your customers to membership in the free pay-per-view movies program. For these cases, the recommendation is to use the order page to start service for your customer and to use questions and miscellaneous fields to record information needed to set up the new membership correctly. Refer to Using Campaigns & Packages to Create a Membership.

Copyright © 2026, Oracle and/or its affiliates.

Membership Fees

Some memberships may require a fee to be paid by the customer when joining. The behavior of this fee may differ depending on whether it is refundable or not.

- If the fee is nonrefundable, you create an SA with a simple adjustment or a one time billable charge to levy the fee. Refer to Defining Control Tables for a Nonrefundable Fee.
- If the fee is refundable, in addition to defining the SA to hold the fee, you need to put code in place to initiate the refund. Refer to Setting Up a Refundable Fee for more information.

Stopping a Membership vs Inactivating a Membership

You may stop a membership by entering an end date for the membership. However, depending on how events are created for your membership, stopping the membership may not prevent new events from being created. For example:

- If you stop a frequent flier membership on May 15th and a bill is subsequently created for the period May 1st through June 1st, your business rules may dictate that frequent flier miles should be accumulated for this bill because the membership was active for part of the period covered by the bill.
- For a capital credits membership, capital credit allocations are calculated for the prior year. If a membership stopped in the middle of the previous year, the membership should still get capital credits for the part of the year that the membership was still in effect.
- Capital credit retirements are calculated based on capital credit allocations accumulated in the past. If a membership has been expired, the members are still eligible for retirement amounts for the accumulated capital credit allocations. These retirements may continue for many years after a membership is stopped.

> **Note**
>
> **Membership Status**. For all the above scenarios, the system assumes that the status of the membership remains active.

You may have some circumstances where you do not want new events to accumulate after a membership is stopped. To prevent new events from being created, you must change the status of the membership to inactive.

If a customer moves away or otherwise stops service with your company, your business rules dictate whether or not the membership should be stopped or changed to inactive. The following issues must be considered:

- Will service credit events still get created for these memberships? If so, is that acceptable?
  - For frequent flier memberships, the miles are calculated based on new bill amounts for the membership's service agreements. If the SAs that contribute to the membership are expired, no new bills are generated so no new miles are accumulated.
- For capital credits, after a customer moves away, the capital credit allocation for the final year of the customer's service may still need to be calculated. In addition, this customer is

Copyright © 2026, Oracle and/or its affiliates.

eligible for the capital credit retirement amounts, even though they are no longer a customer.

- What happens if the customer returns to your company and signs on for service again? Assuming that the same account number is used, is it ok for the membership to be automatically reinstated? Or should the customer reapply? If the customer should reapply, you may consider updating the membership with a stop date when the customer originally stopped service.

> **Note**
>
> There is nothing in the system logic that automatically updates the membership stop date or the status when service agreements that contribute to the membership are stopped.

The Big Picture of SC Events

The service credit events record the accumulation and redemption of awards earned by the membership. The topics in this section provide background information about a variety of service credit event features and functions.

> **Note**
>
> **Events.** For convenience, this document often refers to service credit events simply as "events".

- Event Amounts May Contribute to a Balance
- Events May Indicate a Fiscal Year
- Events May Indicate a Subcategory
- An Event May Cause Other Actions to Occur
- Financial Transactions for an Event
- How Events Are Created
- Creating Pending Events
- Events May Be Transferred
- CPCRALOC - Allocating Capital Credits
- CPCRRETR - Capital Credit Retirement

Event Amounts May Contribute to a Balance

The event amount may be positive or negative, based on whether the event is accumulating points earned for the membership or redeeming points for the membership. For some types of memberships, these credit and debit amounts should be added together to produce the overall balance for the membership.

Copyright © 2026, Oracle and/or its affiliates.

For some types of memberships, a balance is not applicable. This is true for memberships where the redemption of the points does not occur within the system. For example, if the membership accumulates frequent flier miles on behalf of an airline, the customer redeems the earned miles with the airline, not with your company. As a result, there is no need to calculate a balance for the membership.

If you want the system to calculate a balance, you must set a switch on the membership type page. For memberships that calculate a balance, the balance may be viewed on the membership page.

Events May Indicate a Fiscal Year

For some types of memberships, the amount of each event is associated with a specific fiscal year. This is true for capital credit memberships where capital credit allocations are calculated for a fiscal year. When the capital credits are redeemed (also called retired), the retirement events must also indicate a fiscal year. Because all events for this type of membership are related to a fiscal year, the system can calculate the balance for the events by fiscal year.

If you want the system to require a fiscal year for every event in the membership, you must set a switch on the membership type page. For memberships that calculate a balance and indicate a fiscal year, the balance for each fiscal year may be viewed on the membership page.

Events May Indicate a Subcategory

For some types of memberships, the amount of each event is further grouped by a subcategory. This is common for capital credit memberships where multiple capital credit allocations are calculated. For example, each year, you may calculate allocations for generation and for transmission. Events for memberships of this type must indicate a subcategory.

Cooperatives that allocate capital credits for more than one subcategory, typically want to view the balance of the events over time broken down into these subcategories. When the capital credits are redeemed or retired, the retirement events must also indicate a subcategory. Because all events for this type of membership are related to a subcategory, the system can calculate the balance for the events by subcategory.

> **Note**
>
> **Partial Retirement**. For some cooperatives, it is common that certain capital credit allocations are never retired. Refer to Partial Retirement for more information.

If you want the system to require a subcategory for every event in the membership, you must define the valid subcategories on the membership type page. For memberships that calculate a balance and indicate a subcategory, the balance for each subcategory may be viewed on the membership page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Calculating Subcategory Balances**. If events for your membership indicate a subcategory and fiscal year, balances for the membership are calculated for both groupings.

An Event May Cause Other Actions to Occur

When an event is generated, there are often other actions that need to take place. For example:

- If an event is created that allocates points for the customer, you may want to generate a temporary message on the customer's next bill.
- If a capital credit allocation event is created, the general ledger is affected so an adjustment must be created to post to the general ledger. Refer to Financial Transactions for an Event for more information.
- If an event is created to add frequent flier miles, the event must be marked for extraction so that the information is sent to the appropriate airline. Refer to Interface Membership Information to a Third Party for more information.
- If an event to retire capital credits is created, the customer's balance is affected so an adjustment must be created to transfer the capital credit to the members' service agreements. Refer to Financial Transactions for an Event for more information.

The system relies on algorithms linked to the event type to perform the actions described above and any other action that may be required by your business.

> **NOT_SUPPORTED**
>
> Refer to Service Credit Event Types for more information about the algorithms provided with the base product.

> **Note**
>
> **Event Cancellation Algorithms**. The event type also defines algorithms to be executed when an event is canceled.

Financial Transactions for an Event

An event may be related to financial transactions. There are two types of financial transactions that may be linked to the event:

- Financial transactions that contributed to the event. These are bills, payments or adjustments that caused the event to be created. For example, if you generate an event for frequent flier miles as a result of some amount billed for the customer, the bill segments

Copyright © 2026, Oracle and/or its affiliates.

associated with the billed amount are linked to the event. The FTs are linked to the event for audit purposes.

- Financial transactions that are created by the event. In an event may cause other actions to occur, we saw examples where an event completion algorithm may cause adjustments to be created. The FT ID for these adjustments is linked to the event as audit.

> **Note**
>
> **Automatically Canceling Event Financial Transaction**. The created by financial transactions may get canceled if the event is canceled AND you have plugged in an appropriate Event Cancellation Algorithm.

For financial transactions that are created by the event, the system must post the financial transactions to one or more appropriate service agreements. The service agreement to use depends on your business rules. For example:

- If the event affects the balance of accounts linked to the membership, the event completion algorithm may create adjustments for service agreements that contribute to the membership.
- If the event affects the general ledger and does not affect any balance for the account, the event completion algorithm simply needs a service agreement to use for the resulting adjustment. Rather than arbitrarily picking any service agreement in this case, a special miscellaneous service agreement should be linked directly to the membership for such adjustments.

How Events Are Created

Although a user may create an event manually on the service credit event page, most events are created automatically by an algorithm or by a background process. The following topics further describe these options.

- Events Created by an Algorithm
- Events Created as a Result of a Background Process

Events Created by an Algorithm

You would design an algorithm to create your events if the event should be triggered by an occurrence in the system, such as starting service or generating a bill. For example, perhaps every time you are billed for electric and/or gas service, the system adds one frequent flier mile for every $10 spent on service. Refer to How Are Service Credits Earned? for more information.

The base product provides a bill completion algorithm to create an event as a result of the amount billed for service agreements linked to the membership. Refer to CBCM-SC for more information.

Copyright © 2026, Oracle and/or its affiliates.

Events Created as a Result of a Background Process

You would use a background process to create the event if the calculation is done periodically or on an ad-hoc basis and does not get triggered by something occurring in the system. For example, the capital credit allocation and retirement processes are typically run once a year, when the business has determined that they are ready to calculate these amounts.

> **NOT_SUPPORTED**
>
> Refer to Allocating Capital Credits and Capital Credit Retirement for more information.

Creating Pending Events

Events may be created in pending or complete status. Only complete events cause the completion event algorithms to run and cause other actions to occur. You may use the status of pending if you want to review the calculated amount prior to completing the event.

For example, when a cooperative runs the background process to allocate capital credits, it may create the new events in the pending status. This allows the co-op to review the total amount of capital credits calculated for all members. If this total is in line with the co-op's expectations, they run the background process again creating the new events in complete status. Only at this point do the completion event algorithms run.

Events May Be Transferred

In some situations, you may want to transfer service credit events from one membership to another on the service credit event page. For example, if a member dies, you may decide to transfer earned service credits to the membership for a beneficiary. When transferring a service credit:

- The target membership must be of the same membership type as the original membership.
- The original service credit event is canceled and any event cancellation algorithms linked to the SC event type are executed.
- A new service credit event is created and completed and any event completion algorithms linked to the SC event type are executed.
- The new service credit event references the original event ID in the **Transferred From** event field.

> **Note**
>
> Contributed to FTs linked to the original event are not transferred to the new event. If you have created by FTs linked to the original event, the recommendation is to ensure that an event cancellation algorithm cancels this FT and that an event completion algorithm for the new event creates an appropriate new FT for the new membership.

Copyright © 2026, Oracle and/or its affiliates.

CPCRALOC - Allocating Capital Credits

The base product provides a background process (CPCRALOC) that allocates capital credits to members of a cooperative. The result of this process is the creation of new service credit events for capital credit memberships. This background process is typically run after the profits for the previous year are calculated. During that time, this job could be run a number of times for any of the following reasons:

- At the beginning of the allocation process, you may know the total amount to allocate to all members; however, you may estimate the allocation factor that determines the individual amounts that get distributed to the members. You may need to change your estimate and rerun the process until all the individual allocations add up to the desired total amount. To run the job with the estimated allocation factors, indicate that you want events created as pending. When you confirm the correct allocation factor, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type, subcategory and fiscal year. Refer to Creating Pending Events for more information.
- If the cooperative has more than one subcategory, the process must be run for each subcategory. In fact, the above trial-and-error process may need to be followed for each subcategory.

You must provide the service credit event type to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type. To calculate the capital credit allocation, the system finds all applicable bill segments for each service agreement that contributes to the membership.

Capital credit allocations are calculated for a specific fiscal year. The fiscal year is input to the background process and is stamped on any new events created by the process. The system also uses the fiscal year to determine the applicable bill segments for the membership's SAs. It uses the bill segment's accounting date and the accounting calendar to determine the fiscal year for each bill segment.

> **Note**
>
> **Canceled Bill Segments**. Any bill segments that were canceled in the fiscal year are included as well.
>
> **Accounting Calendar**. This process assumes that only one accounting calendar exists for the company. This is the case for companies that operate in a single jurisdiction, which is true for cooperatives.
>
> **Allocation Period**. If the membership was only effective for part of the fiscal year, the allocations are only calculated for the period of time that the membership was in effect.

For each applicable bill segment, the system determines the SQ total and the total sales amount for each segment. To calculate the totals correctly, a characteristic type and value to identify SQ bill lines must be provided to the background process and a characteristic type and value to identify bill lines contributing to the total sales amount must be provided to the background process.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Identifying SQ and Sales Information for Historical Bill Segments for more information.

If your membership type indicates subcategories to be used for the events, you must provide a subcategory as input. If you have multiple subcategories for your membership, this process must be run once for each subcategory.

The calculation of the capital credit allocation amount typically uses an allocation factor. This factor is often different for each subcategory. You will probably define a different bill factor for each subcategory and use the appropriate bill factor when submitting the process for each subcategory.

This process also assumes that the allocation factor differs by the customer's revenue class. It assumes that the bill factor's characteristic type is revenue class. Refer to Designing Bill Factors for Credit Allocation for more information. For each service agreement linked to the membership, the appropriate allocation factor is determined based on the revenue class for the SA type.

Capital credits for the input subcategory are calculated as follows:

Fiscal year sales amount - (fiscal year SQ amount * revenue class allocation factor) = Allocation Amount

The calculated amount is compared against a minimum amount, input to the background process. If the calculated amount is less than the minimum, no event is stored for this amount.

After determining the amount, a new service credit event is created for the membership. The event is stamped with the input subcategory, service credit event type and fiscal year.

The FTs for all the bill segments used to determine the allocation amount are linked to the event as contributed to FTs. Refer to Financial Transactions for an Event for more information.

Note

Rerunnable Process. This process is restartable and rerunnable. When running the process, for each membership the system checks whether there is already a complete event for the input service credit event type, subcategory and fiscal year. If so, it does not create a new event.

NOT_SUPPORTED

Refer to CPCRALOC for more information.

CPCRRETR - Capital Credit Retirement

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Calculation Method**. There are various calculations that different cooperatives may use when determining the amount to retire. The base product has provided one calculation method.

This background process is run when a cooperative is ready to pay out allocated capital credits to its members. This process is referred to as retirement. The result of this process is the creation of new service credit events for capital credit memberships. Capital credits are paid as the cooperative's financial position permits. During that time, this job could be run a number of times for any of the following reasons:

- You may choose to run the process and create pending events to review the results prior to finalizing the retirement. When you are happy with the results, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type and subcategory.
- If the cooperative has more than one subcategory, the process may be run separately for each subcategory. You may also run the process with no subcategory indicated and then all subcategories are processed.

> **Note**
>
> **Partial Retirement**. For some cooperatives there may be capital credits for some allocated amounts that are never retired. If that is the case, simply run this process for each subcategory whose amounts do get retired.

You must provide the service credit event type to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type.

This process receives a percentage to retire as input. It determines the current balance of all completed events for the subcategory. The retirement amount for the subcategory is calculated as the current balance * input percentage to retire. The process then determines the outstanding balance for each subcategory and fiscal year combination. For every combination that still has a balance, the process creates events to relieve the outstanding balance for the oldest fiscal year first. It creates events for subsequent fiscal years until the full retirement amount is used.

In order for the process to be rerunnable and restartable, a Control ID is used to mark retirement events that are created by the process. The Control ID is input to the routine and is determined by the user. When the process runs, it determines whether an event already exists for the subcategory / Control ID combination. If so, new events are not created for that subcategory. If you run the process multiple times for the reasons described above, you should use the same Control ID each time. This way the system knows that the process is being run again for the same retirement calculation.

> **NOT_SUPPORTED**
>
> Refer to CPCRRETR for more information.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Service Credit Memberships and Their Events

The topics in this section describe how to maintain a service credit membership and its events.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Service Credits for background information about service credits.

- Service Credit Membership
- Maintaining Service Credit Events

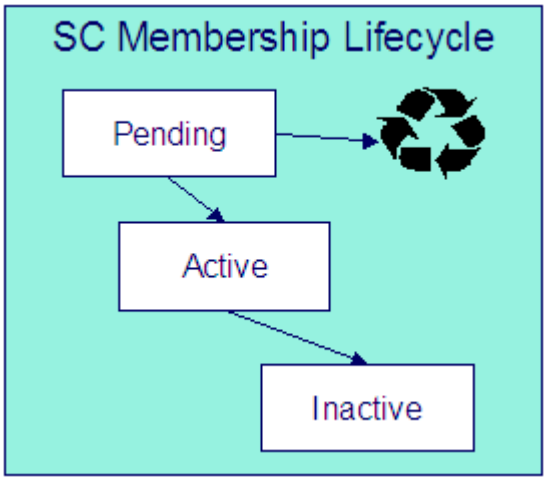
Service Credit Membership

The topics in this section describe how to maintain service credit memberships. Refer to The Big Picture Of Service Credit Membership for background information about memberships.

- Lifecycle of a Membership
- Service Credit Membership - Main
- Service Credit Membership - Account
- Service Credit Membership - Characteristics
- Service Credit Membership - Membership Tree

Lifecycle of a Membership

The following diagram shows the possible lifecycle of a service credit membership.



Copyright © 2026, Oracle and/or its affiliates.

Pending A service credit membership may start out in the Pending state. A membership is typically created in this state if the creation of the membership requires approval or any manual step (e.g. waiting for paperwork) that needs to be completed prior to activation.

> **Note**
>
> **Events for Pending Memberships**. Service credit events may still be created for membership records in the Pending state.

Active A service credit membership is in the Active state when a customer is actively participating in a service credit membership. A membership may be created in the Active state or it may transition to the Active state from Pending when all the necessary approval or paperwork is complete.

> **Note**
>
> **Manual Activation Only**. There is no system functionality that automatically changes the status of a membership from Pending to Active. This is a manual step.

Inactive If a customer is no longer actively participating in a service credit membership, change the status to Inactive. This will prevent any new events from being created for the membership.

Service Credit Membership - Main

This page is used to maintain and display general information about a service credit membership. Open this page using **Menu**, and then **Customer Information**, and then **Service Credit Membership**, and then **Search**.

Description of Page

Membership Info and **Membership ID** are displayed on every page. These values only appear after the membership exists on the database. The **ID** is a system-assigned random number. The **Membership Info** is a concatenation of important details about the membership.

Membership Status defines the state of the membership.

> **NOT_SUPPORTED**
>
> Refer to Lifecycle of a Membership for more information about membership status.

Click the **Activate** button to activate a Pending membership. When this button is clicked, the system executes any Membership Activation algorithms that are linked to the service credit membership type.

Click the **Inactivate** button to transition an Active membership to Inactive. When this button is clicked, you are prompted to provide an **Inactive Reason** and the system executes any Membership Inactivation algorithms that are linked to the service credit membership type.

Copyright © 2026, Oracle and/or its affiliates.

Indicate the **SC Membership Type**. This field is important as it controls numerous aspects of the behavior of the membership and its events.

> **NOT_SUPPORTED**
>
> For more information about what the membership type controls, refer to Setting Up Service Credit Membership Types.

Indicate the **Start Date** and **End Date**, which define the period for which this membership is effective. Leave the end date blank if this membership is effective indefinitely. Refer to Stopping a Membership vs. Inactivating a Membership for more information.

If this membership is related to a record in an external system, use the **External ID** to indicate the unique identifier of the record in the external system.

If events linked to your membership may generate miscellaneous adjustments, indicate the **Service Agreement** to use for these miscellaneous transactions. The account for this service agreement must be one of the accounts linked to the membership.

The **Events** grid displays the events linked to the membership. To view any of the events use the drill down button.

If your SCM type indicates that the membership has a balance, the tree at the bottom displays information about the balance of the membership based on the amounts for completed events. If the SCM Type indicates subcategories and/or a fiscal year is required, subtotals for completed events are also displayed for each subcategory / fiscal year and by fiscal year / subcategory, if applicable. If any pending events exist, the tree includes a node with information about these events.

Service Credit Membership - Account

This page is used to maintain the accounts linked to a service credit membership. Open this page using **Menu**, and then **Customer Information**, and then **Service Credit Membership**, and then **Search** and navigate to the **Account** page.

Description of Page

Indicate the **Account ID** for each account linked to this membership. Typically a membership will be linked to only one account. However, if a person has multiple accounts and the service credits should span the accounts, multiple accounts are linked to the same membership.

The Service Credit Membership zone on the Account Information portal contains information about memberships for the current customer. (When you find one of the customer's records on Control Central - Search, you are automatically transferred to the Account Information portal.)

> **Note**
>
> **Portal Preferences.** The Service Credit Membership zone only appears if you have modified your preferences to display this zone.

Copyright © 2026, Oracle and/or its affiliates.

Service Credit Membership - Characteristics

The characteristics page contains information that describes miscellaneous information about the service credit membership. Open this page using **Menu**, and then **Customer Information**, and then **Service Credit Membership**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined as permissible on the service credit membership record. Refer to Setting Up Characteristic Types & Their Values for more information.

To modify a characteristic, simply move to a field and change its value. To add a new characteristic, click + to insert a row then fill in the information for each field. The following fields display:

Effective Date Define the date on which the characteristic becomes effective. The effective date defaults from the Installation Record.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

> **Note**
>
> **Default Note.** A service credit membership's characteristics default from the service credit membership type.

Service Credit Membership - Membership Tree

The membership tree page displays accounts, SAs and persons linked to the service credit membership. Open this page using **Menu**, and then **Customer Information**, and then **Service Credit Membership**, and then **Search** and navigate to the **Membership Tree** page.

Description of Page

The tree displays each account linked to the membership. For each account, a node exists for:

- Each service agreement whose SA type is included in the service credit membership type 's SA type collection.
- Each financially responsible person linked to the account. These persons are considered the members.

Copyright © 2026, Oracle and/or its affiliates.

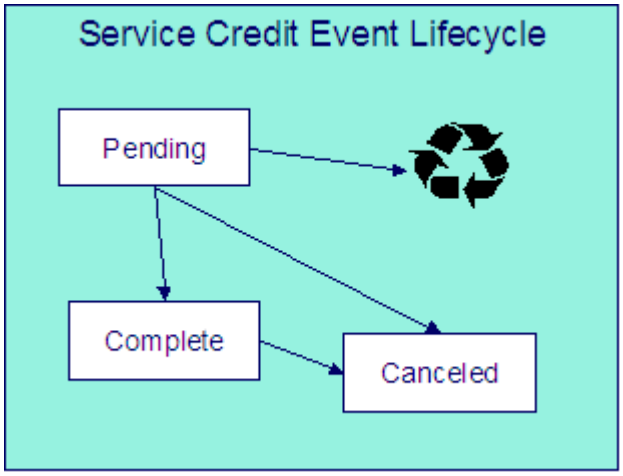
Maintaining Service Credit Events

Service credit events record credits earned or redeemed for a given membership. The topics in this section describe how to maintain service credit events. Refer to The Big Picture of Service Credit Events for background information about memberships and their events.

- Lifecycle of a Service Credit Event
- Service Credit Event - Main

Lifecycle of a Service Credit Event

The following diagram shows the possible lifecycle of a service credit event.



Pending A service credit event may start out in the Pending state. An event is typically created in this state when a review of the event is desired before completing it. Refer to Creating Pending Events for more information.

Complete A service credit event is typically created in the Complete state. When an event is completed, the system executes any event completion algorithms indicated on the service credit event type.

Canceled If a service credit event is no longer valid, it may be Canceled. When an event is canceled, the system executes any event cancellation algorithms indicated on the service credit event type.

Service Credit Event - Main

Open this page using **Menu**, and then **Customer Information**, and then **Service Credit Event**, and then **Search.**

Description of Page

The **Service Credit Event Info** is a concatenation of important details about the service credit event. The **Service Credit Event ID** is a unique system-assigned random number. These values only appear after the service credit event exists on the database.

Copyright © 2026, Oracle and/or its affiliates.

Service Credit Event Status defines the state of the service credit event.

> **NOT_SUPPORTED**
>
> Refer to Lifecycle of a Service Credit Event for more information.

Click the **Complete** button to change a Pending event to Complete. When this button is clicked, the system executes any event completion algorithms that are linked to the service credit event type.

Click the **Transfer** button to transfer this event to another membership. When this button is clicked, you are prompted to provide the membership ID to transfer the event to. The new membership must have the same membership type as the original membership. Refer to Events May Be Transferred for more information.

Click the **Cancel** button to cancel this event. When this button is clicked, the system executes any event cancellation algorithms that are linked to the service credit event type.

The **Create Date/Time** captures the date and time that this event was created. This defaults to the system date/time.

Indicate the service credit **Membership ID** to which this event should be linked.

Indicate the **Service Credit Event Type**. This field may define algorithms to be executed when the event is completed or canceled. Refer to Setting Up Service Credit Event Types for more information.

If your service credit membership type indicates one or more subcategories to be associated with events for this type of membership, indicate the appropriate **Subcategory Name** for this event.

If your service credit membership type indicates that fiscal year is required, indicate the appropriate **Fiscal Year** for this event.

Enter the **Service Credit Event Amount**. If your service credit membership type indicates that your membership has a balance, this amount is used to calculate the overall balance for the membership as well as a balance for the subcategory and/or fiscal year, if applicable.

If this event was created as a result of a transfer from another membership, the **Transferred From Event** ID is displayed.

If the information about this event may be downloaded to a third party system, you should have an event completion algorithm that stamps the appropriate **Batch Control** and **Batch Number** onto the event.

The FT collection displays information about financial transactions related to this event. Two different categories of financial transactions may be linked to the event.

- FTs that contributed to this event. These financial transactions played a part in the creation of this event. For example, if frequent flier miles are earned as a result of newly created bill segments, the FTs related to these bill segments are linked to the event and marked as contributed to FTs. This information is purely for audit purposes. It is the responsibility of the algorithm or background process that creates the event to populate the collection of contributed to FTs. Refer to How Events Are Created for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The amount of the contributed to FTs will not necessarily add up to the amount of the event. Refer to the algorithm or background process that calculates the amount of the event to understand how the event amount is calculated based on the amount of the contributed to FTs

- FTs that were created by this event. These financial transactions are created as a result of this event. For example, if you have generated a capital credit allocation event, perhaps you need to post information to the general ledger. An allocation event should therefore generate an adjustment to post the effect on the GL. The adjustment's FT is linked to the event as an audit. In order to create a financial transaction related to this event, you must plug in an appropriate event completion algorithm on your service credit event type.

> **NOT_SUPPORTED**
>
> Refer to An Event May Cause Other Actions to Occur for more information.

Loans

There are many reasons why your company may loan money to customers. For example, your company may loan money to:

- Finance service upgrades (e.g., to change to dual fuel)
- Support economic development (e.g., to extend a power line)
- Promote energy conservation (e.g., customers borrow to insulate their house)

> **Note**
>
> **Loans are optional.** You need only set up loans if your organization loans money to customers. For information on setting up the system to enable loans, refer to Defining Loan Options.

- The Big Picture Of Loans
- Loan Maintenance
- How To

The Big Picture Of Loans

The topics in this section provide background information about loans.

Copyright © 2026, Oracle and/or its affiliates.

- Loans Are Service Agreements
- Loan Interest Charges
- Loan Amortization Schedules
- Loan Overpayments
- The Lifecycle Of A Loan
- Finalizing And Closing A Loan
- Navigating To The Loan Maintenance Page

Loans Are Service Agreements

Loans are initiated by creating a loan service agreement for a customer. The loan service agreement (and its SA type) contains the loan's terms, including the loan amount, periodic payment amount, the number of amortization periods, the interest rate, and the periodicity of the bills. Loan service agreements are just like other service agreements in many ways:

- The system bills the customer for the periodic payment amount using information on the service agreement.
- The customer pays the billed amount.
- If the customer doesn't pay, credit and collections may start a collection process to encourage the customer to pay (depending upon how the loan debt class has been set up on your system).

Loan service agreements differ from other service agreements in the following ways:

- Loans have interest charges.
- Loans have amortization schedules.
- Loan SAs are maintained on a separate transaction (refer to Loan - Main for more information).

> **Note**
>
> **Loan service agreements are created using Start/Stop.** Loan service agreements are created using start/stop just like all other service agreements. The start/stop transaction has special loan functionality that allows an operator to specify the service agreement-specific loan terms described above. A start option can be specified to override the SA type's interest bill factor.

Loan Interest Charges

When a bill is generated for a loan, the interest charge is calculated using an algorithm. The base package loan interest charge algorithm type (LINT-SI) calculates simple interest based on:

- Unbilled principal (i.e., the payoff balance minus the current balance)
- The number of billing periods covered by the bill

Copyright © 2026, Oracle and/or its affiliates.

- The interest rate

> **Note**
>
> **No interest on interest and no interest on past due amounts.** Just like a classic home loan, the base package algorithm does not calculate interest on interest, nor does it calculate interest on past due amounts. However, your system may be set up to levy a late payment charge.

After the interest charge is calculated, the bill segment FT algorithm generates the financial transactions for the bill. The base package FT algorithm generates a financial transaction that adds the interest charge to the payoff balance. In other words, if the customer wanted to pay off the loan, they would owe the principal balance plus the interest charge.

> **NOT_SUPPORTED**
>
> For more information, refer to Billing For Loans And Interest Calculation.

Loan Amortization Schedules

A loan's periodic payment amount is calculated such that the customer repays the loan in equal amounts over the term of the loan (with the exception of the first and last bill segments, which are typically for different amounts). The amortization schedule is a projection of the amount of principal and interest in each payment over the life of the loan. The amortization schedule may change, for example if the interest rate changes or the customer makes an overpayment (reducing the principal balance).

The amortization schedule is a theoretical projection of the future. The term theoretical is used because the customer's actual payments can cause the schedule to change. For example, if a customer overpays a bill, the loan's length decreases and the amount of principal and interest in each scheduled payment changes.

A loan's amortization schedule is calculated when you click the **Calculate** button on Loan - Main . When this button is clicked, an algorithm plugged in on the loan SA type actually calculates the amortization schedule.

You can view a loan's latest amortization schedule using Loan - Main . You can also change the loan's payment periods or periodic payment amount on this page and recalculate the loan's amortization schedule.

- Negative Amortization

Negative Amortization

The system does not allow you to use a periodic payment amount that is less than the periodic interest amount. If it did, the customer's payoff balance would increase over time (this is referred to as "negative amortization"). There are two methods you can use to specify a payment amount in the system. Depending on which method you use, the system informs you of negative amortization differently:

Copyright © 2026, Oracle and/or its affiliates.

- If you enter a payment amount that is too small on the Start confirmation page or in the **Renegotiate Future Payments** grid on Loan maintenance, you get an error message when you click the **Calculate** button. You must fix the payment amount before you can start the loan or save your changes.
- If you manually override the **Payment Amount** on Loan maintenance, you can save a payment amount that is too small to cover the interest charges. However, the system notifies you that it is not able to calculate the **Forecasted Schedule** because of negative amortization. Additionally, bill segments for the loan SA are produced in error until the payment amount is adjusted appropriately.

Loan Overpayments

Overpayments reduce the principal amount (the amount owed on the loan), which follows the philosophy adopted by a typical home loan. When the payment is made, any overpayments are distributed according to the overpayment distribution algorithm defined for the customer class. Typically, loan service agreements should be set up so that the overpayment is applied to the principal balance.

When the payment transaction is frozen, the system checks to see if there is a credit amount on the loan SA's current balance. If a credit exists, the customer has made an overpayment and an adjustment is created to transfer the amount of the credit from the SA's current balance to its payoff balance, thus reducing the principal by the amount of the overpayment.

NOT_SUPPORTED

For more information about the adjustments created for loan overpayments, refer to Overpayments On Loans.

Note

Overpayments and interest. The base package interest calculation algorithm (plugged in on the loan's SA type) does not take into consideration the exact date that the overpayment is made when calculating the interest for the period. It only takes into consideration the outstanding principal amount (payoff balance - current balance) at the time of the interest calculation.

The Lifecycle Of A Loan

Loans have the same lifecycle as other service agreements (e.g., pending start, active, pending stop). You should rarely have to manually change a loan service agreement's status. Why? Just like other service agreements, the system changes the status behind-the-scenes when various events take place. For example, loans in the pending start state are automatically activated by a system background process.

If you need to manually change a loan service agreement's status (e.g., you need to cancel a loan that was inadvertently started), you can do so from Service Agreement - Main.

Copyright © 2026, Oracle and/or its affiliates.

Finalizing And Closing A Loan

A loan SA is automatically stopped when the last bill segment is generated for the loan. If the remaining principal balance (plus accrued interest) is less than or equal to the loan payment amount, the SA status is changed to pending stop and the bill segment is marked as the closing bill segment. The SA can then be stopped by the user (manually using the SA page) or by the SA activation background process. When the customer pays the last bill, the loan SA is closed.

NOT_SUPPORTED

For more information about transitioning SAs from pending stopped, refer to Finalizing Pending Stops.

Navigating To The Loan Maintenance Page

Any of the following methods can be used to navigate to the Loan Maintenance transaction:

- A user can set up a favorite link associated with the loan transaction.
- Via any of the following menus:
  - Account context menu
  - Service agreement context menu
  - Menu

Note

Alerts and Loans. If the loan SA type is set up with an alert, the Alert Zone highlights if a loan exists for the current account. You are transferred to the SA maintenance transaction (not the loan maintenance transaction) if you drill down on the alert.

Loan Maintenance

You can use loan maintenance to view the loan's amortization schedule and financial history as well as to change the terms of a loan, including the loan amount (prior to SA activation only), interest rate, payment amount and number of payment periods.

Note

Starting Loans. You cannot start a loan from Loan - Main; you must use Start/Stop. Refer to How To Start A Loan Service Agreement for information.

Copyright © 2026, Oracle and/or its affiliates.

Open **Menu**, and then **Customer Information**, and then **Loan**, and then **Search** to maintain a loan.

Description of Page

The fields at the top of the page are informational only. The **Payoff Balance** is the total amount owed by the customer (excluding any interest that may have accrued since the last bill). The **Current Balance** is the amount currently owed by the customer. The **Principal Balance** is the total principal amount left unbilled. Refer to Payoff Balance and Current Balance for Loans for more information.

The **Loan Amount** is the amount originally loaned to the customer. The **Standard Interest Rate** is the standard interest rate defined on the loan's SA type. If the SA type allows, the standard interest rate may be overridden by a start option and/or manually. The **Next Payment Amount** is the effective periodic payment amount as of the next bill.

The **Payment(s)** shows the number of payments that were used to calculate the original loan schedule.

The **Override Interest Rate** grid is used to override the standard interest rate. The interest rate can only be modified if the loan's SA type allows it. Note that changing the interest rate changes the loan amortization schedule. To override the interest rate:

- Specify an **End Date** for the override rate in effect if there is one.
- Click + and specify the **Start Date** and the new **Interest Rate**. You can also specify the **End Date** of the new override rate, but this is not required.

The **Payment Amount** grid is used to maintain the effective periodic payment amount. Note that changing the payment amount changes the amortization schedule. To change the payment amount:

- Click + and specify the **Effective Date** and the new payment **Amount**.

> **Note**
>
> **Entering a payment amount that is too small.** If you save a payment amount that is too small to cover the periodic interest charges, this causes negative amortization. The system is not able to recalculate the amortization schedule and bill segments are produced in error until the problem is corrected.

The **History** grid shows the loan's financial history in reverse chronological order. The financial history grid displays the following information:

- The **Arrears Date** is used to compute how many days old the debt is. In the context of a loan, debt only refers to the billed amount (the amount the customer needs to pay to stay current).
- **Financial Transaction Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown.
- **Current Amount** is the effect of the FT on the service agreement's current balance.
- **Current Balance** is the amount owed by the customer at the time of the transaction.
- **Payoff Amount** is the effect of the FT on the service agreement's payoff balance. The value is grayed out if it is the same as the current amount.
- **Payoff Balance** is the total amount owed by the customer. The value is grayed out if it is the same as the current balance.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Payoff Balance And Current Balance For Loans.

The **Forecasted Schedule** displays the loan amortization schedule. The first row of the schedule shows the current cumulative interest, current cumulative principal and the current principal balance. All other rows show the future amortization schedule based on the loan details. The forecasted schedule includes the following information:

- The **Loan Period** displays the forecasted payment period dates based on the billing frequency and the payment periodicity. The Loan Period is blank for the first row, which shows current cumulative amounts.
- The **Amount** displays the projected amount of that payment. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Interest** displays the projected amount of interest in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Principal** displays the projected amount of principal in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Cumulative Interest** is the total interest amount that would theoretically be billed at that loan period.
- The **Cumulative Principal** is the total principal amount that would theoretically be billed at that loan period.
- The **Principal Balance** is the remaining principal balance that would theoretically be remaining at the loan period.

> **Note**
>
> **Forecasted schedule.** The loan schedule algorithm calculates the values in the forecasted schedule. Depending on the algorithm set up for your installation, the values may have a different meaning.

The **Renegotiate Future Payments** area allows you to change the terms of the loan. You can modify the following options:

- You can modify the **Loan Amount** only if the SA has not been activated. After the SA is activated, this field is for information only.
- The **Interest Rate** allows you to change the interest rate. When changing the interest rate, specify either the number of payment amounts or the payment amount and click Calculate to recalculate the amortization schedule.
- The **Number of Payment Periods** allows you to change the number of periods remaining in the loan's term. The **Frequency** displays the periodicity of the payments. If you change the number of payment periods, use the Calculate button to recalculate the payment amount.
- The **Payment Amount** allows you to change the amount of individual payments. If you change the payment amount, use the Calculate button to recalculate the number of periods.

Copyright © 2026, Oracle and/or its affiliates.

- The **Calculate** button recalculates the payment amount (when you specify the interest rate and the number of payment periods) or the number of payment periods (when you specify the interest rate and the payment amount) and recomputes the loan amortization schedule based on these new terms.

> **Note**
>
> **Implementing New Loan Terms.** The changes that you make using **Override Interest Rate**, **Payment Amount** and **Renegotiate Future Payments** do not take effect until you save the loan. Changes made to the **Renegotiate Future Payments** section are only valid for save after you click Calculate, and they override any other changes.
>
> **Saving Renegotiated Future Payments.** When you save a loan that has renegotiated payments, the system may add new records for the override interest rate, payment amount, or both. It uses the start date of the next loan period as the start date of the new interest rate and/or as the effective date of the new payment amount. If the loan has not yet been billed, the system updates the interest rate and/or payment amount created when the loan was started.

How To

The topics in this section describe how to perform common loan functions.

- How To Start A Loan Service Agreement
- How To Modify The Terms Of A Loan
- How To Adjust Loans
- How To Stop A Loan Service Agreement

How To Start A Loan Service Agreement

Loan service agreements are created using start/stop just like all other service agreements. The start/stop transaction has special loan functionality that allows an operator to specify the service agreement-specific loan terms described above.

Creating a loan service agreement involves the following steps:

- Select the account on Control Central then open Start/Stop.
- Enter a **Start Date** and a **Start Method** of Start a SA.
- Indicate the **CIS Division** / **SA Type** of the loan service agreement and click the **Start** button.

> **Note**
>
> If the service agreement's account's CIS Division's **Restrict to Account CIS Division** flag sis set to *Restricted*, the CIS Division is set to the account's CIS Division and cannot be changed.

Copyright © 2026, Oracle and/or its affiliates.

- After you click **Start**, the Start Confirmation page appears. When you select a loan service agreement type, the Start Confirmation page displays special fields that allow you to specify the loan's terms.
- Select a loan **Start Option**. The start option includes the adjustment type for booking the principal amount, and can also include defaults for the initial loan amount, periodic payment amount, payment periods, and interest rate. The system uses the information on the start option to populate the service agreement's default parameters. You can manually override these parameters:
  - Enter the **Loan Amount** (the amount that the customer is borrowing). Note that the label for the Total Amount To Bill (e.g., Loan Amount) may appear differently depending on how the SA type was defined.
  - The system can automatically calculate either the payment amount (based on the loan amount and the payment periods) or the payment periods (based on the loan amount and the payment amount). Enter either the **Payment Amount** or the **Payment Periods** and click **Calculate**. Note that the label for the Recurring Charge (e.g., Payment Amount) may appear differently depending on how the SA type was defined.
  - Once the loan parameters are populated, click **Start**.

The loan is created in the pending start state. The next time the SA activation process runs, it will activate the loan SA. If you want to activate the loan immediately, navigate to the service agreement page and click **Activate SA**.

> **Note**
>
> **Overriding the Interest Rate.** The **Interest Rate** is a display only field. The interest rate is taken from the value of the interest bill factor defined on the SA type or the contract value of the interest bill factor on the start option if one is defined. After starting a loan, you can manually override the interest rate using Loan - Main if the SA type allows it.
>
> **Changing The Loan Amount.** You can use loan maintenance to change the loan amount for an SA as long as the SA has not yet been activated. It may be helpful to transfer to Loan - Main , and verify the amortization schedule before activating the SA.

How To Modify The Terms Of A Loan

After a loan is active, you may want to modify its terms. You can change number of payment periods, the payment amount, and the interest rate (if allowed by the SA type). To modify the terms of a loan:

- Navigate to the Loan maintenance page.
- If you used the account context menu, the loan you want to modify may be displayed. If not, use the search dialog that appears to search for and select the loan you want to modify.
- In the **Renegotiate Future Payments** section, change the interest rate (if allowed), the number of payments, or the payment amount. Note that either the number of payments or payment amount must be blank.
- Click **Calculate**. The **Forecasted Schedule** is recalculated along with either the new payment amount or the new number of payments (depending on which was left blank).

Copyright © 2026, Oracle and/or its affiliates.

- If the renegotiated payments are satisfactory to the customer, save the changes.

When you save the changes, appropriate entries are made in the **Override Interest Rate** and **Payment Amount** grids.

> **Note**
>
> **Saving Loan Modifications.** You must save the renegotiated payments after clicking Calculate for the loan modifications to be effective.

How To Adjust Loans

You issue ad hoc adjustments if you need to change a loan's principal and/or current balance outside of the normal billing and payment functions.

An adjustment can:

- Change a loan's payoff balance.
- Change a loan's current balance (i.e., how much the customer thinks they currently owe).
- Change both the loan's payoff and current balance.

> **NOT_SUPPORTED**
>
> For details about the adjustment types to apply, refer to Adjusting Loan Amounts. For information about creating adjustments, refer to Adjustments.

How To Stop A Loan Service Agreement

Normally, a loan SA is automatically stopped when the bill segment for the last payment in the amortization schedule is generated for the loan (refer to Finalizing And Closing A Loan).

If a customer decides to payoff a loan before the term is complete, you can:

- Transfer to the SA maintenance page for the loan service agreement, specify an **End Date** for the loan, and click Save. This transitions the loan SA to pending stop.

> **Note**
>
> **End Date.** The loan SA cannot be stopped until the specified end date is reached, so do not specify an end date that is in the future unless you do not want to stop the loan until that date.

Once the loan is in pending stop and the end date has been reached, the system transitions the loan SA to stopped. On the service agreement's normal bill cycle, the system final bills the SA for any outstanding balance plus accrued interest charges.

Copyright © 2026, Oracle and/or its affiliates.

If you want to stop the loan and create the final bill immediately, complete these additional steps:

- On the loan's SA maintenance page, click the Stop SA button to transition the loan SA to stopped.
- Manually, create a bill for the loan service agreement. Remember that if you only want to generate a bill segment for a specific service agreement (e.g., the loan service agreement), transfer to the bill segment page. From there, you can generate and freeze a bill segment for a specific service agreement. When finished, return to the bill page and complete the bill.

When the customer pays the final bill, the loan SA is closed by the system.

NOT_SUPPORTED

For more information about SA states, refer to The Lifecycle Of A Service Agreement.

Non-Billed Budgets

A Non-Billed Budget (NBB) is a payment plan used to level out a customer's payments over time. Non-billed budgets are typically used when your company bills on an infrequent basis and you want to provide your customers with a mechanism to make smaller payments more frequently. For example:

- Rather than receive a bill once a quarter, a customer could set up an Non-Billed Budget to pay fortnightly
- Rather than pay two large bills a year, the customer could set up an Non-Billed Budget to pay 10 out of 12 months (sparing themselves during the holiday season)

Non-Billed Budgets can cover one or more service agreements linked to the same account. You can have multiple active Non-Billed Budgets for an account, but any given service agreement can only be covered by a single Non-Billed Budget at any point in time.

Note

Non-Billed Budgets are optional. The business processes described in this section are only relevant if your organization offers Non-Billed Budgets to its customers.

NOT_SUPPORTED

For information about setting up non-billed budget functionality for your system, refer to Defining Non-Billed Budget Options.

- The Big Picture Of Non-Billed Budgets
- Maintaining Non-Billed Budgets

Copyright © 2026, Oracle and/or its affiliates.

- How To

The Big Picture Of Non-Billed Budgets

The following topics provide background information about Non-Billed Budgets.

- Non-Billed Budget Service Agreements
- Which SAs Are Covered By A Non-Billed Budget?
- Non-Billed Budgets Versus Other Plans For Payment
- There Are Two Types Of Non-Billed Budgets
- Navigating To The Non-Billed Budget Maintenance Page
- Recommending Scheduled Payments

Non-Billed Budget Service Agreements

Non-Billed Budgets are initiated by creating a Non-Billed Budget service agreement for a customer. The Non-Billed Budget SA and its SA type maintain the details of the Non-Billed Budget, such as recommendation rules, covered SAs, scheduled payments and credit balance. Non-Billed Budgets are just like other service agreements in some ways:

- Non-Billed Budgets have a current and a payoff balance.
- If the customer doesn't pay a monitored Non-Billed Budget, credit & collections may start a collection process to encourage the customer to pay (depending upon how the debt class has been set up on your system).

Non-Billed Budgets differ from other service agreements in the following ways:

- Non-Billed Budgets have a special role of Non-Billed Budget.
- Non-Billed Budgets have scheduled payments.
- Non-Billed Budgets maintain a list of covered SAs (utility and other service agreements to which the Non-Billed Budget applies)
- Non-Billed Budgets are started and maintained on a separate transaction (refer to Non-Billed Budget - Main for more information).

Which SAs Are Covered By A Non-Billed Budget?

A Non-Billed Budget can apply towards one or more of the account's billable SAs whose SA type is flagged as Eligible for Non-Billed Budget. SAs can be added to or removed from a Non-Billed Budget at any time. Additionally, an SA can be linked to more than one Non-Billed Budget as long as the Non-Billed Budgets to which it is linked do not have overlapping start and end dates.

A list of the SAs covered by a non-billed budget is maintained with the Non-Billed Budget SA. This list is used at bill completion to determine the financial transactions that should occur.

Copyright © 2026, Oracle and/or its affiliates.

Non-Billed Budgets Versus Other Plans For Payment

Unlike a regular budget but like a payment plan, a Non-Billed Budget has one or more scheduled payments. However, unlike payment plans, bills are not produced for the scheduled payment amounts. Instead, customers are simply notified of the payment schedule and it is their responsibility to make the scheduled payments. In general, the following statements highlight the differences between Non-Billed Budgets and other payment plans.

- There is a payment schedule that is not dependent on billing, unlike budget plans.
- Non-Billed Budgets are not limited to SAs of a single debt class, unlike pay plans.
- Debt is not transferred from regular SAs to the non-billed budgets, unlike payment arrangements.

> **Warning**
>
> Non-Billed Budgets and Budget Plans! An SA should not be on a Non-Billed Budget if it is on a budget plan . A Non-Billed Budget overrides any budget functionality in most cases. However, placing an SA that is on a budget on a Non-Billed Budget may result in indeterminate results. For example, when the Non-Billed Budget is broken, the stop algorithm for the Non-Billed Budget syncs current and payoff balances for those utility SAs that were covered by the Non-Billed Budget.

There Are Two Types Of Non-Billed Budgets

There are two types of Non-Billed Budgets: monitored and unmonitored. As the terms suggest, the difference between monitored and unmonitored Non-Billed Budgets is whether they are monitored by the account debt monitor. However, this also means that monitored and unmonitored Non-Billed Budgets receive different financial treatment by the system.

In general, monitored Non-Billed Budgets have their current balances monitored by the account debt monitor while the current balances of any covered SAs are zero (and therefore don't violate any debt collection criteria). With monitored Non-Billed Budgets, customers make required prepayments that are applied towards their utility bills (until the Non-Billed Budget is stopped). They are responsible for making sure the payments on their Non-Billed Budget are up to date, but they do not have to payoff any balances on covered SAs.

Unmonitored Non-Billed Budgets do not have their current balances monitored by the account debt monitor while the current balances of any covered SAs are maintained and therefore subject to the account debt monitor. When customers are set up on an unmonitored Non-Billed Budget, the payments that they make towards the budget are optional. They are, however, financially responsible for paying the current balances of any covered SAs.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Financial Transactions For Unmonitored Non-Billed Budgets for detailed information about the financial differences between monitored and unmonitored Non-Billed Budgets.

Note

SA Type Determines Monitored Status. The SA type that you select when setting up a Non-Billed Budget determines whether it is monitored or unmonitored (i.e., it's a setting on the SA type). Your company may choose to implement only monitored non-billed budgets, only unmonitored Non-Billed Budgets, or both.

Navigating To The Non-Billed Budget Maintenance Page

Any of the following methods can be used to navigate to the Non-billed Budget Maintenance transaction:

- An alert in the Alert Zone highlights if a Non-Billed Budget exists for the current account. You are transferred to the non-billed budget transaction if you drill down on the alert.
- You can setup an order's campaign to automatically transfer a user to the Non-Billed Budget transaction whenever an order is completed for a given SA. This might be useful for campaigns that are associated with services that are typically on a Non-Billed Budget.
- A user can setup a favorite link associated with the Non-Billed Budget transaction.
- Via any of the following menus:
  - Account context menu
  - Service agreement context menu
  - Menu menu

Recommending Scheduled Payments

When creating a Non-Billed Budget, you must select a recommendation rule. The recommendation rule establishes the amount to be paid and the dates on which the payments are due. In other words, the recommendation rule creates a recommended payment schedule.

The recommendation rules that you can use are specific to your implementation. The recommendation rule that you select can have default values for its parameters. If your implementation allows, you can manually override one or more of these default values. For example, you may be able to specify the period covered by the Non-Billed Budget and the day of the month on which the payments are due.

If your implementation allows, you can manually modify the recommended payment schedule to suit the needs of a particular customer.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Designing Recommendation Rules for more information.

Maintaining Non-Billed Budgets

Non-Billed Budget service agreements are started and maintained on a different transaction than other SAs. The topics in this section explain how to start and maintain Non-Billed Budgets.

- Non-Billed Budget - Main
- Non-Billed Budget - History

Non-Billed Budget - Main

The Main page contains basic information about the Non-Billed Budget. Open **Menu**, and then **Customer Information**, and then **Non-Billed Budget**, and then **Search** to maintain this information.

Description of Page

Non-Billed Budget SA (service agreement) information and **SA ID** are displayed on each page. These values only appear after the service agreement exists on the database. The **SA ID** is a system assigned random number that stays with a service agreement for life. The **Non-Billed Budget SA** information is a concatenation of important details about the service agreement and its account.

The **Account ID** is identification of the account that is financially responsible for the service agreement.

Indicate the **CIS Division** and **SA Type**. These fields control aspects of the service agreement's behavior, such as whether the Non-Billed Budget is monitored or unmonitored. These fields are gray when the status is not Pending Start.

Note

If the service agreement’s account’s CIS Division’s **Restrict to Account CIS Division** flag sis set to *Restricted*, the CIS Division is set to the account’s CIS Division and cannot be changed.

NOT_SUPPORTED

For more information about Non-Billed Budget SA types, refer to Defining Non-Billed Budget Options.

Copyright © 2026, Oracle and/or its affiliates.

If **Sched**uled **Payment Auto Pay** is Included in Auto Pay and the account is set up for automatic payment, the scheduled payments are covered by auto pay. If Excluded from Auto Pay, the scheduled payments are excluded from auto pay if it is set up for the account.

Recommendation Rule is used to create a recommended payment amount and payment schedule. The SA type controls which recommendation rules are allowed for the Non-Billed Budget SA.

Current Balance indicates the amount the customer currently owes on the Non-Billed Budget. If the Non-Billed Budget is unmonitored, the current balance is always zero.

Payoff Balance displays the total amount the customer would owe if they wanted to close the service agreement. For Non-Billed Budgets, the payoff balance typically represents an accumulated credit that has not yet been distributed to the covered SAs (at bill completion time or when the Non-Billed Budget is stopped).

The **Start Date** defines when the financial relationship begins. The **End Date** is a display only field that indicates when the financial relationship terminates. The **Expiration Date** is the date on which the SA expires. The **Renewal Date** is the date on which the SA is scheduled to be renewed by the SA renewal background process. If the SA is not renewed by the expiration date, the stop process is initiated.

The **Recommendation Rule Parameter Values** grid specifies the parameters and their values to use for the selected recommendation rule:

- **Parameter** is the name of the parameter from the recommendation rule algorithm type.
- **Value** is the value used for the parameter. If the recommendation rule allows, you can override the default values by specifying your own.

> **Note**
>
> The recommendation rule's payment schedule algorithm type (NBPS-MON) controls the number and type of parameters. The recommendation rule controls the default values and whether they can be overridden.

The **Covered Service Agreements** grid lists the utility and other SAs that are on the Non-Billed Budget. You must create the list of covered SAs before the recommendation rule can recommend a payment amount and list of scheduled payments. The grid displays the **SA ID**, **SA Information**, **Current Balance** and **Payoff Balance** for covered SAs.

The scheduled payments grid displays the scheduled payments created by the recommendation rule (after you click the Recommend NBB button). You can manually create or modify the **Scheduled Date** and or **Scheduled Amount** of any scheduled payment in the list. The **Total of Scheduled Payments** shows the total of all scheduled payments in the list.

> **NOT_SUPPORTED**
>
> Refer to Scheduled And Actual Payments On The Non-Billed Budget for more information on what occurs when the base Non-Billed Budget scheduled payment background process processes scheduled payments that are due.

In the Action grid:

Copyright © 2026, Oracle and/or its affiliates.

- Click **Recommend** to create a recommended payment schedule based on the recommendation rule.
- Click **Renew** to manually renew the Non-Billed Budget. This executes the SA renewal algorithm for the SA type and returns new renewal and expiration dates for the SA.
- Click **Break** to manually cancel the Non-Billed Budget. This executes the break Non-Billed Budget algorithm defined for the SA type, and changes the status of the SA to pending stop.

Non-Billed Budget - History

The Financial History page displays the Non-Billed Budget's historical financial transactions. Open **Menu**, and then **Customer Information**, and then **Non-Billed Budget**, and then **Search** and navigate to the **History** page to view this information.

Description of Page

Non-Billed Budget SA (service agreement) information and **SA ID** are displayed on each page. These values only appear after the service agreement exists on the database. The SA ID is a system assigned random number that stays with a service agreement for life. The Non-Billed Budget SA information is a concatenation of important details about the service agreement and its account.

The **Account ID** identifies who is financially responsible for the service agreement.

The grid shows the Non-Billed Budget's financial history in reverse chronological order, including:

- The **Arrears Date** is used to compute how many days old the debt is. For monitored Non-Billed Budgets, debt refers to the scheduled payment amounts (i.e., the customer only owes the scheduled payment amounts up to the most recent scheduled payment date). The arrears date for a schedule payment is the scheduled payment date.
- **Financial Transaction Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown.
- **Current Amount** is the effect of the FT on the service agreement's current balance.
- **Current Balance** is the amount owed by the customer at the time of the transaction.

How To

The topics in this section describe how to perform common Non-Billed Budget functions.

- How To Start A Non-Billed Budget Service Agreement
- How To Cancel / Stop Non-Billed Budget SAs
- How To Renew Non-Billed Budgets
- How To Set Up Automatic Payment For A Non-Billed Budget

How To Start A Non-Billed Budget Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

To set up a new Non-Billed Budget:

- Navigate to the Non-Billed Budget maintenance page.
- If the account is not already specified, select the **Account** for which you want to start the Non-Billed Budget.
- Define the **CIS Division** and **SA Type** to use for the Non-Billed Budget service agreement.
- If necessary, change the **Start Date** (the current date is defaulted).
- Select the **Recommendation Rule** to use and modify any of the default parameters to meet the needs of the customer.
- Verify the list of **Covered Service Agreements**. All the account's SAs that are eligible for Non-Billed Budget are listed. Click to remove unwanted SAs from the list.
- Click **Recommend**.
- On the pop-up dialog, specify the **Schedule Start Date** and click **Recommend**.
- Make any necessary changes to the expiration date, renewal date and scheduled payments grid and click Save.

How To Cancel / Stop Non-Billed Budget SAs

To stop a Non-Billed Budget:

- Navigate to the Non-Billed Budget maintenance page.
- If the account with the Non-Billed Budget you want to stop is not specified, select the **Account** of the Non-Billed Budget you want to stop.
- If the account has more than one Non-Billed Budget SA, select the Non-Billed Budget you want to stop.
- Click **Break**.

How To Renew Non-Billed Budgets

Most Non-Billed Budgets are renewed automatically by a background process. However, you may manually renew Non-Billed Budgets that are eligible for renewal if desired.

To renew a Non-Billed Budget:

- Navigate to the Non-Billed Budget maintenance page.
- If the account with the Non-Billed Budget you want to renew is not specified, select the **account** of the Non-Billed Budget you want to renew.
- If the account has more than one Non-Billed Budget SA, select the Non-Billed Budget you want to renew.
- If desired, change the recommendation rule or any of the parameters.
- Click **Renew**.

The new expiration and renewal dates are calculated and the new payment schedule is appended to the list of scheduled payments.

Copyright © 2026, Oracle and/or its affiliates.

How To Set Up Automatic Payment For A Non-Billed Budget

If a customer wants to pay their non-billed budget scheduled payments automatically:

- Set up the account for automatic payment (as described under How To Set Up A Customer To Pay Automatically).
- Navigate to the Non-Billed Budget Maintenance page and select the Non-Billed Budget for the account.
- Set the **Sched**uled **Payment Auto Pay** field to indicate that the scheduled payments are Included in Auto Pay.

Case Management

> **Note**
>
> **Separate module.** The Case Management functionality is associated with a separate Case Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

Your organization can use case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,
- the processing of a market message in a deregulated environment

> **Note**
>
> **Your implementation controls how your cases behave.** Since virtually all aspects of case management functionality are controlled by your implementation, you can use it to handle a myriad of business requirements. Refer to Defining Case Management Options for the details.

- Background Topics
- Maintaining Cases

Copyright © 2026, Oracle and/or its affiliates.

Background Topics

The topics in this section provide background information about case management functionality.

- Case Type
- Creating Cases
- Log Information
- Alerts
- To Do's and Cases
- Reports

Case Type

A case is controlled by the case type that it references. Some aspects of the case that are controlled by the case type include:

- The lifecycle of the case
- The applicability of a person, account, and/or premise on the case
- Which fields are required during the various stages of a case's life
- Which users can access cases of a given type
- Additional processing that should take place when a case enters or leaves a given state

Refer to Case Type Controls Everything for a description of the business rules that you control when you set up your case types.

Creating Cases

To create a case, open the case management page in add mode, select the appropriate case type, fill in the other necessary fields, and click save. Oracle Utilities Customer Care and Billing also includes scripts that can help create cases. Refer to Scripts and Cases for more information.

You can also develop plug-ins and user exits to so that the system will create cases when specific events take place. For example, you could set the system to create a case when an umbrella agreement is ready for renewal or when a quote is created.

Please refer to Maintaining Cases for a detailed description of the case page and field descriptions.

Log Information

A case log contains an entry for every recorded event during the lifecycle of the case. There are two types of log entries:

Copyright © 2026, Oracle and/or its affiliates.

- **Automatic entries.** The system automatically creates an entry in the log when a case is created or there is a status change. Users cannot modify or delete these log entries.
- **Manual entries.** Users can add manual entries to record significant events at their discretion.

Refer to Case - Main for the page used to view the log.

Refer to Case Statistics for a Given Status for information about a report that uses the log to provide productivity statistics.

Alerts

The alert zone will highlight if the person / account / premise in context have cases in a state that is configured as "alertable" if you plug-in the appropriate algorithm on the installation record. Refer to Alert Info Is Controlled By An Installation Plug-In for the details.

To Do's and Cases

You can configure the system to inform users of cases that require their attention. Refer to To Do's and Cases for the details.

Reports

Several reports summarize case information. Refer to the Reports chapter for the details.

Maintaining Cases

Your organization can use the case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,
- the processing of a market message in a deregulated environment

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Since an implementation can configure virtually every aspect of a case, the look and feel of the case management page are dynamic based on how case types have been configured.

- Case - Main
- Case - Log

Case - Main

Open the case management page by selecting **Menu**, and then **Customer Information**, and then **Case**, and then **Search**.

Main Information

Case Info contains a concatenation of important information about the case. **Case ID** is the system-assigned unique identifier of the case. These values only appear after the case is added to the database.

> **Note**
>
> **Formatting may be performed by a plug-in.** Refer to Case Info May Be Formatted By An Algorithm for more information.

Case Type defines the type of case. This field is protected after the case has been added. Refer to Case Type Controls Everything for more information.

> **Note**
>
> Available case types can be restricted by CIS Division. The restriction of case type by CIS division is dynamic. It is based on the account in context, the premise in context, or the current user. The restriction of case type is applied using these rules in sequence.
>
> - If there is an account in context and that account has a CIS Division, that CIS Division is used to restrict the case types that appear.
> - If there is a premise in context and that premise has a CIS Division, that CIS Division is used to restrict the case types that appear. Note: If there are both an account and premise in context, the account CIS Division is used.
> - If there is no account or premise in context with a CIS Division and the current user is associated with CIS Division(s) those CIS Division(s) are used to restrict the case types that appear. See Users and CIS Division for more information.

Status shows the current state of the case. Refer to Case Lifecycle for more information.

Copyright © 2026, Oracle and/or its affiliates.

If the case has any To Do entries associated with it, a hyperlink will appear summarizing the number and status of the entries. You can select the link to view these entries. Refer to To Do's and Cases for more information.

Date / Time Opened shows when the case was created. If the case is in a final state, the **Date / Time Closed** will also be shown. Refer to One Initial State and Multiple Final States for more information.

Script references the business process assistant script that can help you work on the case. You can click on the hyperlink to execute this script. This field is hidden if no script is associated with this case's current Status. Refer to Scripts and Cases for more information.

The **Actions** section contains buttons you can use to action the case. This area is suppressed if no actions can be performed for the case given its current status. A button will be gray if you do not have security rights to the action. Refer to Buttons Are Used To Transition A Case for more information.

When the user adds a new case or changes the state of a case manually the system attempts to auto-transition the case to subsequent statuses as necessary. If auto-transition rules apply to the new state (and to subsequent ones) they would be executed right away. If the case has been automatically transitioned to its current state the following message area appears right after the action area. If an error is encountered in the process a corresponding error message is displayed in the Auto-transition Message Area. Refer to Automatic Transition Rules for more information.

> **Note**
>
> The Auto-transition Message Area only appears when a case has just been automatically transitioned to its current state online. It only remains visible until the data for the case is refreshed.

Use the **Comments** area to describe anything interesting or unusual about the case.

Person references the person associated with this case. This field is suppressed if the Case Type indicates that a person is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Person currently displayed in the Sidebar.

Account references the account associated with this case. This field is suppressed if the Case Type indicates that an account is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Account currently displayed in the Sidebar.

Premise references the premise associated with this case. This field is suppressed if the Case Type indicates that a premise is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Premise currently displayed in the Sidebar.

> **Note**
>
> The case type controls the relationship validation among person, account, and premise.

Copyright © 2026, Oracle and/or its affiliates.

Responsible User references the user who has overall responsibility for the case. This field is suppressed if the Case Type indicates that a responsible user is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final).

The **Contact Information** area is used to define how to contact the individual who initiated the case. These fields are suppressed if contact information is not allowed on cases of this type. The fields cannot be modified if the case is in a closed status. The following information can be defined.

Contact Person references the person who should be contacted. This defaults to the Person defined above.

Preferred Contact Method indicates how the person prefers to be contacted. The following options are available:

- Not Applicable. If the person doesn't want to be contacted (or the contact method is unusual), use this option. You might want to consider describing the situation in the Contact Instructions.
- Person Contact. If this method is selected, select a person contact related to the Person link to the customer contact. The customer contact must contain a Person to use this method.
- Postal. If this method is selected, the Contact Person's address appears. Refer to Where Does The Address Come From for a description of which address the system displays.

The following methods appear when the Legacy Person Phone and Email **Option Type** on the Customer Information Options Feature Configuration has been enabled.

- Email. If this method is selected, the Contact Person's email address appears adjacent.
- Phone. If this method is selected, the Contact Person's phone number appears.
- Fax. If this method is selected, the Contact Person's fax number appears (the system knows which of a person's phone numbers is a fax number by the phone type).
- SMS. If this method is selected, the Contact Person's short message service (SMS) information displays.

> **Note**
>
> While on a case, users can add a person contact by selecting Add Person Contact from a Person Context menu. This will open a portal-based page that allows a person contact to be added. This process will not lose context or information that has been added to the case. The new person contact can be used within the case.

Contact Instructions indicates any special instructions indicated for when or how to return a call.

Callback Phone Type, Callback Phone Number, and Extension indicate the actual phone number and phone type that should be used to call.

Characteristics

The **Characteristics** grid contains fields that contain important information about the case. This grid is suppressed if the Case Type does not use characteristics. This grid is protected if the case is closed (i.e., if its status is defined as being final).

You can only choose **Characteristic Types** defined as permissible for this case type. Refer to Additional Information for more information. If you attempt to use an **Action** to transition a case

Copyright © 2026, Oracle and/or its affiliates.

into a status that requires additional information, a pop-up will prompt you for the additional fields. Refer to Required Fields Before A Case Enters A State for more information.

Case - Log

To view the log entries associated with a case, open by selecting **Menu**, and then **Customer Information**, and then **Case**, and then **Search** and navigate to the **Log** page.

Description of Page

The **Log** grid displays log entries (in reverse chronological order) that audit the progress of the case. Please note the following about the entries in this grid:

- The system automatically creates log entries when a case is created, every time its status changes, or whenever an error occurs during the transition of a case. Algorithms plugged into a case type state can also be configured to create log entries. You cannot modify or delete the log entries.
- You can manually add a log entry by pressing the + button and enter the **Details**. You cannot modify or delete this information after saving it.
- The following information appears in the grid:
- **Log Date/Time** contains the date and time the log entry was created.
- **Details** contains user-specified or system-generated information about the log entry.
- **Related Object** is populated on log entries that were created to record the creation of some other object. For example, if a case plug-in creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
- **Log User** contains the user who caused the log entry to be created.
- **Log Type** indicates how the log entry was created. The possible values are:
- Created. Only one entry per case will be of this type. This entry is automatically created by the system when the case is created.
- User Details. A user creates this type of entry manually.
- Status Transition. This type of entry is created by the system each time the case's status is changed (as a result of pressing one of the Action buttons).
- System. An algorithm on the case type state may be designed/configured to create this type of entry when it does something auditable. For example, an algorithm that creates a customer contact could insert a log entry to show that it did this (it will also populate the FK to the customer contact, which will appear in the **Related Object** column).
- Transition Error. The background status transition process creates an entry of this type if an error occurs when automatically updating a case to a subsequent status (as directed by either an Enter Status algorithm or an Auto-Transition algorithm). An entry of this type lists the case states involved with the failed transition, and the algorithm in which the error was generated.
- Exception. The background status transition process creates an entry of this type under the same circumstances as a Transition Error. A log entry of this type indicates why the background transition failed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is CASE_LOG_TYPE_FLG.

Premise Management

> **Note**
>
> **Separate module.** The Premise Management functionality is associated with a separate Premise Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

The premise management functionality facilitates the following:

- Grouping premises together under a single parent premise
- Quick & efficient setup and maintenance of apartment complexes and high-rises
- A single view of properties for a landlord, including service status
- Ability to change the landlord or service status for a group of premises in one transaction
- And more...

The topics in this section describe premise management functionality.

- Overview
- Premise Management Page

Overview

The topics in this section provide background information about premise management functionality.

- Define Premise Hierarchy
- Manage Premises for a Landlord
- Manage Groups of Premises

Define Premise Hierarchy

Premise management allows you to define premise hierarchy. For example, you could also have the notion of

Copyright © 2026, Oracle and/or its affiliates.

- A building
- Multiple floors
- Multiple apartments per floor

To define this hierarchy you must create a premise for each level in the hierarchy and for each "child" premise, indicate its appropriate parent premise. Once you have set up your premise hierarchies, you may view the hierarchies on trees on the premise page and on the control central premise tree.

In addition, the system provides the following alerts:

- When viewing a parent premise, an alert indicates how many child premises exist
- When viewing a child premise, an alert indicates that it is linked to a parent premise.

Manage Premises for a Landlord

The premise management page provides the ability to view all the premises for the landlord along with information about the accounts linked to each premise.

You may use this page to assign, change or remove the landlord agreement for multiple premises in a single action. For example if a landlord sells their properties to another person, this page facilitates updating all premises with one action rather than changing the landlord for each premise separately. For example, if the landlord sells only a portion of their properties.

This page may also be used to start or stop service for one or more premises for the landlord. For example, a landlord plans to sell a high-rise apartment complex with 100 apartments. Imagine that 8 of these apartments are vacant and the landlord's account is currently paying for any usage to these apartments. In addition to changing the landlord agreement for all 100 apartments to the new landlord, the service for the vacant apartments should be transferred to the new landlord. A user can go to premise management, use the search criteria to view all premises for this landlord in a given apartment complex whose service is currently being paid for by the landlord and start service for these premise for the new landlord. (Using the back-to-back start/stop functionality, service will be stopped for the old landlord.)

Manage Groups of Premises

The powerful search criteria on the premise management page allows you to view different groups of premises based on any combination of account information, landlord information, parent premise address information and general geographic type information. You may also define search criteria such as "premises **not** linked to account 1234567890" or "premises linked to **any** landlord" or "premises **not** linked to any parent premise". For example, you could display all premises in a given latitude / longitude that are not linked to any parent premise.

Once your search results display the desired list of premises, you could perform mass updates such as assigning a parent premise or starting service. Following are examples of situations where this page would be helpful:

- The developer of a high-rise condominium complex is ready to start service for a group of condos. A user uses the premise management search criteria to display the premises linked to a give parent premise (representing the condo complex) that are not linked to any account. The user selects the premises in this list that the developer would like to start and starts service for all premises at once.

Copyright © 2026, Oracle and/or its affiliates.

- There is a shopping mall in your service territory and each store in an individual premise. You may decide to define a parent premise for the entire shopping mall and link each store's premise to the parent. Once the parent premise is defined, a user navigates to the premise management page and displays the individual premises based on a similar address or based on a similar geographic type. The user then assigns the parent premise for all the individual premises using a single action.

Premise Management Page

This page allows you to display a list of premises using a combination of search criteria. In addition, you can perform actions on one or more of the resulting premises, including:

- Assigning a parent premise to one or more premises
- Removing the link to a parent premise from one or more premises
- Assigning a landlord to one or more premises
- Removing the link to a landlord from one or more premises
- Start service for one or more premises
- Stop service for one or more premises

Open this page using **Menu**, and then **Customer Information**, and then **Premise Management.**

Description of Page

The top half of the page is where you enter the criteria used to search for premises.

> **Note**
>
> **Multiple search criteria may be specified.** You can search for premises using a combination of search criteria. For example, if you enter both a **Parent Premise Filter** of Linked To Any Premise and an **Account Filter** of Not Linked to Any Account; only those premises that have a parent premise but are not linked to an account will be displayed.
>
> **At least one positive criterion is required**. You may not define your search criteria to only use filter values that start with "Not Linked...". At least one "positive" filter value is required.

The following table describes each of the different search methods.

Search Method	Description

Copyright © 2026, Oracle and/or its affiliates.

Landlord Filter	Use this filter to narrow down your search based on landlord. Enter one of the following values: Not Applicable Linked To This Landlord - for this option, you must enter a **Landlord Agreement** Not Linked To This Landlord - for this option, you must enter a **Landlord Agreement** Linked To Any Landlord Not Linked To Any Landlord. A filter value of Not Applicable defaults.
Parent Premise Filter	Use this filter to narrow down your search based on parent premise. Enter one of the following values: Not Applicable Linked To This Parent Premise - for this option, you must enter a **Premise ID** for the parent premise Not Linked To This Parent Premise - for this option, you must enter a **Premise ID** for the parent premise Linked To Any Parent Premise Not Linked To Any Parent Premise. A filter value of Not Applicable defaults.
Account Filter	Use this filter to narrow down your search based on account. Enter one of the following values: Not Applicable Linked To This Account - for this option, you must enter an **Account ID** Not Linked To This Account - for this option, you must enter an **Account ID** Linked To Any Account Not Linked To Any Account. A filter value of Not Applicable defaults.
Premise Filter	Use this filter to narrow down your search based on account. Enter one of the following values: Located At This Address - for this option, you must enter address constituents Located At This Geographic Type / Value - for this option, you must enter a **Geo Type** and **Value** Not Applicable Show This Specific Premise - for this option, you must enter a **Premise ID** A filter value of Not Applicable defaults.

The **Select All** / **Clear All** buttons are used to select premises if you plan on issuing any of the mass update actions at the bottom of the page. Refer to the description of the "mass update" actions below for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **50 premises at a time.** Clicking **Select All** selects the first 50 premises in the grid. If more than 50 premises exist, you must select them in batches.

The grid that follows contains the premises that match your search criteria. The following information appears in the grid:

- Select box. Use this checkbox to select premises for mass update actions.
- The **Premise Information** column shows information about each premise. Click the hyperlink to transfer to the Premise page where you can update information about the premise in question.
- The **Account Information** column shows information about all accounts linked to the premise, if any. Only accounts with non-canceled, non-closed service agreements linked to this premise are displayed. The column displays the name of the customer's main person followed by a list of status values for any of the account's service agreements linked to this premise. For example, you may see Customer A (pending stop), Customer B (pending start). If multiple service agreements linked to the account & premise have the same status, the status is only displayed once.
- The **Parent Premise Information** column displays information about the premise's parent premise, if one exists.
- The **Landlord Information** column displays the landlord agreement description for the premise's landlord, if there is one.

This transaction has sophisticated logic that can be used to perform "mass updates" to the premises that appear in the grid using the buttons at the bottom of the page. The buttons are enabled if you select at least one row from the premise grid. The following points describe these mass update actions:

- Start Service
- Stop Service
- Assign Parent Premise
- Remove Parent Premise
- Assign Landlord
- Remove Landlord

Start Service

The **Start Service** button is used to start service for one or more premises. When clicked, the **Start SA Groups Confirmation** window opens.

> **Note**
>
> **Similar to Start/Stop**. This page is similar to the Start SA Confirmation page that appears on start/stop. The differences are to accommodate starting multiple premises.

Copyright © 2026, Oracle and/or its affiliates.

In order to start service for the selected premises, select the **Account ID**.

The **Start Date** is the date on which the service is scheduled to start.

Use **Start Requested By** to indicate the name of the person who requested the start.

Check **Include Child Premises** if all the child premises in the hierarchy below the selected premises should also be started for this account.

Click **Show SA Groups** to populate the remainder of the page.

The **Service Points** section summarizes the service points that exist for all the chosen premises. A row is displayed for each different **SP Type** along with a **Count** of the number of service points of that type. By default, all rows are checked, assuming that you want to start service for all service points (and the SA Groups To Start scroll below displays proposed SAs to start for each SP type). You may uncheck any SP Type row if you do not wish to start service for that SP type. Click **Show SA Groups** to rebuild the SA Groups To Start information.

> **Warning**
>
> This page displays summary information about the service agreements and service points that are being turned on. It is recommended that you start groups of premises that have similar service point configurations. If you attempt to start service for many premises, each with different service point configurations, the start logic may be very complicated to verify on this page.

The **SA Groups To Start** scroll contains an entry for every service agreement that will be created if you click the **Start** button. If you do not want a service agreement to be created, click the - button or uncheck **Start this SA Group**. If you would like to start service agreements that are not SP-based service agreements (for example a deposit or a connection charge), click the + button to add a new SA group. The system assumes that you want to create a new service agreement for each selected premise.

Turn on the **Start this SA Group** switch if you want service agreements of this type to be created. If this is the first customer at the selected premises, this switch is checked for all service agreements that appear in the scroll. For each premise that was linked to a previous customer, this switch is on only for those types of service agreements used by the previous customer(s). (We assume the new customers only want the services used by the previous customers). Service agreements that are possible, but were not used by the previous customers, appear in the scroll with this switch turned off.

If Start This SA Group is checked, the system displays a message indicating the number of service agreements (for the number of service points) that will be created if you choose Start.

For a description of the remaining fields, refer to Start/Stop - Start Confirmation.

Click **Start**. Standard start logic is followed when clicking start.

> **Note**
>
> The start button is disabled if changes are made to fields on the page that may impact the SA Groups To Start collection. You must click **Show SA Groups** to rebuild the SA groups and enable the start button.

Copyright © 2026, Oracle and/or its affiliates.

Stop Service

The **Stop Service** button is used to stop service for one or more premises.

> **Note**
>
> that this functionality stops service for any account linked to the selected premises. If you want to stop service at the selected premises for only certain accounts, use the account filter criteria to narrow down the selection of premises prior to clicking Stop Service.

When clicked, the **Stop SA Group Confirmation** window opens.

In order to stop service for the selected premises, enter a **Stop Date**.

Enter a **Restrict to Account** if you want to stop only service agreements linked to this account. If not specified, all active services agreements for the premise are stopped regardless of the account.

Use **Stop Requested By** to indicate the name of the person who requested the stop.

Check **Include Child Premises** if the child premises in the hierarchy below the selected premises should also be stopped.

Click **Show SA Groups** to populate the remainder of the page.

The system builds a collection of service agreements currently linked to the service points for the selected premises. The grid contains the **CIS Division** and **SA Type** for each SA "group" and displays the number of service agreements in the **SA Group Summary**.

By default, the system checks each SA group assuming that you want to stop service for all SAs. You may uncheck any of the SA groups to not stop service for those SAs.

Click **Stop**. Standard stop logic is followed when clicking stop.

> **Note**
>
> The stop button is disabled if changes are made to fields on the page that may impact the SA collection. You must click **Show SA Groups** to rebuild the SA collection and enable the stop button.

Assign Parent Premise

The **Assign Parent Premise** button is used to assign a parent premise to one or more premises. When clicked, the **Assign Parent Premise** window opens.

Specify the **Parent Premise ID** and click **Assign**.

Copyright © 2026, Oracle and/or its affiliates.

Remove Parent Premise

The **Remove Parent Premise** button is used to remove the parent premise from one or more premises. When clicked, the parent premise ID of all selected premises is reset.

Assign Landlord

The **Assign Landlord** button is used to assign a landlord to one or more premises. When clicked, the **Assign Landlord** window opens.

Specify the **Landlord** agreement ID.

Check **Include Child Premises** if the child premises in the hierarchy below the selected premises should also be assigned this landlord.

Click **Assign**.

Remove Landlord

The **Remove Landlord** button is used to remove the landlord agreement from one or more premises. When clicked, the **Remove Landlord** window opens.

Check **Include Child Premises** if all the child premises in the hierarchy below the selected premises should also have their landlord agreement removed.

Click **Remove**.

Umbrella Agreement Management

Your organization may use umbrella agreement functionality to manage many situations, including (but not limited to) the following:

- Creating a "negotiated contract" with a large customer with many sites.
- Creating a "negotiated contract" with all customers in a common geographic area, for example a specific city or an apartment complex
- Grouping and managing the proposal service agreements created during the quoting process
- Managing the renewal process for special contracts
- Applying override rate terms to one or more service agreements for a given effective period
- more...

Copyright © 2026, Oracle and/or its affiliates.

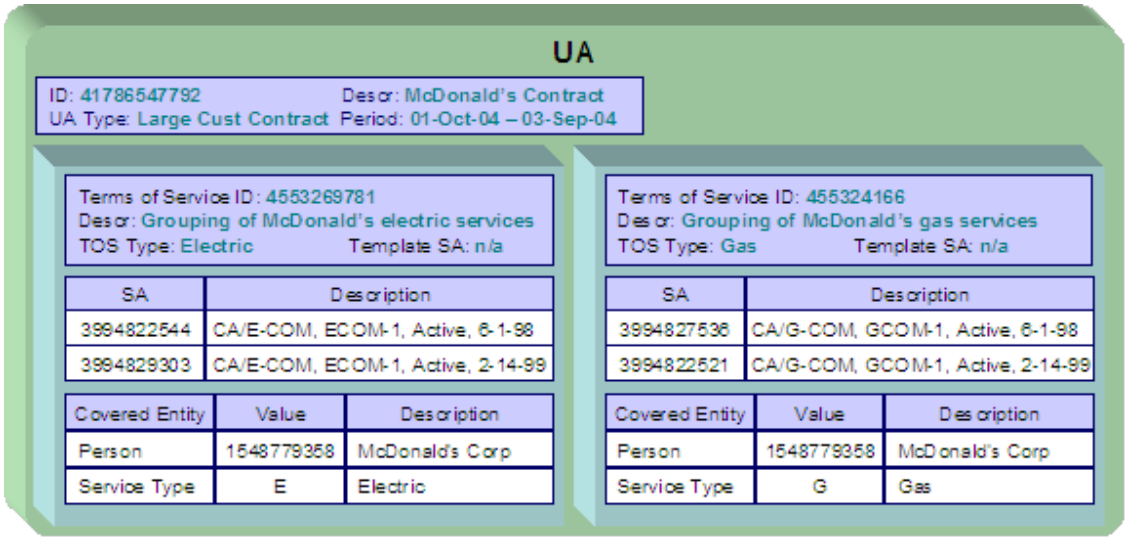
> **Note**
>
> **Separate module.** The umbrella agreement functionality is part of the Contract Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

- Overview Of Umbrella Agreement & Terms Of Service
- Umbrella Agreements & Proposals
- Additional Umbrella Agreement Topics
- Umbrella Agreement Maintenance
- Terms Of Service Maintenance

Overview Of Umbrella Agreement & Terms Of Service

An umbrella agreement is used to group together terms for a contract. It contains high level information about the contract such as the effective dates, the renewal date, if applicable and an account management group, if applicable. You may also define characteristics and contacts for the umbrella agreement.

The terms of the umbrella agreement are defined using one or more terms of service (TOS) records. For each TOS record, you must link one or more service agreements to indicate the services covered under the TOS.



The Terms of Service may point to a 'template service agreement' as a way to specify the rating terms and conditions to be used by all service agreements linked to the TOS. If a TOS does not define a template SA, then the TOS is used largely as a grouping mechanism rather than a billing mechanism.

Umbrella agreements and their terms of service records are used to support specific business scenarios that your organization supports.

- TOS Covered Entities
- Rate Terms Overridden By Terms Of Service

Copyright © 2026, Oracle and/or its affiliates.

- Umbrella Agreement Renewal

TOS Covered Entities

Your implementation may configure a Terms of Service type to require that entities "covered" by the terms of service record be referenced on the TOS. For example, imagine that you define an umbrella agreement to provide special discounts to all the McDonalds franchises in your service area. When creating the terms of service record for the umbrella agreement, you may link the person record representing the McDonalds corporate entity as a covered entity for the TOS.

> **Note**
>
> that there is no base product functionality that uses this collection. However, your implementation may use this information in an algorithm, for example, it could be used in a campaign / package eligibility algorithm.

Refer to Terms Of Service Covered Entities for information about defining covered entities for a terms of service type.

Rate Terms Overridden By Terms Of Service

Your organization may require rate information to be temporarily overridden for a service agreement that is linked to a TOS. In order to do this, the TOS record must reference a template SA. Refer to Overriding Rate Terms for information about defining your Terms of Service types to support template service agreements.

The topics in this section further describe the possible rate term overrides that may occur when a service agreement is linked to a terms of service record.

- Rate Schedule Override by TOS
- Override Of Other Rate Terms By TOS

Rate Schedule Override by TOS

If a service agreement is linked to a terms of service, you may configure the system to use the rate schedule from the terms of service record's template SA rather than the one from the customer's service agreement. The customer's SA type controls this option.

> **NOT_SUPPORTED**
>
> Refer to Configure SA Types for Umbrella Agreements for more information.

Copyright © 2026, Oracle and/or its affiliates.

If a service agreement is linked to a terms of service record, the rate information tree on the service agreement's rate info tab tells the user whether the rate from the service agreement or from the template SA on the TOS is currently applicable.

If a given service agreement's rate schedule is being taken from the terms of service record's template SA, then the override rate schedule is used not just in bill calculations, but for any logic that requires the service agreement's rate schedule. For example, one of the recommended budget amount algorithms may estimate consumption and apply the service agreement's rate. In this case, the rate used by the algorithm is the one linked to the terms of service record's template SA.

Override Of Other Rate Terms By TOS

There are other terms associated with a rate that may change when linking a service agreement to a terms of service record that has a template SA. These include contract riders, contract terms and tax exemptions.

When applying the rate to create the customer's bill segment, if a bill factor on any of the calculation rules indicates that it is eligible for tax exemptions, contract terms or contract riders, a flag on the bill factor indicates where the system should look for the appropriate value. The Terms of Service Usage flag tells the system to find applicable information either on the TOS's template SA only, on the customer's SA only or it should first check the TOS, then check the customer's SA.

NOT_SUPPORTED

Refer to Configure Bill Factors for Umbrella Agreements for more information.

Note

Independent of rate source. The flag values on the bill factors are independent of the rate source setting for the SA type. For example, your implementation may configure the SA type's rate source to indicate that rate should be taken from the TOS template SA. However, if a bill factor is eligible for tax exemptions, your implementation may configure the bill factor to indicate that the exemption information should be taken from the customer's SA, not the template SA.

Umbrella Agreement Renewal

Although umbrella agreement records require an end date, your organization probably allows for renewal of agreements. A renewal may involve full re-negotiation of all terms, or may be as simple as extending the current terms. If umbrella agreements of a given type are eligible to be renewed, you configure the UA type to indicate how to calculate the renewal date and which algorithm to execute when the renewal date is reached.

- Umbrella Agreement Renewal Background Process

Copyright © 2026, Oracle and/or its affiliates.

Umbrella Agreement Renewal Background Process

The UA renewal background process (UARENEW) finds complete umbrella agreements whose renewal date has arrived and executes the renewal algorithm for the associated UA type.

> **Note**
>
> It is the responsibility of the renewal algorithm to update the renewal date as per the business practice. Refer to Renewable Umbrella Agreements for more information.

Umbrella Agreements & Proposals

Umbrella agreements and their terms of service records may reference proposal service agreements. This functionality allows you to create the full contract for your customer during the quoting stage.

Additional Umbrella Agreement Topics

The following topics describe additional functionality related to umbrella agreements.

- Umbrella Agreement Alerts
- Orders And Umbrella Agreements
- Umbrella Agreement Reports

Umbrella Agreement Alerts

The alert zone will highlight if any of the account's SAs are linked to an effective umbrella agreement (via a non-canceled terms of service record).

Orders And Umbrella Agreements

When starting new service for a given account and premise, you may know ahead of time that the new service should be linked to an existing umbrella agreement. For example, perhaps all the service in a specific low income housing area qualifies for a special negotiated contract for the next year.

The system provides algorithms to allow you to add a new service agreement to an existing terms of service record when using the Order transaction to start service.

Refer to Configure A Campaign To Link An SA To A Terms Of Service for more information.

Copyright © 2026, Oracle and/or its affiliates.

Umbrella Agreement Reports

The system supplies a Report providing umbrella agreement information for one or more umbrella agreements.

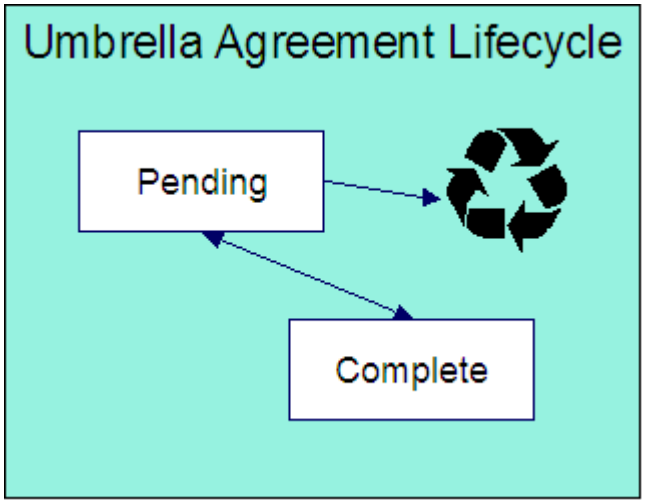
Umbrella Agreement Maintenance

Umbrella agreements help you manage your contracts by allowing you to group together service agreements for the contract with one or more terms of service records for the umbrella agreement.

- Lifecycle Of An Umbrella Agreement
- Umbrella Agreement - Main
- Umbrella Agreement - Contacts
- Umbrella Agreement - Characteristics

Lifecycle Of An Umbrella Agreement

The following diagram illustrates the lifecycle of an umbrella agreement.



Pending The umbrella agreement is created in pending status.

You may delete a pending umbrella agreement. Deleting a UA also deletes its terms of service records.

Complete Only complete umbrella agreements are considered effective. You may transition a UA to complete if there is at least one terms of service record linked to it and if none of its TOS records are in pending status.

Under unusual circumstances, a complete umbrella agreement may be changed back to pending to correct errors.

Copyright © 2026, Oracle and/or its affiliates.

Umbrella Agreement - Main

Open this page using **Menu**, and then **Umbrella Agreement Management**, and then **Umbrella Agreement**, and then **Search.**

Description of Page

Umbrella Agreement Info contains a concatenation of important information about the umbrella agreement. **Umbrella Agreement ID** is the system-assigned unique identifier of the umbrella agreement. These values only appear after the UA is added to the database.

Umbrella Agreement Status shows the current state of the UA. The buttons adjacent to the status allow you to transition the UA's status.

- Click **Complete** to change the status of a pending umbrella agreement to complete.
- Click **Reopen** to change the status of a complete umbrella agreement back to pending.

> **NOT_SUPPORTED**
>
> Refer to Lifecycle of an Umbrella Agreement for more information.

Enter a **Description** for the umbrella agreement.

The **Umbrella Agreement Type** defines the type of umbrella agreement. Refer to Setting Up Umbrella Agreement Types for more information.

The **Start Date** and **End Date** define the effective dates of the contract. If the UA type is renewable, the **Renewal Date** indicates the date the umbrella agreement is processed by the renewal background process.

You may assign an **Account Management Group** to this umbrella agreement. The AMG controls the roles assigned to To Do entries associated with the umbrella agreement. Refer to Setting Up Account Management Groups for more information.

Use the **Comments** area to describe anything interesting or unusual about the umbrella agreement.

This bottom of this page contains a tree that shows the various objects linked to the umbrella agreement. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Umbrella Agreement - Contacts

The contacts page defines persons associated with the umbrella agreement. Open this page using **Menu**, and then **Umbrella Agreement Management**, and then **Umbrella Agreement**, and then **Search** and navigate to the **Contacts** page.

Description of Page

The contacts collection allows you to define persons that are associated with the umbrella agreement. For each contact, define the **Person ID** for the person record that contains the

Copyright © 2026, Oracle and/or its affiliates.

demographic information for the contact. For each contact, use the **Comments** to define information about the contact.

Umbrella Agreement - Characteristics

The characteristics page contains information that describes miscellaneous information about the umbrella agreement. Open this page using **Menu**, and then **Umbrella Agreement Management**, and then **Umbrella Agreement**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined on umbrella agreement's umbrella agreement type .

The following fields display:

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

> **Note**
>
> **Default Note.** An umbrella agreement's characteristics default from the umbrella agreement type.

Terms Of Service Maintenance

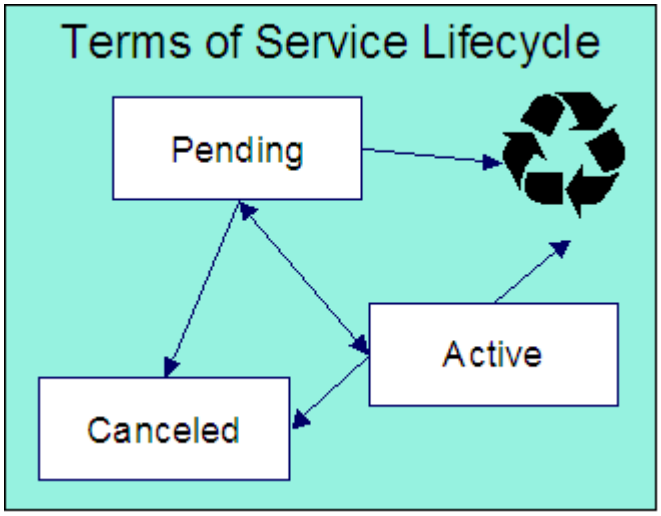
A terms of service (TOS) record defines one or more service agreements participating in a given umbrella agreement. The TOS record may also reference a template SA that could be used to override rate terms for the participating service agreements.

- Lifecycle Of Terms Of Service
- Terms Of Service - Main
- Terms Of Service - SAs

Lifecycle Of Terms Of Service

The following diagram illustrates the lifecycle of a terms of service record.

Copyright © 2026, Oracle and/or its affiliates.



Pending The terms of service record is created in pending status. You may delete a pending TOS. If you want to keep a record of the TOS, you may cancel it rather than deleting it.

Active Only active terms of service records are considered effective. Change the status to active when you have entered all the desired information for this TOS. You may delete an active terms of service if the umbrella agreement is still pending. (Otherwise you must cancel the TOS.)

Canceled You may cancel a pending or active terms of service if it should no longer be used. Canceling a terms of service record unlinks all its service agreements.

Terms Of Service - Main

Open this page using **Menu**, and then **Umbrella Agreement Management**, and then **Terms of Service**, and then **Search.**

Description of Page

Terms of Service Info contains a concatenation of important information about the terms of service record. **Terms of Service ID** is the system-assigned unique identifier of the terms of service record. These values only appear after the TOS is added to the database.

The **Umbrella Agreement ID** defines the umbrella agreement to which this terms of service is linked.

Status shows the current state of the terms of service record. The buttons adjacent to the status allow you to transition the TOS record's status.

- Click **Activate** to change the status of a pending terms of service record to active.
- Click **Pend** to change the status of an active terms of service record back to pending.
- Click **Cancel** to change the status of a pending or active terms of service record to canceled.

NOT_SUPPORTED

Refer to Lifecycle Of Terms of Service for more information.

Copyright © 2026, Oracle and/or its affiliates.

Enter a **Description** for the terms of service record.

The **Terms of Service Type** defines the type of terms of service record. Refer to Setting Up Terms of Service Types for more information.

Use the **Comments** area to describe anything interesting or unusual about the umbrella agreement.

The **Characteristics** collection contains information that describes miscellaneous information about the terms of service record.

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined on the terms of service record's terms of service type .

The following fields display:

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

> **Note**
>
> **Default Note.** A terms of service record's characteristics default from the terms of service type.

The **Covered Entities** collection allows you to define specific entities that are "covered" by this terms of service record. Refer to TOS Covered Entities for more information.

> **Note**
>
> **Characteristic Types.** You can only choose covered entity characteristic types defined on the terms of service record's terms of service type.

The following fields display:

Covered Entity Characteristic Type Indicate the type of characteristic used to define this covered entity type.

Characteristic Value Indicate the value of the characteristic used to define the covered entity value.

> **Note**
>
> **Default Note.** A terms of service record's covered entity types default from the terms of service type.

Copyright © 2026, Oracle and/or its affiliates.

Terms Of Service - SAs

This page is used to maintain the service agreements linked to the terms of service record.

Open this page using **Menu**, and then **Umbrella Agreement Management**, and then **Terms of Service**, and then **Search** and navigate to the **SAs** page.

Description of Page

Terms of Service Info contains a concatenation of important information about the terms of service record. **Terms of Service ID** is the system-assigned unique identifier of the terms of service record.

The filters control the service agreements that appear in the grid. The following points describe the various options:

- Use the **SA Filter** to define the types of service agreements that appear in the grid. The following options are available:
  - Account. Use this option to restrict service agreements to those linked to a given **Account ID**.
  - Address. Use this option to restrict service agreements to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - All. Use this option if you do not wish to restrict service agreements based on service agreement attributes.
  - All Proposals. Use this option to restrict service agreements to only proposal service agreements.
  - All Real SAs. Use this option to restrict service agreements to only "real" service agreements (i.e., not proposal SAs).
  - Geographic Type. Use this option to restrict service agreements to those linked to service points associated with a given **Geographic Type** and **Geographic Value**.
  - SA Type. Use this option to restrict service agreements to those linked to a given **CIS Division** and **SA Type**.
  - Specific SA. Use this option to reference a specific **SA ID**.
- Use **Status Filter** to restrict the service agreements based on their status. Status values for both proposal service agreements and regular service agreements are included in the filter. The following options are available:
  - All. This option shows all service agreements, including proposal service agreements, regardless of status.
  - Refer to Service Agreement Lifecycle for the various status values for real service agreements.
  - Refer to Proposal SA State Transition for more information about status values for proposal service agreements.

Don't forget to click the search button after changing the filters.

The grid that follows contains the **Service Agreement**s that match your search criteria along with its **Account Info**.

Use the "+" to link new service agreements to the terms of service record.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Link SA Using Order**. You may also link a service agreement to a TOS through orders if you have configured an appropriate campaign and package.

Note you may remove a service agreement from the TOS using the delete button as long as the change has not been committed to the database. As soon as the SA's link to the TOS has been committed to the database, you must use the **Remove** action to unlink the service agreement.

The **Select All** / **Clear All** buttons are used to select service agreements if you plan on issuing the mass update action **Remove** at the bottom of the page. Clicking **Remove** removes the service agreement from the terms of service record.

Overdue Financial Obligations

The system periodically monitors how much your customers owe to ensure they haven't violated your overdue rules. When a violation is detected, the system creates an overdue process. The overdue process contains the events meant to prod the customer to pay (e.g., letters, disconnect field activities, To Do entries, write-off outstanding debt, etc.). This section describes how to manage your overdue (i.e., collections) processing.

> **Note**
>
> The overdue processing module has been designed to collect on virtually anything from an unpaid bill to an unmatched financial transaction. You tell the system what you collect on by configuring the various overdue processing control tables. In this release, the base-package is delivered with the algorithms that support collecting on overdue bills. If your organization practices balance-forward accounting (i.e., collection is based on overdue service agreement balances), you will not use this functionality. Rather, you will use the functionality described under Credit and Collections.

- Overdue Processing Background Information
- Overdue Process Maintenance
- Cut Process Maintenance
- Payment Arrangement for Bills

Overdue Processing Background Information

In the section, The Big Picture Of Overdue Processing, we describe how overdue processing works and how to set up the control tables that automate most functions. You will find that many of your questions regarding when and how overdue and cut processes are created and canceled are described in this section.

Copyright © 2026, Oracle and/or its affiliates.

Overdue Process Maintenance

An overdue process is a series of events (e.g., letters, To Do entries) meant to encourage an account to pay its delinquent debt. The topics in this section describe the pages on which overdue processes are maintained.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Overdue Processes for background information.

- Overdue Process - Main
- Overdue Process - Events
- Overdue Process - Log
- How To Perform Common Overdue Process Functions

Overdue Process - Main

The Main page contains core overdue process information. Open this page using **Menu**, and then **Credit & Collection.**, and then **Overdue Process**, and then **Search**

Description of Page

> **Note**
>
> Refer to How To Perform Common Overdue Process Functions for more instructions describing how to use this page.

Overdue Process contains a concatenation of important information about the process. **Overdue Process ID** is the system-assigned unique identifier of the process. These values only appear after the overdue process is added to the database.

> **Note**
>
> **Summary information may be overridden.** Refer to Overdue Process Information Is Overridable for how your implementation can override the summary information that appears throughout the system.

Account ID identifies the overdue process's account.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multiple overdue processes may be linked to an account.** It's important to be aware that it's possible for multiple, active overdue processes to be linked to an account. The Alert zone will contain a summary of the account's overdue processes.

Status defines the state of the overdue process. The following values may exist:

- Active, Events Pending. Overdue processes are initially created in this state. An overdue process remains in this state until there are no Pending or Waiting events.
- Inactive, Canceled by User. An overdue process will exist in this state when it's been manually canceled by a user. Navigate to the **Log** tab to see when this happened and who did it.
- Inactive, Canceled by System. An overdue process will exist in this state when it's been canceled by the system (typically because the overdue obligations were satisfied). Navigate to the **Log** tab to see when this happened.
- Inactive, Completed. An overdue process will exist in this state when the system completes its last event. Navigate to the **Log** tab to see when this happened.

The **Cancel** button appears if the overdue process is Active, Events Pending and the process's **Overdue Process Template** has a Cancel Logic plug-in. Click this button to cancel the process.

The **Trigger Events** button appears if the overdue process is Active, Events Pending. Click this button to activate all pending events that are ready for activation.

> **Note**
>
> **If you can't wait for the Overdue / Cut Event Manager to run.** The Overdue / Cut Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button on the **Main** tab to activate the process's events (that are ready for activation).

Overdue Process Template defines the template that was used to create the overdue process's events. You can override these events on the **Events** page. This field is unprotected when the process is Active and all events are in either the Pending or Canceled states.

> **Note**
>
> **Changing the template.** If you change the template, the system will delete the events and replace them with the new template's events.

Start Date/Time defines the start date/time of the overdue process. This field is protected after the overdue process is saved on the database. This field is used to derive the trigger date of some overdue events when the process is first created. Subsequent changes to the start date change the trigger dates on the events accordingly.

Copyright © 2026, Oracle and/or its affiliates.

Inactive Date/Time is the date and time that the overdue process became Inactive. This field is hidden if the process is Active. You can see more details about when and how the inactivation occurred by navigating to the **Log** tab.

Base Date is reserved for future releases. This field is protected when the overdue process is Inactive.

Enter any **Comments** about the overdue process. This field is protected when the overdue process is Inactive.

The **Collecting On** grid holds the overdue objects. For example, if this process manages overdue bills, the grid contains a list of the bills being collected on by this process. This grid is unprotected when the process is Active . The following points describe the columns that appear in the grid:

- Press the + button to add a new overdue object to the process.
- Press the - button to remove an overdue object to the process.
- The next column contains the unique identifier of the overdue object. For example, if this process manages overdue bills, this column will contain the bill ID. The type of object that's specified in this column is control by the overdue process template.
- **Original Amount** contains the original amount of the overdue object.
- **Unpaid Amount** contains the unpaid amount of the overdue object.

The Trees at the bottom of the page shows a variety of information about the overdue process including its events and cut processes. You can click on hyperlinked tree nodes to navigate to the page on which the related object is maintained.

Overdue Process - Events

This page contains the activities that are performed to persuade the customer to pay the outstanding debt. Open this page using **Menu**, and then **Credit & Collection**, and then **Overdue Process**, and then **Search** and navigate to the **Events** page.

Description of Page

> **Note**
>
> Refer to How To Perform Common Overdue Process Functions for more instructions describing how to use this page.

Refer to the first tab for a description of **Overdue Process** and **Overdue Process ID**.

The **Overdue Events** scroll contains the process's overdue events. Refer to Overdue Processes Are Created From Templates for information about how the system defaults a process's events from its template.

- Note, all information in the scroll is protected if the event's Status is Complete or Canceled.
- **Event Sequence** is the unique identifier of the event.
- **Status** defines the state of the event.
- The **Cancel** button appears if the event is Pending or Waiting, and the event's **Overdue Event Type** has a Cancel Logic plug-in. Click this button to cancel the event.

Copyright © 2026, Oracle and/or its affiliates.

- The **Overdue Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).
- The remaining fields work in unison:
  - If **Dependent on Other Events** is turned on, the event can only be triggered after the events specified in the **Event Dependencies** grid are all Complete or Canceled. You use the **Days After** field to define when this event is activated. For example, if you enter 5, this event will be activated 5 days after all of the dependent events are completed / cancelled. If you enter 0, this event will be activated immediately after the dependent events are completed / cancelled. Refer to Calendar vs Work Days for a descriptions of how days are counted.

If all dependent events are already Completed or Canceled , you cannot specify **Days After**. Rather, you must enter the desired activation date in the **Trigger Date** field.

- If **Dependent on Other Events** is turned off, enter the desired activation date in the **Trigger Date** field. If this field is off, the **Days After** field and the **Event Dependencies** grid are hidden.

> **Note**
>
> **If you can't wait for the Overdue / Cut Event Manager to run.** The Overdue / Cut Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button to activate the process's events (that are ready for activation).

Overdue Process - Log

This page contains log entries that highlight significant events in the process's life. Open this page using **Menu**, and then **Credit & Collection**, and then **Overdue Process**, and then **Search** and navigate to the **Log** page.

Description of Page

Please note the following about the entries in this grid:

- The system automatically creates log entries when significant events occur. Refer to Overdue Log for more information.
- You can manually add a log entry to an Active process by pressing the + button and enter the **Details**. You cannot modify or delete this information after saving it.
- The following information appears in the grid:
  - **Date/Time** contains the date and time the log entry was created.
  - **Details** contain the user-specified or system-generated information about the log entry.
  - **Relate Object** is populated on log entries that were created to record the creation of some other object. For example, if an overdue event creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
  - **Related Process / Event** appears if the log entry was created by an overdue event, a cut process or a cut event. It references the unique identifier of the process / event.

Copyright © 2026, Oracle and/or its affiliates.

- **Log User** contains the user who caused the log entry to be created.
- **Log Type** indicates how the log entry was created. The possible values are:
  * User. A user manually created this entry.
  * System. This system created this entry.

How To Perform Common Overdue Process Functions

The topics in this section describe how to perform common overdue process maintenance functions. Refer to The Big Picture Of Overdue Processing for high-level information about overdue processing.

- How To Create An Overdue Process
- How To Change Overdue Events
- How To Cancel An Overdue Process

How To Create An Overdue Process

99.9% of all overdue processes are created by the Overdue Monitor and require no human intervention. The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new overdue process.
- After the account is populated on Control Central, choose the **Overdue Process +** option on the account context menu to transfer to the overdue process transaction in add mode for the account.
- After the Overdue Process - Main page appears, specify the appropriate **Overdue Process Template**. The template is used to default the process's events (the template also controls numerous business rules, for example, when it will be canceled, how it will be canceled, etc). Refer to How To Change Overdue Events for a description of how you can override these events.
- Enter the overdue object(s) in the **Collecting On** grid. You must define at least one object.
- Save the overdue process.

How To Change Overdue Events

When an overdue process is first created, it has one or more overdue events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when an overdue process is initiated are defined on the overdue process's overdue process template. The following points describe how to add / change / delete events on an overdue process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the overdue process whose events need to be changed.

Copyright © 2026, Oracle and/or its affiliates.

- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. Note, an account's Active overdue processes can be selected from the Alerts zone.
- To add a new event, transfer to the Overdue Process - Events tab and press the + button in the **Overdue Events** scroll. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Status** of Pending.
  - Choose the desired **Overdue Event Type**.
- If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then:
  - specify the sequences of the dependent events in **Event Dependencies** and
  - specify the how many days after the completion of the last dependent event that the new event should be triggered.
  - If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, click on the event in the tree on Overdue Process - Main. This will transfer you to the **Events** tab where you can click the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
- To change an existing event, click on the event in the tree on Overdue Process - Main. This will transfer you to the **Events** tab where you can make the desired changes.
- After all desired changes have been made, save the overdue process.

How To Cancel An Overdue Process

The system will cancel an overdue process when the cancel criteria defined on the overdue process are satisfied. The following points describe how to manually cancel an overdue process.

- Use **Control Central** to choose the account with the overdue process to be cancelled.
- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. A list of all overdue processes associated with the account appears. If only one overdue process exists, it is automatically selected for you. Note, an account's Active overdue processes can be selected from the Alerts zone.
- On the **Main** tab, click Cancel and answer any prompts.

Cut Process Maintenance

A cut process contains the activities to stop a service agreement (in the hopes that lack of service will inspire the customer to pay). It should be noted that a separate cut process is

Copyright © 2026, Oracle and/or its affiliates.

created for each service agreement. The topics in this section describe the pages on which cut processes are maintained.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Cut Processes for background information.

- Cut Process - Main
- Cut Process - Events
- How To Perform Common Cut Process Functions

Cut Process - Main

The Main page contains core cut process information. Open this page using **Menu**, and then **Credit & Collections**, and then **Cut Process**, and then **Search.**

Description of Page

> **Note**
>
> Refer to How To Perform Common Cut Process Functions for more instructions describing how to use this page.

Cut Process contains a concatenation of important information about the process. **Cut Process ID** is the system-assigned unique identifier of the process. These values only appear after the cut process is added to the database.

> **Note**
>
> **Summary information may be overridden.** Refer to Cut Process Information Is Overridable for how your implementation can override the summary information that appears throughout the system.

Overdue Process and **Overdue Event** define overdue event that created the process. Refer to Cut Processes Exist Under An Overdue Process for more information.

Service Agreement identifies the service agreement being cut.

Premise identifies the service agreement's characteristic premise (if any).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multiple cut processes may be linked to an overdue process / event.** It's important to be aware that it's possible for multiple, active cut processes to be linked to the overdue event. You can see all cut processes associated with an overdue process on the tree of Overdue Process - Main. The Alert zone will contain a summary of the account's cut processes.

Status defines the state of the cut process. The following values may exist:

- Active, Events Pending. Cut processes are initially created in this state. A cut process remains in this state until there are no Pending or Waiting events.
- Inactive, Canceled by User. A cut process will exist in this state when it's been manually canceled by a user. Navigate to the overdue process's Log tab to see when this happened and who did it.
- Inactive, Canceled by System. A cut process will exist in this state when it's been canceled by the system (typically because the overdue obligations on the overdue process were satisfied). Navigate to the **Log** tab to see when this happened.
- Inactive, Completed. A cut process will exist in this state when the system completes its last event. Navigate to the overdue process's Log tab to see when this happened.

The **Cancel** button appears if the cut process is Active, Events Pending and the process's **Cut Process Template** has a Cancel Logic plug-in. Click this button to cancel the process.

The **Trigger Events** button appears if the cut process is Active, Events Pending. Click this button to activate all pending events that are ready for activation.

> **Note**
>
> **If you can't wait for the Overdue / Cut Event Manager to run.** The Overdue / Cut Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button to activate the process's events (that are ready for activation).

Cut Process Template defines the template that was used to create the cut process's events. You can override these events on the **Events** page. This field is unprotected when the process is Active and all events are in either the Pending or Canceled states.

> **Note**
>
> **Changing the template.** If you change the template, the system will delete the events and replace them with the new template's events.

Start Date/Time defines the start date/time of the cut process. This field is protected after the cut process is saved on the database. This field is used to derive the trigger date of some cut events when the process is first created. If you need to subsequently change the trigger date on the events, navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

Copyright © 2026, Oracle and/or its affiliates.

- Change the **Cut Process Template** to something (anything). Doing this causes the **Start Date/Time** to become unprotected.
- Specify the base date of the cut events in the newly unprotected **Start Date/Time**.
- Change the **Cut Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Start Date/Time**.

Inactive Date/Time is the date and time that the cut process became Inactive. This field is hidden if the process is Active. You can see more details about when and how the inactivation occurred by navigating to the overdue process's Log tab.

Enter any **Comments** about the cut process. This field is protected when the cut process is Inactive .

The tree at the bottom of the page shows a variety of information about the cut process including its events. You can click on hyperlinked tree nodes to navigate to the page on which the related object is maintained.

Cut Process - Events

This page contains the activities that are performed to cut the service agreement. Open this page using **Menu**, and then **Credit & Collection**, and then **Cut Process**, and then **Search** and navigate to the **Events** page.

Description of Page

> **Note**
>
> Refer to How To Perform Common Cut Process Functions for more instructions describing how to use this page.

Refer to the first tab for a description of **Cut Process** and **Cut Process ID**.

The **Cut Events** scroll contains the process's cut events. Refer to Cut Processes Are Created From Templates for information about how the system defaults a process's events from its template.

- Note, all information in the scroll is protected if the event's Status is Complete or Canceled.
- **Event Sequence** is the unique identifier of the event.
- **Status** defines the state of the event.
- The **Cancel** button appears if the event is Pending or Waiting, and the event's **Cut Event Type** has a Cancel Logic plug-in. Click this button to cancel the event.
- The **Cut Event Type** defines the event's activity (e.g., Email sent, a To Do entry is generated, a letter is sent).
- The remaining fields work in unison:
  - If **Dependent on Other Events** is turned on, the event can only be triggered after the events specified in the **Event Dependencies** grid are all Complete or Canceled. You use the **Days After** field to define when this event is activated. For example, if you enter 5, this event will be activated 5 days after all of the dependent events are completed / cancelled. If you enter 0 (zero), this event will be activated immediately

Copyright © 2026, Oracle and/or its affiliates.

after the dependent events are completed / cancelled. Refer to Calendar vs Work Days for a descriptions of how days are counted.

If all dependent events are already Completed or Canceled , you cannot specify **Days After**. Rather, you must enter the desired activation date in the **Trigger Date** field.

- If **Dependent on Other Events** is turned off, enter the desired activation date in the **Trigger Date** field. If this field is off, the **Days After** field and the **Event Dependencies** grid are hidden.

> **Note**
>
> **If you can't wait for the Overdue / Cut Event Manager to run.** The Overdue / Cut Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button on the **Main** tab to activate the process's events (that are ready for activation).

How To Perform Common Cut Process Functions

The topics in this section describe how to perform common cut process maintenance functions. Refer to The Big Picture Of Cut Processing for high-level information about cut processing.

- How To Create A Cut Process
- How To Change Cut Events
- How To Cancel A Cut Process

How To Create A Cut Process

Cut processes are created when an overdue event is activated. You cannot create an ad hoc cut process.

How To Change Cut Events

When a cut process is first created, it has one or more cut events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when a cut process is initiated are defined on the cut process's cut process template. The following points describe how to add / change / delete events on a cut process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the cut process whose events need to be changed.
- After the account is populated on Control Central, choose the **Cut Process** option on the account or service agreement context menus to transfer to the cut process transaction in update mode. Note, an account's Active cut processes can be selected from the Credit and Collections zone.
- To add a new event, transfer to the Cut Process - Events tab and press the + button in the **Cut Events** scroll. At this point, the event has not been added to the database; rather, it

Copyright © 2026, Oracle and/or its affiliates.

just exists in memory. Before you add the event to the database, you must specify the following information:

  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Status** of Pending.
  - Choose the desired **Cut Event Type**.
- If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then:
  - specify the sequences of the dependent events in **Event Dependencies** and
  - specify the how many days after the completion of the last dependent event that the new event should be triggered.
- If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, click on the event in the tree on Cut Process - Main. This will transfer you to the **Events** tab where you can click the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
- To change an existing event, click on the event in the tree on Cut Process - Main. This will transfer you to the **Events** tab where you can make the desired changes.
- After all desired changes have been made, save the cut process.

How To Cancel A Cut Process

The system will cancel a cut process when the cancel criteria defined on the overdue process are satisfied. The following points describe how to manually cancel a cut process.

- Use **Control Central** to choose the account with the cut process to be cancelled.
- After the account is populated on Control Central, choose the **Cut Process** option on the account or service agreement context menu to transfer to the cut process transaction in update mode for the account. A list of all cut processes associated with the account appears. If only one cut process exists, it is automatically selected for you. Note, an account's Active cut processes can be selected from the Credit and Collections zone.
- On the **Main** tab, click Cancel and answer any prompts.

Payment Arrangement for Bills

A payment arrangement is an agreement with a customer to payoff severely overdue debt in installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

Organizations that practice open-item accounting who collect on overdue bills may allow customers to pay off their overdue bills in installments by setting up a payment arrangement for one or more bills. This transaction is used to maintain bill-oriented payment arrangements.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Bill-Oriented Payment Arrangements for background information.

Open this page using **Menu**, and then **Credit & Collection**, and then **Payment Arrangement for Bills**, and then **Search** page.

Description of Page

The **SA** (service agreement) **Info** is a concatenation of important details about the payment arrangement service agreement (PA SA). The **SA ID** is the unique identifier of the PA SA. These values only appear after the PA SA exists on the database.

The **Current Balance** that appears beneath **SA Info** contains the PA SA's current balance. When you initially set up a PA SA this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **SA Info** contains the PA SA's payoff balance. The payoff balance is the total amount to be paid off over the PA SA's life. It is only non-zero after delinquent funds have been transferred to the PA SA. This balance is only displayed when it differs from the **Current Balance**.

NOT_SUPPORTED

Refer to Current Amount versus Payoff Amount for more information.

Account ID is the account associated with the PA SA.

The occurrences of **Current** and **Payoff Balance** beneath **Account** are the account's respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Unpaid Bills** grid contains a row for each of the account's unpaid bills.

Adjacent to each Unpaid Bill is its **Unpaid Amount** and a check box used to select the bill. Every selected bill's unpaid amount will be transferred to the PA SA when you click the **Create** button.

Warning

The system automatically indicates that all unpaid debt should be transferred to the payment arrangement. It does this by checking each bill by default.

Total Candidates is the sum of ALL unpaid bills (not just the selected bills).

New Payoff Balance contains the PA SA's payoff balance. This is the total amount of debt that will be paid off over the lifetime of the PA SA. This value equals the current payoff balance on the PA SA plus all selected debt from the **Candidate SAs** scroll.

The next fields work in unison.

Copyright © 2026, Oracle and/or its affiliates.

- If you want to define the number of future bills that the customer has to payoff the selected unpaid amount, enter the number of **Installments**. When you tab out of the field, the system sets the **Arrange Amount** equal to **New Payoff Balance** / **Installments**.

> **Note**
>
> **Arrange Amount may not add up.** The **Arrange Amount** is rounded up so that if you multiply the number of installments by the calculated installment amount, the result will be greater than the PA SA's payoff amount. Don't worry - only the actual dollar amount of the PA SA's total debt will be billed by the system. The last installment will be slightly lower than the other installments.

- If you want to define the amount charged on future bills to payoff the unpaid amount, enter the **Arrange Amount**. For example, if the customer agrees to pay off their delinquent debt by paying an additional $50 on each bill, you'd enter "50" in this field. Note - If you enter a dollar amount in this field and tab out, the system will calculate the number of installments and display the number in the **Installments** field.

Define the **CIS Division** and **SA Type** of the payment arrangement service agreement.

> **Note**
>
> **Payment arrangement SA types.** There is a field on SA type called Special Role. Only those SA types with a role of Payment Arrangement may be selected.

Clicking the **Create** (or **Change**)button causes the system to perform the following:

- It creates / updates a payment arrangement service agreement (PA SA). It sets the installment amount on this service agreement equal to the installment amount specified on the page.
- It transfers unpaid debt from the selected bills to the PA SA. The adjustment type used to transfer these funds is defined on the PA SA's SA type.
- It links the "transfer from" adjustments to the unpaid financial transactions on the bills. This will cause the unpaid FT's to be linked to Balanced match events. This means that any Active Overdue Processes for the bills should Inactivate (i.e., cancel) because the bills will no longer be consider unpaid. Refer to How Are Overdue Processes Cancelled for more information..
- It reduces the current balance on the PA SA by the amount of debt transferred to the PA SA. Why? Because when delinquent debt is transferred to the PA SA, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to pay off the debt in installments (and therefore they don't currently owe anything). The adjustment type used to transfer these funds is defined on the PA SA's SA type.

Clicking **Break** causes the PA SA to become broken. Please refer to the note below for what transpires when a PA is broken.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Break logic is in a plug-in.** Please be aware that the logic that is executed when a payment arrangement is broken exists in a plug-in algorithm (plugged-in on the account's Collection Class Overdue Rules). The base package algorithm performs many functions (refer to Breaking A Bill Oriented Payment Arrangement for the details). The bottom line is that when a PA SA is broken, the originating bills will have their debt reinstated and this will again be subject to the Overdue Monitor. Because this logic is in a plug-in, you can develop alternative logic and plug it in if the base package logic is not satisfactory.

Clicking **Cancel** causes the PA SA to become canceled. Cancellation should be used when you want to "logically delete" a PA SA because it shouldn't have been created.

The logic described above for breaking a payment arrangement is executed when a user cancels a payment arrangement; the only difference is that the PA SA is marked with a different characteristic type / value than when it is broken.

- How To Set Up A Payment Arrangement
- How To Add Additional Debt To A Payment Arrangement
- How To Break A Payment Arrangement
- How To Cancel A Payment Arrangement

How To Set Up A Payment Arrangement

To set up a new payment arrangement:

- Open Payment Arrangement for Bills in add mode.
- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills**).
- Enter the number of **Installments** or the **Arrange Amount**.
- Define the **CIS Division** and **SA Type** of the new payment arrangement service agreement.
- Click the **Create** button.

How To Add Additional Debt To A Payment Arrangement

To transfer additional debt to an existing payment arrangement:

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills** amounts).
- Enter the number of **Installments** or the **Arrange Amount**.

Copyright © 2026, Oracle and/or its affiliates.

- Click the **Change** button.

> **Note**
>
> **After transfer.** After transferring the funds to the payment arrangement service agreement, confirm the account's payoff and current balances make sense (the payoff balance is the total amount of debt that the account will eventually have to payoff, the current balance is how much they currently owe. The account's payoff and current balances are displayed in the page's second section. If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

How To Break A Payment Arrangement

You would break a payment arrangement when a customer doesn't make the agreed payments. Typically, your overdue processes will break a payment arrangement behind-the-scenes if a customer doesn't pay. Refer to Breaking A Bill Oriented Payment Arrangement for information about how the system monitors payment arrangements.

If you want to manually break a payment arrangement,

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.
- Click the **Break** button.

How To Cancel A Payment Arrangement

You would cancel a payment arrangement if you made a mistake and never intended to set up the payment arrangement in the first place. To cancel a payment arrangement

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.
- Click the **Cancel** button.

Rebate Claims

The Rebate Claims feature allows you to search, view, and maintain rebate claims submitted as part of a conservation program.

> **NOT_SUPPORTED**
>
> For more information about conservation programs, see Conservation Programs.

Copyright © 2026, Oracle and/or its affiliates.

- Rebate Claim Query Portal
- Rebate Claim Portal

Rebate Claim Query Portal

You use the Rebate Claim Query Portal to search for rebate claims. Use the following procedure to search for rebate claims:

- Select **Menu**, and then **Customer Information**, and then **Rebate Claim**, and then **Search** to open the Rebate Claim Query portal.
- Enter search criteria in the Rebate Claim Search zone and click **Refresh**. You can search by:
  - Customer Name
  - Account ID
  - Conservation Program
  - Start and End Dates
- A list of rebate claims matching the search criteria appears, displaying the following for each:
  - Claim Date
  - Rebate Claim (Status, Claim Date, Conservation Program, and Total Amount)
  - Account (Name, Customer Class, Current Balance, and Account ID)
  - Total Rebate Amount
- To view a rebate claim, click the **Rebate Claim** link. The selected rebate claim opens in the Rebate Claim portal.

Rebate Claim Portal

You use the Rebate Claim portal to view, edit, and maintain rebate claims. To add a rebate claim, open **Menu**, and then **Customer Information**, and then **Rebate Claim**, and then **Search.**

The topics in this section describe the base-package zones that appear on the Rebate Claim portal.

- Rebate Claim Zone
- Adding Rebate Claims
- Rebate Claim Line Zone
- Rebate Claim Log Zone

Rebate Claim Zone

You use the Rebate Claim zone to view the details of rebate claims. The Rebate Claim zone opens when you first create a new rebate claim, or when you select a rebate claim on the

Copyright © 2026, Oracle and/or its affiliates.

Rebate Claim Search zone. The Rebate Claim zone displays the following information about the selected rebate claim:

- Basic Information: information about the claim, including:
  - Conservation Program
  - Status (Pending, Issue Detected, Approval in Progress, Receipt Verification, Complete, or Canceled)
  - Rebate Claim number
  - Claim Date
  - Total Rebate Amount
- Customer Details: details about the customer who submitted the claim, including:
  - Person
  - Account
  - Premise
  - Mailing Address
- Contact Details: contact information for customer who submitted the claim, including:
  - Name
  - Phone Number
  - Fax Number
  - Email Address
- Premise Information: information about the physical premise of the customer who submitted the claim, including:
  - Premise Property Type
  - Residential Property Type
  - Building Vintage
- Payment Release Authorization Details: details about the person to whom the rebate payment should be made, including:
  - Authorized By
  - Authorization Date
  - Payee Name
  - Phone Number
  - Mailing Address
- Issue Details: issues related to the rebate claim, including manually submitted issues as well as those detected through the Verify action.
- Approval Details: details related to users assigned to approve the rebate claim, including:
  - Current Approver
- Reject Details: details related to why a claim was rejected, including:
  - Reject Reason
  - Date Rejected
  - Rejected By

Copyright © 2026, Oracle and/or its affiliates.

To view a different rebate claim, select the rebate claim from the **Select a Rebate Claim** drop-down list and click **Go.**

- Rebate Claim Status
- Rebate Claim Actions
- Editing Rebate Claims
- Verifying Rebate Claims
- Approving Rebate Claims
- Rejecting Rebate Claims
- Noting Issues Related to a Rebate Claim
- Canceling Rebate Claims

Rebate Claim Status

Rebate claims are assigned a status to indicate the current state of the claim within the system. Valid statuses include:

- **Pending** indicates the claim is pending. This is the initial state of a rebate claim when first created.
- **Issue Detected** indicates that one or more issues have been detected with the claim. Issues can include those manually created as well as issues detected through the Verify action.
- **Approval in Progress** indicates the claim is awaiting approval.
- **Receipt Verification** indicates that receipts associated with the rebate claim's claim lines need verification.
- **Complete** indicates the claim was approved and is complete.
- **Rejected** indicates the claim was rejected.
- **Canceled** indicates the claim has been canceled.

Rebate Claim Actions

You can perform a number of actions on a rebate claim, including:

- **Edit:** Used to edit the details of the rebate claim.
- **Verify:** Used to verify the supplied details of the rebate claim, and change its status to either Approval in Progress (if the claim passes the verification) or Issue Detected (if any issues are detected). This helps ensure the rebate claim has been properly set up.
- **Approve:** Used to approve a submitted rebate claim, and change its status to Complete or Receipt Verification (if receipts had not been previously verified).
- **Reject:** Used to reject a submitted rebate claim, and change its status to Complete.
- **Issue Detected:** Used to note an issue related to the rebate claim, and change its status to Issue Detected.
- **Cancel:** Cancels a rebate claim, changing its status to Canceled.

Copyright © 2026, Oracle and/or its affiliates.

The actions available are based on the current status of the rebate claim. The table below summarizes the actions available at each status.

Status	Valid Action
Pending	Edit, Verify, Issue Detected, Cancel
Issue Detected	Edit, Verify, Cancel
Approval in Progress	Approve, Reject, Issue Detected
Receipt Verification	Issue Detected
Completed	None
Rejected	Issue Detected
Canceled	None

Editing Rebate Claims

Use the following procedure to edit a rebate claim:

- Click **Edit.**
- Edit the details of the rebate claim as needed.
- Click **Save.**

Verifying Rebate Claims

To verify a rebate claim, click **Verify.**

Approving Rebate Claims

To approve a rebate claim, click **Approve.**

Rejecting Rebate Claims

Use the following procedure to reject a rebate claim:

- Click **Reject.**
- Select the **Reject Reason** on the Reject Rebate Claim dialog.
- Click **OK.** To close the dialog without rejecting the claim, click **Cancel.**

Noting Issues Related to a Rebate Claim

Use the following procedure to note an issue related to a rebate claim:

- Click **Issue Detected.**
- Enter details about the issue in the Comments field of the Issue Comments dialog.

Copyright © 2026, Oracle and/or its affiliates.

- Click **OK.** To close the dialog without saving the comment, click **Cancel.**

Canceling Rebate Claims

Use the following procedure to cancel a rebate claim:

- Click **Cancel.**
- Select the **Cancel Reason** on the Cancel Rebate Claim dialog.
- Click **OK.** To close the dialog without cancelling the rebate claim, click **Cancel.**

Adding Rebate Claims

You can also add rebate claims from the **Rebate Claim Query** portal. Use the following procedure to add a rebate claim:

- Select **Menu**, then **Customer Information**, then **Rebate Claim**, then **Add** or click **Add** on the title bar of the **Rebate Claim Search** zone. The user is navigated to **Control Central** to select the customer filling the rebate claim. Click **Continue** once the correct account is in context.
- Select the Conservation Program associated with the rebate claim and click **OK.**
- Complete the Rebate Claim details, including:
  - Claim Date for the rebate claim
  - Customer Details
  - Contact Details
  - Premise Information
  - Payment Release Authorization Details
- Click **Save**. The rebate claim is displayed on the Rebate Claim portal.
- Edit the claim as needed in the Rebate Claim zone.
- Add claim lines to the claim using the Rebate Claim Line zone.

Rebate Claim Line Zone

You use the Rebate Claim Line zone to add, view, and edit rebate claim lines associated with a rebate claim. A rebate claim can have one or more claim lines, one for each item for which a rebate is requested. For example, if a customer purchases new energy efficient appliances (such as a washing machine and dryer), the rebate claim they submit would have a rebate claim line for each appliance. The Rebate Claim Line zone displays the following details for each claim line:

- Rebate Definition
- Manufacturer/Model
- Rebate Amount
- Icons to Edit, Delete, and Verify claim lines
- Adding Rebate Claim Lines

Copyright © 2026, Oracle and/or its affiliates.

- Deleting Rebate Claim Lines
- Editing Rebate Claim Lines
- Verifying Rebate Claim Lines

Adding Rebate Claim Lines

Use the following procedure to add a rebate claim line to a rebate claim:

- Click **Add Rebate Claim Line** in the title bar of the Rebate Claim Line zone.
- Select the **Service Agreement** associated with the claim line from the dropdown list.
- Select the **Rebate Definition** for the claim line from the dropdown list. This list is filtered based on the Conservation Program associated with the rebate claim.
- Select the **Manufacturer/Model** for the claim line from the dropdown list. This list is filtered by the Rebate Definition selected.
- Enter the **Install Date** for the claim line.
- Enter the **Number of Units** being claimed.
- Click **Save.** To return to the Rebate Claim Line zone without adding the claim line, click **Cancel.**

Deleting Rebate Claim Lines

Use the following procedure to delete a rebate claim line from a rebate claim:

- Click the delete icon for the claim line you wish to delete.
- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the claim line, click **Cancel.**

> **Note**
>
> You can delete a rebate claim line only when the rebate claim is in Pending status.

Editing Rebate Claim Lines

Use the following procedure to edit an existing rebate claim line on a rebate claim:

- Click the edit icon for the claim line you wish to edit.
- Edit the details of the claim line as appropriate.
- If you are revising the data based on updated information, check the **Data Revised** checkbox and select the reason for the revision from the **Data Revised Reason** dropdown list. The **Original Data Details** section displays the data originally entered for the claim line.

Copyright © 2026, Oracle and/or its affiliates.

- If the claim line contained bad data, check the **Bad Line** checkbox and select the reason for the revision from the **Bad Line Reason** dropdown list. The **Original Data Details** section displays the data originally entered for the claim line.
- Click **Save.** To return to the Rebate Claim Line zone without changing the claim line, click **Cancel.**

> **Note**
>
> You can edit a rebate claim line only when the rebate claim is in Pending status. In addition, you cannot edit a rebate claim line after it has been verified.

Verifying Rebate Claim Lines

Use the following procedure to verify a rebate claim line on a rebate claim:

- Click the **Verify** button for the claim line you wish to verify.
- Verify that the details of the claim line match the submitted receipts.
- Click **Verify.** To return to the Rebate Claim Line zone without adding the claim line, click **Cancel.**

> **Note**
>
> You cannot edit a rebate claim line after it has been verified.

Rebate Claim Log Zone

This is a standard log zone. You use the Rebate Claim Log zone to view logging details associated with a rebate claim.

- Adding Rebate Claim Log Entries

Adding Rebate Claim Log Entries

Use the following procedure to add a rebate claim log entry:

- Click **Add Log Entry** in the title bar of the Rebate Claim Log zone.
- Enter the log entry details in the Add Rebate Claim User Log dialog.
- Click **OK.** To close the dialog without saving the log entry, click **Cancel.**

Communications

- Device Events

Copyright © 2026, Oracle and/or its affiliates.

- Activities
- Communications
- Completion Events
- Notification Suppressions

Device Events

- About Device Events
- Processing Device Events

About Device Events

Device events are developments that have taken place relative to a device, and can include power outages, power restorations, tampering alerts, command completions, among others.

Attributes used to define device events include the following:

- **Device Event Date/Time**: the date and time of the event. This date/time is flexible in its interpretation. For single events it is the moment the event took place. For events that are paired, like a power up and a power down, it will represent the start of the event when received with the first of the pair and the end of the event when received with the second of the pair. For an event that includes the duration, where the device event end date/time is also supplied, this date/time represents the start date/time.
- **Device Event End Date/Time**: the end date and time of events with durations. This will only be populated if the device event date/time is also populated.
- In addition, device events also reference details specific to the head-end system that sent the event, including the following:
- **Sender**: the head-end system (defined as a service provider in Oracle Utilities Smart Grid Gateway) from which the event was sent.
- **External Sender ID**: the external ID for the head-end system that sent the event.
- **External Event Name**: the external, head-end-specific name for the event. This name is translated into a "standard" event name within Oracle Utilities Smart Grid Gateway.
- **External Source Identifier**: an identifier for the source of the event.

Standard Event Names

When the system is loading a device event, it first determines which head-end system sent it, and then determines which device produced it. The system then maps the device event name sent by the head-end system to a standard event name that corresponds to the event. Standard event names are beneficial because the event names sent by head-end systems vary, even when the events are essentially the same. What one head-end system calls "Primary Power Down," another system may call "Last Gasp." Creating a standard event name, such as "Power Outage - Device," makes it easier to send the device events to an external system such as Oracle Utilities Network Management System. In this way, the external systems that receive the events do not need to be configured to accept all of the possible names for a single type of event.

Refer to Understanding Device Event Types for additional information about how the device events are mapped using standard event names.

Copyright © 2026, Oracle and/or its affiliates.

Processing Device Events

Device Events received from external systems are processed through two major stages:

- **Device Event Seeder**: This is the common interface point that receives all device event information and performs critical validations and conversions. During this process the device event seeder identifies the appropriate specific device event business object for further processing. The choice is made based on the originating external system and the type of event being received. For example, a device event for a "Last Gasp" would result in the paired event first device event being selected.
  If for some reason the device event seeder encounters an error an instance of the device event eeder will be created to manage the resolution of that error.
- **Specific Device Event**: Each specific device event has the appropriate logic for the type of event that has been received. A tamper event might create a service issue monitor while a "Last Gasp" might create an outage activity.

Device Event Seeder Processing

The device event seeder receives device event data in a common structure, however, the content of the data (identifiers, date/times, etc) can vary depending on the system and the particular device that generated it. To accurately translate the external content to values understood by the system the device event seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into device event seeder processing:

- **Identifier Types:** Each device event must identify the device the event has been generated from. The system supports a multitude of identifiers for the device. The service provider configuration specifies which identifier type is used for the device which enables the device event seeder to search for the correct destination for the device event. For example, a service provider might indicate that devices are identified by a serial number.
- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has the following processing methods that are used in the evaluation of the device event:
- **Device Event Mapping :** Maps a given external device event name to a standard event name. That standard event name is eventually used to identify the appropriate device event for specific processing.
- **Time Zone Translation:** Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into IMD Seeder processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming device event date/times will be stored without an adjustment for DST.

With these inputs the device event seeder is able to translate the external identifiers into the appropriate identifiers to tie the device event to the originating device in the system..

Copyright © 2026, Oracle and/or its affiliates.

Device Event Seeder Critical Validations

For device events to continue on to specific device event processing they must pass the critical validations:

- The device must be identified.
- The device event must have an event date/time.

If either of the critical validations fails the device event processing will be halted and an instance of the device event seeder will be created

Note: There is special logic that exists for handling the situation where either the device event date/time or the device event end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time.

Device Event Seeder Time Zone Translation and Conversion

To provide the most accurate tracking of device events all device event data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the device event seeder contains functionality to convert incoming date/times from a source time zone into the time zone of Oracle Utilities Meter Data Management (as indicated by the base time zone)

There are three components to identifying the source time zone of the incoming device event:

- **Service Provider IMD Import Date/Time Format:** When the device event date/times are sent in a format that identifies the time zone for each date/time the date/times will be delivered to the device event seeder already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
    1. The time zone as indicated by the device event data
    2. The service point time zone
    3. The device configuration time zone
    4. The base time zone

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

Specific Device Event Processing

Types of Device Events

Standard

Momentary device events, such as a tamper alert, are processed using the standard device event. These device events do not provide any special processing.

Copyright © 2026, Oracle and/or its affiliates.

Paired Device Events

Device events with a duration are defined using paired event device events: paired event first and paired event last. With the first of the pair representing the start of the event, and the last of the pair representing the end of the event. Events of this type can be configured to create or complete activities that represent the event. For example, a "Last Gasp" event might create an outage activity that is completed when "Power Restoration" event is received.

When pairs of events arrive in rapid succession (such as a last gasp followed quickly by a power restoration), these "paired event" business objects are designed to prevent them from being sent to subscribing applications.

Communication Response

The communication response is a special type of event used for head end systems that respond to smart meter commands with device events. When received these events create a communication in that will initiate the completion of the originating smart meter command.

Refer to About Communication In for more information on a communication in.

Sending Device Events to Subscribing Systems

When device events are received, they are typically passed onto to another subscribing system, such as Oracle Utilities Meter Data Management, a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.

The means of sending device event information to subscribing system is defined in the "How to Send Device Event Related Information" processing method for the service provider representing the subscribing system. Device event information can be sent via outbound communication business object, outbound communication, or batch process.

The method for sending device events to subscribing systems (business object, outbound message, or batch process) can be defined for each device event category, and can be overridden for individual device event types (including the ability to exclude specific device event types within a category. In addition, a default event processing method can be also be configured that applies when methods of transmission aren't specified at the individual category level.

The "Subscribe to Device Event" business service is used to process device event subscription requests, and allow external applications to manage the categories of events they receive.

Refer to About Device Event Notification Suppressions for more information on a suppression device event notifications.

Creating Service Issue Monitors and Service Investigative Orders

Device events can be configured to create service issue monitors and service investigative orders. The service issue monitor type to create is defined on the device event type. When a device event is received, an Enter algorithm on the device event's Processed state creates the service issue monitor, based on the Service Issue Monitor Type specified for the device event type.

Refer to Understanding Service Task Types for more information on configuring service issue monitor types.

Copyright © 2026, Oracle and/or its affiliates.

Device Event Prioritization

By default, device events are created and/or received into the system in an initial state is "Pending" and are transitioned to subsequent states via the batch processing. This batch processing can be skipped by setting the Execution Method flag on the device event to "Real Time" (D1RT), in which case an Enter algorithm transitions the device event to the next state in its lifecycle immediately. This feature can be used to prioritize processing of certain types of device events.

The Execution Method flag can be set in in unsolicited device events sent from the head-end system, or can be inserted during Oracle Utilities Smart Grid Gateway Adapter processing.

Oracle Utilities Smart Grid Gateway Adapters set this flag to "Real Time" on all device events created as the result of smart meter commands.

Activities

- About Activities
- Maintaining Activities

About Activities

Activities cover a range of functional process within the system from smart meter commands, synchronization of consumption between related measuring components, service order management activities, and statistics and aggregations.

The table below provides the full list of activity type categories supported and a brief description of how they are used:

Activity Type Category	Activities of this type category are used to
Bulk Activity	Initiate a smart meter command for a group of meters using Oracle Utilities Smart Grid Gateway
Command Request	Initiate a smart meter command for an individual meter using Oracle Utilities Smart Grid Gateway
Consumption Sync	Trigger related measuring component estimation synchronization
Device Event Activity	Represent a device event. These are used when viewing usage and events with the Oracle Utilities Meter Data Management 360 Degree View.
Dimension Scanner	Trigger dimension scanning when aggregating usage, statistics, and other data.
Error Activity	Represent errors in activity processing.
Extract Request	Retrieve usage and device events from a head-end system using Oracle Utilities Smart Grid Gateway
Field Activity	Initiate field work via a message to an external field work system. Activities of this type are most often created by a request orchestration activity.

Copyright © 2026, Oracle and/or its affiliates.

Meter Read Download Activity	Download a list of meters to be read.
Multi Device Command Request	Initiate a smart meter command for multiple meters (for head-end systems that support multi-device commands) using Oracle Utilities Smart Grid Gateway
Non-Dispatchable Activity	Represent steps or stages in a larger process in which the system is waiting for a resolution from another system. Examples include "Wait for Measurement" or "Wait for Scheduled Read."
Orchestration Maintenance	Update or cancel an active service order activity.
Payload Statistics	Calculate processing statistics for usage and device event import processing
Request Orchestration	Orchestrate the steps necessary to fulfill a service request, based on the current state of the service point related to the request.
Suppression	Suppress sending of device events to subscribing systems (such as Oracle Utilities Network Management System)
Usage Transaction Correction Processor	Trigger correction processing for usage transactions when corrected usage is received that would impact an existing usage transaction

Maintaining Activities

You use the **Activity** portal to view and maintain activities.

Refer to About Activities for additional functional information.

You can access the portal by selecting **Main**, **Communication**, **Activity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When creating new activities using this portal, if the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal will be available.

This portal includes the following zones:

- **Activity**: defines the basic attributes of the activity
- **Field Activity Attachments**: lists attachments related to the activity (displayed only if the current activity is a field activity)
- **Activity Hierarchy**: a tree that displays the activity and related activities (including field activities and smart meter commands), inbound and outbound communications, and completion events

Copyright © 2026, Oracle and/or its affiliates.

- **Activity Related Completion Event**: lists completion events (if any) related to the activity

Manually Completing Service Order Field Activities

Service order field activities in the "Communication in Progress" state can be manually completed or canceled by clicking **Complete** in the **Record Actions** section of the **Activity** zone. This opens the **Field Activity Manual Completion** page where the user can enter appropriate details to manually complete the field activity (such as completion information, service point information, device information, and so on). When submitted, this page creates an inbound communication and transitions the field activity to the next state as appropriate.

Note: Implementations can define custom service scripts for each field task type in the **Manual FA Completion Controls** section on the Field Task Type extendable lookup to customize the sections displayed on the **Field Activity Manual Completion** page.

Communications

- About Communication Out
- About Communication In
- Maintaining Communications

About Communication Out

These transactions are typically created from an activity and are tasked with sending a message to an external system. They follow one of two communication patterns:

- Two-way: these messages are sent to the external system without an expectation of an immediate response, or a response is received but it is only an acknowledgement of a successful receipt of the message. The true response to the message is expected to be received in a process detached from the communication out. For the full lifecycle of the communication to be complete a communication in must be received.
- One-way: these messages are sent to the external system and responded to in full within a single transaction. No communication in will be received.

The format of the message sent to the external system is typically driven by a processing method for that external system and the transportation method is controlled by the external system's message options.

If an error is encountered sending the message to the external system the communication out can perform automated retry of the messages for a configurable rate of retry and number of retry attempts.

For one-way communications the communication out will be responsible for creating the completion event.

Once the message has been communicated and a response received the outbound communication will update the initiating activity with the result.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to the following topics for additional information on communications functionality:
>
> - Understanding Processing Methods for additional information on how processing methods are configured.
> - See About Completion Events for details on completion events.
> - See External Systems in the Oracle Utilities Application Framework *Administrative User Guide* for additioal information on configuring communication between your system and external systems.

About Communication In

These transactions represent a message sent from an external system and are received in two situations:

- In response to a communication that had been sent to the external system. This is the most common usage of the communication in.
- An unsolicited message received from an external system such as a real time event. Note: this is not a common pattern.
- Upon receipt the communication in will link itself to the communication out that initiated it. The response message will then be validated to ensure that it is technically and functionally correct.
- For many smart meter commands and field activities executed by service order management the communication in is responsible for creating the appropriate completion events. For example, if the response was for the On Demand Read smart meter command a completion event will be created that will generate the appropriate initial measurement data.
- Similar to the communication out a communication in that encounters an error can perform an automated retry with a configurable rate of retry and number of retry attempts.

Maintaining Communications

The Communication portal is used to maintain communication in and communication out records.

Refer to About Communication Out and About Communication In for additional functional information.

You can access the portal from the **Main > Communication > Communication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Communication**: defines the basic attributes of the Communication
- **Communication Hierarchy**: a tree that displays the communication and related activities (including field activities and smart meter commands), inbound and outbound communications, and completion events.

Copyright © 2026, Oracle and/or its affiliates.

Completion Events

- About Completion Events
- Maintaining Completion Events

About Completion Events

Completion events are used to implement the outcome of an activity. They are created by either the communication out (for one-way communication) or the communication in (for two-way communication). Completion events are executed by the activity orchestrating the communication with the external system.

Some examples of the tasks a completion event will perform are:

- Commissioning, decommissioning, connecting, disconnecting a smart meter
- Installing or Removing a meter or item
- Creating initial measurement data
- Updating a device or service point

Maintaining Completion Events

The Completion Event portal is used to maintain completion events.

Refer to About Completion Events additional functional information.

You can access the portal from the **Main > Communication > Completion Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Completion Event**: defines the basic attributes of the Completion Event
- **Completion Event Related Objects**: displays data related to the completion event such as a device, service point, or initial measurement.

Notification Suppressions

- About Device Event Notification Suppressions
- Maintaining Notification Suppressions

About Device Event Notification Suppressions

When a device event is processed by the system it can be sent to one or many external applications interested in being notified about the event. A prime example of this is the Outage Management System subscribing to any device events related to a meter either losing power or regaining power.

Copyright © 2026, Oracle and/or its affiliates.

There are situations, such as a field worker performing maintenance on a meter or on a node that impacts many meters, that will result in a large amount of device events being created, processed, and distributed to subscribers.

Device event notification suppressions provides a way for a subscribing system, in anticipation of scheduled maintenance, to suppress notification of certain device events (power down, power up, tamper, etc) from one or more meters.

Maintaining Notification Suppressions

The **Activity** portal is used to maintain notification suppressions records.

Refer to About Device Event Notification Suppressions for additional functional information.

You can access the portal from the **Main > Communication > Notification Suppression**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Activity**: defines the basic attributes of a notification suppression activity

Integrations

- Business Flags
- Analytics Integrations

Business Flags

- About Business Flags
- Maintaining Business Flags

About Business Flags

Business flags provide a view into the status of a service point across applications (such as Oracle Utilities Customer Cloud Service, Oracle Utilities Meter Data Management, Oracle Utilities Customer Care and Billing and so on). Business flags are currently available in two forms:

- **Service Point Business Flags**: are often informational in nature but also allow for manual investigation to be taken.
- **Service Point Monitor Business Flags**: are seen as symptomatic of an issue that needs to be investigated. These business flags feed into a service issue monitor and may result in a service investigative order being created.

For additional details see About Business Flags in the Oracle Utilities Application Framework *Administrative User Guide*.

Copyright © 2026, Oracle and/or its affiliates.

Business Flag Allowed Actions

Allowed actions are implemented presented to the user as selectable options that will guide the user through a particular method of investigating or resolving the business flag. The following actions are supported:

- **Create Field Activity**: This will initiate a field activity associated to the business flag's service point. The type of field task will be selected by the user. Note: this option requires Oracle Utilities Service Order Management.
- **Request Customer Notification:** This will initiate a message to the appropriate external system (e.g. Oracle Utilities Customer Care and Billing) that will identify a notification to be sent to the customer. The type of communication will be determined by the notification type selected by the user.

Note: not all business flags support allowed actions and not all business flags that support allowed actions support all allowed actions.

Maintaining Business Flags

Information for this functionality is available in the Business Flags section in the Oracle Utilities Application Framework *Administrative User Guide*.

Analytics Integrations

- Using the Meter Insights Investigation Viewer

Using the Meter Insights Investigation Viewer

The Meter Insights and Investigations functionality leverages the Oracle Utilities Analytics Insights business flag integration to support receipt of revenue protection events (referred to as "insights") from Oracle Utilities Analytics Insights for research and investigation using Oracle Utilities Meter Data Management (or Oracle Utilities Meter Solution Cloud Service). Revenue protection insights sent from Oracle Utilities Analytics Insights for investigation are events that suggest potential theft or loss of revenue such as unexpected consumption from an inactive meter, evidence of meter tampering, discovery of an improperly installed meter (such as a meter installed upside down or in reverse), and so on. Insights received from Oracle Utilities Analytics Insights create business flags, which in turn create related Meter Insight Investigation service tasks (referred to as “investigations”). Business users can use the **Meter Insights Investigations Viewer** portal to search, view, and investigate these investigations. This portal is intended to provide as much investigative information as possible so as to allow the user to complete as much work as possible without leaving the portal.

Searching

Use the following procedure to search for investigations:

1. Navigate to the portal by selecting **Main**, then **Insights**, then **Meter Insights Investigation Viewer**.

   By default the **Investigations** zone displays search results for unassigned records and records assigned to the current user.

Copyright © 2026, Oracle and/or its affiliates.

2. To search by specific search criteria or display the search criteria and filters, click the **Show Filters** link or icon. The **Investigations** zone expands to display the available search filters, including:
    - Business Flag Type
    - Address (including City and Postal Code)
    - Average Daily Usage
    - Priority
    - Service Type
    - Service Point Type
    - Assigned to User ID
    - Flags to include Unassigned records, and investigations with current active field activities.
3. Enter your search criteria and click **Search**.

The **Investigations** zone displays a default set of data about each investigation. To select the information to display for each, click the **Display** link or icon, select or deselect the data you wish to view or hide, and click **Display**.

Viewing Service Issue Monitors and Consumption

To view additional information about an investigation, click the **Expand Section** icon in the far left column. The section expands to display:

- Comments
- Service Point Information
- Field Work Information
- A list of insights (business flags) related to the investigation
- Consumption (a section similar to the **Final Values Overlay** zone in the **360 Degree View** portal)

To view additional information about the investigation, service point, or field activity, click the icon at the end of its information string. A new browser window will open displaying the **Service Task**, **Service Point** tab of the **360 Degree View** portal or **Activity** portal (respectively). This allows the user to investigate the issue in more detail as needed without having to navigate away from the **Meter Insights Investigations Viewer** portal.

To expand the **Consumption** section, click the **Expand Section - Consumption** icon on the far right side of the **Consumption** section heading. The section expands to display the following:

- **Parameters**: The default parameters used to display consumption for the service point. This includes **Measuring Component**, **Value Identifier Type**, **Start Date/Time**, **End Date/Time**, **Final Values Overlay Profile**, and **Event Bar Profile**. These values are based on the **Consumption Data Source** parameters on the investigation's **Investigation Type** (Service Task Type). The measuring component and value identifier type is based on service point type, interval/scalar flag, and UOM/TOU/SQI combination, and the date range is based on the start and end dates of the investigation and the **Days Before to Graph** and **Days After to Graph** parameters.
- **Values to Graph**: A list of measuring components to graph, including measuring components defined in the current **Final Values Overlay Profile**, as well as other measuring components selected by the user.

Copyright © 2026, Oracle and/or its affiliates.

- **Measurement Data**: A graph and tabular display of the consumption and values to graph, and event bars based on the current **Event Bar Profile**.

To change the displayed consumption, adjust the **Parameters** and **Values to Graph** and click **Load** (in the **Parameters** section).

Working With Investigations

Users can also add comments, assign a recipient for To Do entries, and change the status of investigations from this portal.

- To add or edit a comment, click the **Edit** icon in the second column from the left. The **Edit** icon is replaced by a checkmark icon and an "X" icon, and a **Comments** box opens. Enter or edit the comment text and click the checkmark icon. To delete a comment, click the "X" icon.

  Comments are saved on the Investigation, and can be viewed from the **Log** tab on the **Service Task** portal.
- To assign a recipient for To Do entries, click the **Edit** icon in the second column from the left. The **Edit** icon is replaced by a checkmark icon and an "X" icon, and the **Assigned To** column displays a drop-down list. Select the user to which To Do entries should be assigned, and click the checkmark icon.
- To change the status of the investigation, click the icon in the **Action** column. A popup menu will open displaying available options including:
  - Initiate Field Work: Select this to create a field activity to investigate the issue from the field (this transitions the investigation to the Waiting for Field Work state).
  - Issue Resolved: Select this if the issue has been resolved (this transitions the investigation to the Issue Resolved state).
  - No Issue: Select this if there is no issue (this transitions the investigation to the No Issue state).
  - Inconclusive: Select this if/when the investigation is inconclusive (this transitions the investigation to the Inconclusive state).
  - Discarded: Select this to discard issue (this transitions the investigation to the Discarded state).

See Meter Insights and Investigation Processingin the *Administrative User Guide* for more information about what happens when the investigation enters one of these states.

Aggregations

- Standard Aggregations
- Dynamic Aggregation

Standard Aggregations

- About Aggregations
- Creating Aggregators
- Base Product Aggregators

Copyright © 2026, Oracle and/or its affiliates.

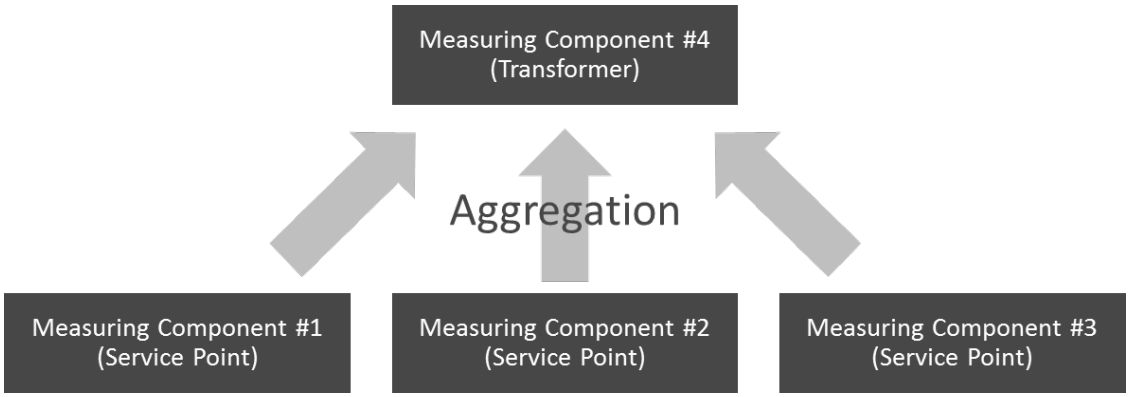
- Maintaining Aggregations

About Aggregations

Aggregations are measurements that represent a summarization of other measurements from a set of devices, items, and/or measuring components. For example, an aggregation may be based on the sum of the electric consumption of all residential customers in a particular postal code within the utility's service territory. The group of related measuring components for which measurements are summarized are referred to as constituent measuring components.

Every type of aggregation has one or more dimensions. For example, an aggregation might be defined with three dimensions as follows:

- Substation
- Feeder
- Transformer



Aggregations are defined by aggregator measuring components. A separate aggregator measuring component should be defined for every distinct combination of dimensions for a given type of aggregation. For example, the transformer aggregation described above would require an aggregator measuring component for every distinct combination of substation / feeder / transformer found among the electric Service Points.

Aggregations can also be aggregated. For example, the aggregator measuring components for each substation / feeder / transformer combination can in turn be used to aggregate consumption at each substation / feeder combination. In this case, a distinct aggregator measuring component would be defined for each unique substation / feeder combination. Likewise, the aggregator measuring components for each substation / feeder combination could be used to aggregate consumption at each substation, in which case, distinct aggregator measuring components would be defined for every unique substation. Finally, the aggregator measuring components for each substation can be used to aggregate total consumption for all substations, with an aggregator measuring component defined to represent the entire service area.

Aggregation Calculations

The system periodically aggregates consumption via batch process, using a deferred monitor on the aggregator measuring components. In addition, users can re-aggregate data in real-time if they don't wish to wait for the batch process, or if the original aggregation needs to be re-calculated due to incorrect data. Users can also create ad hoc aggregations "on the fly" and these will persist in the database in the same manner as other aggregations.

Copyright © 2026, Oracle and/or its affiliates.

Understanding Aggregation Periods

The start and end dates and times for aggregation calculations are based on the following:

- Aggregation Horizon
- Aggregation Lag
- Aggregation Cut Off Time

Whenever aggregation is performed for an aggregator measuring component, consumption is aggregated for every day in its "Aggregation Horizon." The "Aggregation Horizon" is the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component.

Aggregation calculations typically lag behind the current date by a few days to give the system time to upload and perform validations and create final measurements. The amount of lag time is referred to as the "Aggregation Lag" and is the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the end of the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.

Aggregation is always performed through a given "through time" (such as 12:00 AM) rather than through the actual time of the aggregation calculation. This time is referred to as the "Aggregation Cut Off Time." For example, the stop time for aggregation calculations with an Aggregation Lag of 2, and an Aggregation Cut Off Time of 10:00 PM will always be 10:00 PM 2 days prior to the date on which the calculations are performed. In the above examples, aggregations performed on January 9 would have an end date/time of 10:00 PM on January 7, and aggregations performed on January 10 would have an end date/time of 10:00 PM on January 8.

Aggregation and Re-Aggregation

The use of the aggregation horizon means that aggregated totals for some days will be re-aggregated until those days are no longer covered by the aggregation horizon. For example, with an aggregation horizon of 5 days and an aggregation lag of 2 days, on the night of January 9 the aggregation period would be January 3 through January 7 (as in the above example).

On the night of January 10, the horizon will shift 1 day (to January 4 through January 8, again as above). This means the following:

- The totals for January 3 calculated on January 9 will be untouched (January 3 now falls outside the aggregation horizon)
- The totals for January 4 through January 7 will be re-derived because corrections may have occurred (and they still fall within the aggregation horizon)
- The totals for January 8 will be calculated for the first time (because it now is within the aggregation horizon).

Copyright © 2026, Oracle and/or its affiliates.

Manually Read Meters and Aggregation Lag

Aggregations that include manually read meters often have much longer aggregation lags than those for automatically read meters. This allows more time for manual meter readings to be imported into the system for use in aggregations.

For example, suppose a situation where manual meter reads arrive approximately 1 month after the date of the reading. In this case, it wouldn't be until a meter read upload on February 7 that the last manual reads including consumption from January 6 will exist. Since we don't want to perform aggregations for January 6 until there's a decent chance that all consumption for that date exists, an aggregation lag of 32 days ensures that the data for January 6 is in the system when the aggregation is performed on February 7.

When An Aggregator Measuring Component Contains Both Manual and AMI Measuring Components

The previous section suggests that if an aggregator measuring component contains both manually read and AMI measuring components, the aggregator measuring component should have a long lag time rather than the short time shown in the earlier example.

While this is one possible approach, an alternative approach could allow users to see timely aggregations of the AMI channels and only lag the manual channels. To do this they could configure the system as follows:

- Create an aggregator measuring component that only aggregates manual measuring component types (horizon: 5 days, lag: 31 days)
- Create a separate aggregator measuring component that only aggregates interval measuring component type: 5 days types (horizon: 5 days, lag: 2 days)
- Create a third aggregator measuring component that aggregates the above aggregator measuring component types (horizon: 5 days, lag 31 days)

Expanding The Aggregation Horizon Periodically

Even with a 30-ish day aggregation lag, it's possible for an account with several months of estimated consumption to have its consumption "invalidated" when real readings arrive whose dial readings are less than the estimated readings (or when major retroactive data change takes place). Another similar situation is when a user manually corrects consumption after the aggregation horizon has moved forward.

To address situations like these, the use of ad-hoc aggregations with a user-specified aggregation horizon supports the notion of periodically (such as once a month or once a week) aggregating with a long aggregation horizon (up to 90 days) to catch as many retroactive changes as possible.



A Note About Using Non-Effective Dated Dimensions

Be aware that whenever aggregation takes place, the system uses master data that exists as of that date. For example, if an aggregation is performed using non effective-dated master data attributes (for example, aggregating by measuring component type) and a measuring component's type is changed, all days in the aggregation horizon will reflect the change regardless of when the user makes the change, because the measuring component type relationship is not effective-dated.

The impact of this depends on the nature of the change:

Copyright © 2026, Oracle and/or its affiliates.

- If the master data was wrong, this provides a way to re-aggregate the data to accurately reflect the master data.
- If the change occurred effective as of a given date (but there's no effective dated attribute), this can result in incorrect aggregation values.

Because of this, it is recommended that where possible dimensions used in aggregations be effective-dated.

For information on configuration, reference the related section for Aggregations in the Administration Guide.

Creating Aggregators

There are two ways in which aggregator measuring components are created:

- Automatically: The system will periodically create new aggregator measuring components when new "dimensional values" are detected. For example, if a new substation is added to the system, a new aggregator measuring component will be created for the substation.
- Manually: An end-user can create an aggregator measuring component at their discretion

Automatic Creation of Aggregator Measuring Components

The system automatically creates aggregator measuring components when it detects new instances of the dimension(s) being aggregated. This activity is referred to as dimension scanning. For example, if the system is configured to aggregate individual substations, a user will NOT have to manually set up an aggregator measuring component when a new substation if referenced on a service point. Rather, the system will create the new aggregator when it detects a substation on a service point that doesn't have an aggregator measuring component.

Similarly, if is configured to aggregate distinct combinations of rate class (on usage subscription type) and postal code (on service point), the system will create an aggregator measuring component when it detects new combinations of rate class and postal code.

The periodic monitoring to ensure sufficient aggregation measuring components exist is implemented via monitoring algorithms on a related activity business object that references a corresponding measuring component type. This means that there must be a separate activity type / activity combination for every type of aggregation performed. These monitor algorithms include:

- A Monitor algorithm on the Active state that simply transitions the activity to the transitory Scan state. Users can manually transition the activity to the Scan state by clicking the "Scan" button on the activity.
- An Enter algorithm on the Scan state finds the distinct combinations of dimensional values for this type of aggregation and checks that an aggregator measuring component that references the activity's aggregator measuring component type exists for every instance and creates new aggregator measuring components as appropriate. The algorithm also populates the dimensional values on the new measuring components (mapped to searchable characteristics), and sets the Consumption Aggregated Through Date/Time field on the new aggregator measuring components to the current date and the Aggregation Cut Off Time on the aggregator measuring component type.

Hard-Wiring Service Types for Aggregations

If the dimensions used for aggregation do not include at least one service type-specific dimension (for example, in the case of aggregating by postal code only) AND the aggregation should not commingle measurements of different service types, it's important to:

Copyright © 2026, Oracle and/or its affiliates.

- Declare the specific service type for the aggregation on the dimension scanner activity instance
- Declare the desired service type-specific measuring component types on the aggregator measuring component type

For example, if an electric and gas implementation aggregates consumption by postal code, they will need 2 activities: one will reference the Electric service type and the desired aggregator measuring component type, the other will reference the Gas service type and its aggregator measuring component type. The aggregator measuring components will have two dimensional attributes: postal code and service type.

Manual Creation Of Aggregator Measuring Components

Users can also create aggregator measuring component manually. This is useful when an ad hoc aggregation for an ad hoc time period is required, or when there is no activity that automatically creates instances of a given combination of dimensions. To do this, the user simply clicks the "Add" link on the Aggregator Search zone title bar. The BPA script:

- Prompts the user to define the type of aggregation (this will present a list of measuring component types whose measuring component category is aggregator)
- Prompts the user to define the dimensional value(s), the desired aggregation time, and if this aggregation should be performed indefinitely or if it is a "one time"

Base Product Aggregators

A number of standard aggregation processes are delivered as part of the base product. These provide a means of aggregating measurement data by postal code and service type for both physical meters and items.

Aggregator Name	Description
Aggregator - Postal and Service Type	Aggregates measurement quantities for constituent measuring components from physical meter based on postal code and service type dimensions.
Item Aggregator - Postal and Service Type	Aggregates measurement quantities for constituent measuring component from items based on postal code and service type dimensions.

Aggregators Used With Analytics

Other aggregators shown below are delivered for use by Oracle Utilities Analytics and Oracle Utilities Analytics Visualization (OUAV). These aggregate measurement data (including quantities and counts) for constituent measuring components based on the following dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

Aggregator Name	Description
Measurement Measured Quantity Aggregator OUAV Subject Area: Meter - Aggregated Measurement Quantity	Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes.

Copyright © 2026, Oracle and/or its affiliates.

Measurement Quality Count Aggregator OUAV Subject Area: Meter - Aggregated Measurement Count	Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes.
Measurement Timeliness Count Aggregator OUAV Subject Area: Meter - Aggregated Timeliness Count	Aggregates measurement counts that arrived on time or are late.
Measurement Timeliness Quantity Aggregator OUAV Subject Area: Meter - Aggregated Timeliness Quantity	Aggregates measurement quantities that arrived on time or are late.

See the Oracle Utilities Analytics Visualization User Guide for more information about the OUAV Subject Areas.

Maintaining Aggregations

This portal is used to display and maintain Aggregations.

Refer to About Aggregations for additional functional information.

You can access the portal from the **Main > Totals and Trends > Consumption Statistics**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component**: defines the basic attributes of the aggregator measuring component.
- **Measuring Component Attributes:** displays attributes that describe the dimensions and criteria for the aggregation measuring component (applies to dynamic aggregation measuring components only)
- **Measurements — Interval**: displays aggregated measurement data for the selected aggregator measuring component.
- **Interval - Audit View**: allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement.
- **Final Values Overlay**: displays final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components.
- **Time of Use Overlay**: displays an overlay of the TOU periods on a final interval measurement along with totalized TOU consumption based on a user-defined time period and TOU map
- **Related Measuring Component Sets**: lists measuring components sets of which the aggregator measuring component is a member (applies to dynamic aggregation measuring components only)
- **Usage Subscriptions Directly Linked to MC**: lists usage subscriptions that are directly linked to current measuring component

A few functions are available in many of these zones that allow you to interact with the measurement data. For more information of these functions refer to the Applying Measurement Functions section.

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Aggregation

- Dynamic Aggregation Overview
- Understanding Dynamic Aggregation Execution
- Aggregation Measuring Components

Dynamic Aggregation Overview

Standard aggregation processing uses specific algorithms configured to work with a specified set of dimensions. For example, the base package Service Type and Postal aggregation uses specific algorithms for dimension scanning and finding constituent measuring components. In contrast, dynamic aggregation uses dynamic queries for the dimension scanning, find constituents, and aggregation processes.

Dynamic aggregation can be used for analysis (to view aggregated data to understand load demand, for instance) as well as settlement processing performed by Oracle Utilities Market Settlements Management.

Dynamic aggregation results can be viewed using Oracle Utilities Analytics Visualization. See Viewing Dynamic Aggregation Results Using Oracle Utilities Analytics Visualization and the Oracle Utilities Analytics Visualization User Guide for more information.

Understanding Dynamic Aggregation Execution

This section provides an overview of how dynamic aggregation processing is executed, including descriptions of the different types of dynamic aggregation processing and the different types of data that can be aggregated.

Dynamic Aggregation Execution: Aggregation Groups and Aggregation Group Runs

Aggregation processing is managed via aggregation groups. Aggregation Groups outline the sequence of aggregations that must occur for a given aggregation process. Each aggregation group includes a set of Measuring Component Sets, each of which has one or more related Aggregation Measuring Components.

When processing is initiated for an aggregation group, the system creates an Aggregation Group Run record to track the status and progress of the aggregation processing. This record is used to manage the aggregation process for each group and track which Measuring Component Sets should be executed next and when the aggregation is complete.

The following steps outlines the aggregation process:

1. The Dimension Scan and Aggregation monitor batch control (D1-DSAEM) initiates processing for all Aggregation Groups. This should be scheduled at least once per day. This monitor process runs all monitor algorithms on each aggregation group. The base package aggregation group has two monitor algorithms:
    - Aggregation Group Run Maintenance (D1-AGRM)
2. The Aggregation Group Run Maintenance algorithm verifies if the Aggregation Group is ready to aggregate by looking at the processing timetable defined for the aggregation group and the last scheduled execution. If it is time for the next aggregation then the

Copyright © 2026, Oracle and/or its affiliates.

algorithm creates a pending Aggregation Group Run which will then be used to track the status of the Measuring Component Sets for the aggregation.

3. At the completion of the Dimension Scanner and Aggregation monitor batch process, the Submit Dimension Scanner Batch (D1-SUBDIMSCN) post-processing algorithm searches for any Aggregation Group Run entries that are in progress where the next Measuring Component Set is not manual. If at least one is found then it will initiate the Aggregation Group Dimension Scanner batch process (D1-AGDS).
4. The Aggregation Group Dimension Scanner batch process identifies the next Measuring Component Set for each Aggregation Group Run that is in progress. If that next Measuring Component Set is not manual then it will retrieve each Aggregation Measuring Component for the Measuring Component Set and execute the monitor algorithms for that Aggregation Measuring Component. These include an aggregation algorithm (specific to the type of data to be aggregated such as interval, billed quantities, etc.) and in some cases a value derivation algorithm.
5. For each Aggregation Measuring Component that is monitored by the Aggregation Group Dimension Scanner batch process, the aggregation process:
    a. Executes the appropriate Find Constituents algorithm to find the list of constituents that are to be aggregated
    b. Extracts the data for those constituents that should be aggregated
    c. Executes the appropriate aggregation algorithm against that data and store the results
    d. Executes validation algorithms if applicable
    e. Executes value derivation algorithms if applicable
6. The Aggregate Next Measuring Component Set (D1-AGGNXMCS) post-processing algorithm on the Aggregation Group Dimension Scanner batch process updates the Aggregation Group Run the batch information for each Measuring Component Set that was aggregated, and requests the Dimension Scan and Aggregation monitor batch process to run again.
7. This again executes the monitor algorithms for all the Aggregation Groups which will do one of several things:
    - If the Aggregation Group Run is in progress and the next Measuring Component Set is Automatic then nothing will happen.
    - If the Aggregation Group Run is in progress and the next Measuring Component Set is Manual it will create a To Do.
    - If the Aggregation Group Run does not have a next Measuring Component Set, it will set the Aggregation Group Run to Complete

    > **Note**
    >
    > The Dimension Scanner algorithm will be executed again but since there will not be a pending Aggregation Group Run it will not do anything.

8. Once all Measuring Component Sets are complete the Aggregation Group Run Maintenance algorithm marks the Aggregation Group Run as complete.

Aggregation Scheduling

Refer to Understanding Aggregation Periods for basics about aggregation periods.

Copyright © 2026, Oracle and/or its affiliates.

Aggregation Scheduling for Aggregation Groups can be based on either a Defined Lag, or a Processing Timetable.

- **Defined Lag** scheduling uses the following parameters:
  - **Aggregation Cut Off Time**: The end time for aggregation calculations performed for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This is used to ensure a consistent end time for aggregation periods. This is especially useful when aggregating other aggregations.
  - **Aggregation Lag**: The number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.
  - **Aggregation Horizon**: The number of days in the aggregation period for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This reflects the time period during which there's a potential change in measurement data for one or more of the measuring components associated with Aggregation Measuring Components. This together with the Aggregation Lag is used to determine the start and end dates of an aggregation period.
- **Processing Timetable** scheduling uses a Processing Timetable Type to drive aggregation schedule.

With dynamic aggregation, the aggregation period is consistent for all members of an Aggregation Group when an aggregation is performed. This is despite the fact that there may be differences between Measuring Component Sets and even Measuring Components within an Measuring Component Set as to how far the data has been aggregated through. In addition, individual Aggregation Measuring Components all use the same aggregation period when aggregation is performed.

Anchor Time

Dynamic aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When processing calculations for measuring components whose type specifies an **Anchor Time**, if the Start Time and End Time of the calculation period do NOT match the **Anchor Time** on the measuring component type, the Start and Stop Time are shifted to align with the closest time to the **Anchor Time** that aligns with the interval size of the measurement data being processed. The Start Time is shifted backwards (to an earlier time) and the End Time is shifted forwards (to a later time). In the case of hourly intervals with Start Times and End Times on the hour, no shift occurs. In the case of daily intervals, the shift is to the nearest Anchor Time. For example, in the Anchor Time is 07:00AM but the Start Time and End Time are both 12:00, the Start Time would be shifted to the previous 07:00AM, and the End Time would be shifted the next 07:00AM.

- Dynamic Aggregation Processing
- Types of Aggregations

Dynamic Aggregation Processing

This section provides additional details around the specific types of processing used with dynamic aggregation.

Similar to standard aggregation, dynamic aggregation processing comprises three primary types of processes:

- **Dimension Scanning**: a process that searches for unique combinations of dimensions that are not yet represented by an Aggregation Measuring Component.
- **Find Constituents**: a process that finds all source data to be aggregated for a given Aggregation Measuring Component, based on the Aggregation Measuring Component's dimensions.
- **Aggregation**: a process that aggregates source data that corresponds to the Aggregation Measuring Component's dimensions.

Dynamic aggregation processing is driven by the following types of administrative data:

- **Data Sources** define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or other sources of data.
- **Aggregation Measuring Component Types** define the most important properties of aggregation measuring components used to store aggregated data.
- **Aggregation Groups** define the ordering of a series of related aggregations and the schedule of aggregation.
- **Measuring Component Sets** define the dimensions and criteria by which aggregation will be performed.

Refer to Dynamic Aggregation Configuration Overview for more information about configuration of this data.

- Dynamic SQL
- Dimension Scanning
- Find Constituents
- Aggregation Calculation

Copyright © 2026, Oracle and/or its affiliates.

- Foundation and Composite Aggregation

Dynamic SQL

Dynamic aggregation uses dynamic SQL queries to perform dimension scanning, find constituents, and aggregation processing. These dynamic queries are based on configuration of:

- Data Sources
- Aggregation Measuring Component Types
- Measuring Component Sets

The Source of Dynamic SQL

Data sources provide template SQL that is used as the basis for dynamic queries. This template SQL is extended by additional SQL generated from the data source configuration as well as configuration defined for measuring component sets, measuring components, and measuring component types that reference the data source. The ways in which the template SQL is extended is based on the type of process:

- **Dimension Scanning**: The data source template SQL is extended by the criteria, dimensions, and individually managed items configured on the measuring component sets being processed.
- **Find Constituents**: The data source template SQL is extended by attributes defined on the aggregation measuring components being processed.
- **Aggregation**: The data source template SQL is extended by the value identifiers defined on the measuring component type of the aggregation measuring components being processed.

Dimension Scanning

The dimension scanning process analyzes master data to identify unique dimension combinations and create an aggregation measuring component for each. Several objects play a role in this process:

- The Aggregation Group holds the monitor algorithm that performs the scanning
- The Data Source defines the template SQL for identifying dimensions
- Measuring Component Sets define the dimensions and criteria for identifying the dimension combinations

See Dynamic SQL for more about the role of data sources and measuring component sets in generating dynamic SQL queries for dimension scanning.

This process is executed when aggregation is initiated for an Aggregation Group. This process should be run on a frequent basis (daily or more frequent) and should be executed prior to aggregation processing being run. Each time it is executed the dimension scanning process takes the following high level steps:

1. Identifies all Measuring Component Sets for the Aggregation Group that are not imported from an external source
2. Identifies all yet to be represented dimension combinations
3. Creates an Aggregation Measuring Component based on the Measuring Component Set's Measuring Component Type for each dimension combination identified in step 2.

Copyright © 2026, Oracle and/or its affiliates.

4. Adds Attributes for each Aggregation Measuring Component created to describe the dimensions and criteria for that Measuring Component
5. Adds an entry in the Measuring Component Set participants table for each Aggregation Measuring Component created

This processing is performed by a monitor algorithm (Aggregation Group Dimension Scanner: D1-AGDS) on the Finalized status of the Aggregation Group business object.

Find Constituents

The Find Constituents process finds all source data to be aggregated for a given aggregation measuring component, based on the Aggregation Measuring Component's dimensions. The queries used for this process are based on data source template SQL extended by attributes defined on the Aggregation Measuring Components being processed.

The Find Constituents process is initiated by the aggregation algorithm on the Aggregation Measuring Component (each type of Aggregation Measuring Component — interval, item, billed quantities — has its own aggregation algorithm), and is performed by algorithms defined for the Find Constituents system event on the Aggregation Measuring Component business object.

This process differs slightly when finding constituents for interval and scalar measurements stored on measuring components, item data, and billed quantities.

Measuring Components

The Find Constituents process for measuring components uses the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm which retrieves data based on the following:

- Attributes on the Aggregation Measuring Component based on the Measuring Component Set configuration
- Measuring Component Types on the Data Source

Measuring component constituents are inserted into a global temporary table (GTT1) with the following information:

- Measuring Component ID
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Measuring Component's installation history.

Items

The Find Constituents process for items uses the Find Constituents for Items (D1-FICONITEM) algorithm which retrieves data based on the following:

- Attributes on the Measuring Component based on the Measuring Component Set configuration

Copyright © 2026, Oracle and/or its affiliates.

- Item Types / Service Point Types on the Data Source (for badged vs. unbadged items, respectively)

Item constituents are inserted into a global temporary table (GTT3) with the following information:

- Maintenance Object (Device or Service Point)
- Device Type: Item Type
- Item Count
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Item's installation history.

Billed Quantities

The Find Constituents process for billed quantities uses the Find Constituents for Billing Data (D1-FINDCONSB) algorithm which retrieves data based on the following:

- Attributes on the Aggregation Measuring Component based on the Measuring Component Set configuration
- Usage Subscription Types on the Data Source

Billed quantity constituents are inserted into a global temporary table (GTT5) with the following information:

- Usage Subscription ID
- Usage ID
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Usage Subscription's installation history.

Aggregation Calculation

The aggregation calculations process aggregates source data that correspond to the Aggregation Measuring Component's dimensions. The queries used for this process are based on data source template SQL extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Components being processed.

The aggregation calculation process is initiated by Aggregation by MC Set Monitor batch process (D1-AMCS), and is performed by monitor algorithms defined on the Active state of the

Copyright © 2026, Oracle and/or its affiliates.

Aggregation Measuring Component business object. The algorithms used by each type of Aggregation Measuring Component are as follows:

Aggregation Measuring Component Business Object	Aggregation Algorithm
Dynamic Aggregator - Billed Quantities (D1-AggBilledServiceQuantities)	Aggregate Billed Service Quantities (D1_AGGBSQ)
Dynamic Aggregator - Composite (D1-AggregatorComposite)	Aggregate Interval Measurements (D1_AGGINT)
Dynamic Aggregator - Customer Quantities (D1-AggregatorCustomerQuantity)	Aggregate Customer Specific Quantities (D1_AGGCSQ)
Dynamic Aggregator - Interval (D1-AggregatorInterval)	Aggregate Interval Measurements (D1_AGGINT)
Dynamic Aggregator - Items (D1-AggregatorItems)	Aggregate Item Data (D1_AGGITEM)
Dynamic Aggregator - Scalar (D1-AggregatorScalar)	Aggregate Scalar (D1_AGGSCLR)

See Types of Aggregation for more information about aggregation calculations for each type.

Other Calculations

In addition to basic aggregation processing, other calculations can be performed on aggregated measurements as needed. For instance, a common step in aggregating date for use in market settlements is applying a loss factor to aggregated data to reflect system losses (line loss, etc.) between generation-level quantities and end consumption quantities.

These types of calculations can be performed by value derivation algorithms. These are algorithms based on the Derive Aggregated Quantity Using a Formula (D1-DERAGGQTY) algorithm type which apply a configured formula to aggregated measurement values. For instance, to apply a loss factor, an algorithm can retrieve effective-dated factor values for a loss factor and multiply each aggregated measurement value by the appropriate factor value.

The Execute Aggregation Value Derivation (D1-EAGGVALD) algorithm on Aggregation Measuring Components executes any value derivation algorithms specified for Value Identifiers on the Aggregation Measuring Component Type.

Aggregation Calculation and Anchor Time

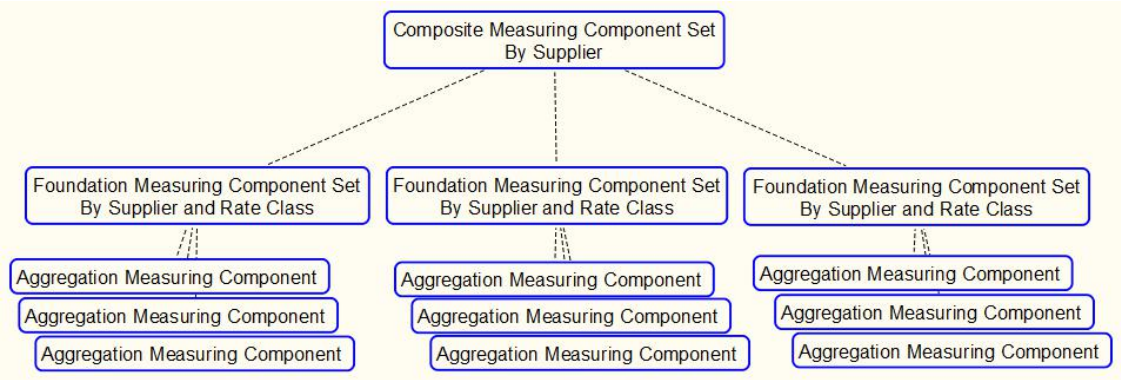
Dynamic aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

Foundation and Composite Aggregation

Dynamic aggregation supports two “levels” of aggregation. Aggregating source data, from measuring components, items, or billed quantities is referred to as “Foundation” aggregation. Aggregation of aggregated data is known as “Composite” aggregation. Composite aggregation provides the ability to aggregate data from a set of other aggregations.

The diagram below illustrates a simple example of the relationship between foundation and composite measuring component sets. Each of the three foundation measuring component sets has a number of aggregation measuring component participants that aggregate data based on a set of dimensions and a particular supplier and rate class. Those in turn are participants in the composite measuring component set that aggregates based on the supplier (including data for all rate classes).

Copyright © 2026, Oracle and/or its affiliates.



A common scenario where composite aggregation would be used is to consolidate data across several foundation aggregation measuring component sets that pulled data from a set of data sources ordered by precedence such that the best data was sourced when available and when not the aggregations fell back to a lesser quality data source:

- Foundation: Interval measurements from usage subscriptions and / or service points
- Foundation: Scalar measurement from usage subscriptions and / or service points
- Foundation: Interval measurements from estimations (backcasted using actual weather)
- Composite: Aggregation by supplier

In this example the goal of the first aggregation would be to retrieve as much interval data as possible where that interval data covered the entire period being settled for. The second aggregation would take scalar data for those instances where interval data was not available and apply a profile to produce the appropriate intervals. The third aggregation would fill in any remaining gaps that were not covered by interval or scalar data with estimations based various forecasting models.

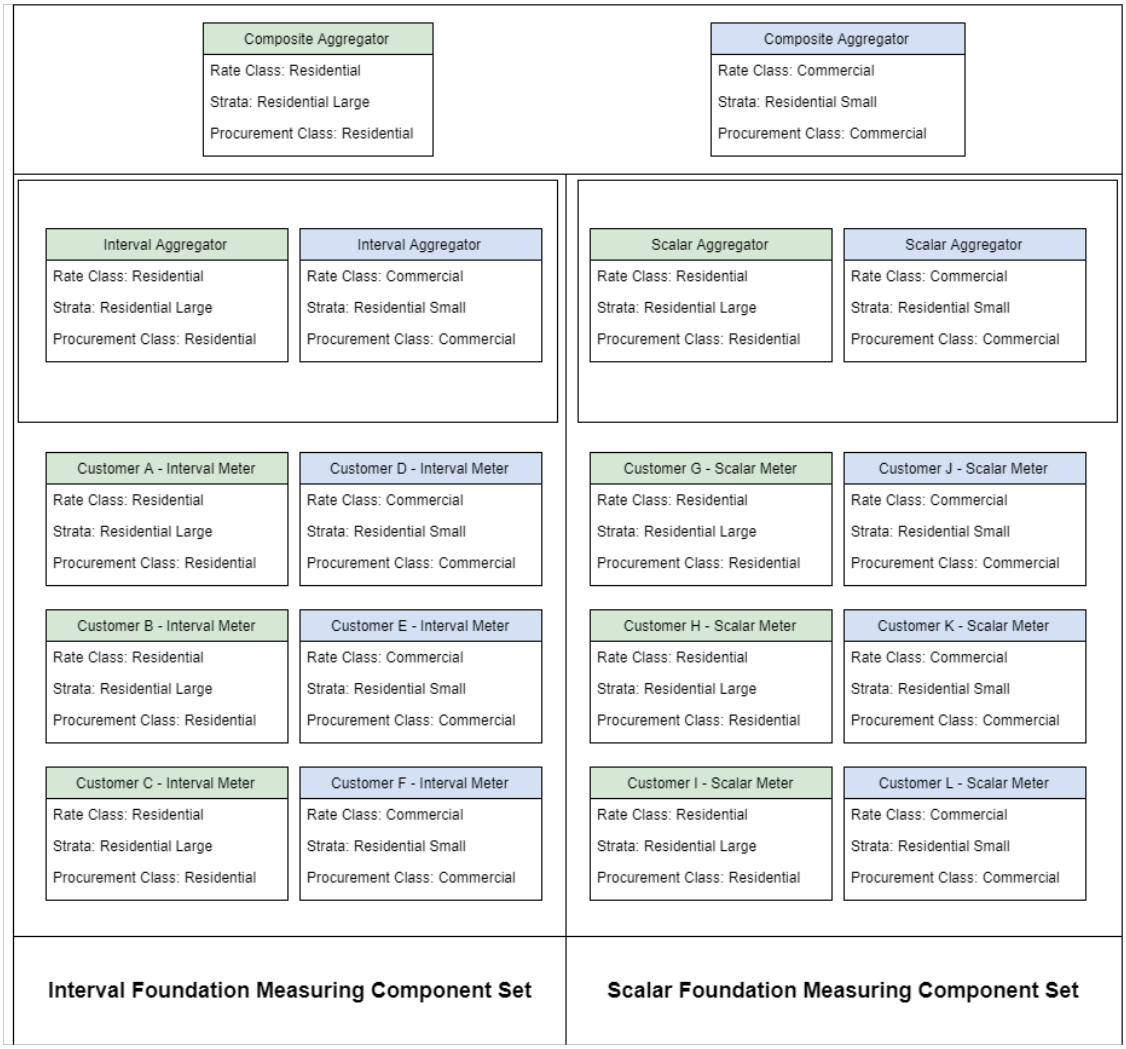
At the completion of those first three aggregations there would be the full resolution of data for the customer base. However, the values would be spread out across many aggregation measuring components tied to those measuring component sets. This is where the composite aggregation functionality comes into play. The composite aggregation measuring component would reference the first three foundation aggregations and sum together the data from each aggregation into a single instance that represents the total usage for the period being settled.

Let’s look at a slightly more involved example. The diagram below illustrates a set of foundation aggregations based on:

- Rate Class
- Strata
- Procurement Class

This shows how the physical measuring component data (designated by Customer A, Customer B, etc.) will be aggregated into separate aggregation measuring components based on source of data (interval or scalar). Then it shows how those aggregations are then aggregated to produce a total number that represents all 3 aggregation sources merged together.

Copyright © 2026, Oracle and/or its affiliates.



Types of Aggregations

This section describes the processing used by different types of aggregations, including:

- Aggregating Measurement Data (Interval and Scalar)
- Aggregating Item Data
- Aggregating Billed Service Quantities
- Aggregating Aggregated Data (Composite Aggregation)

- Aggregating Measurement Data
- Aggregating Item Data
- Aggregating Billed Quantities
- Aggregating Aggregated Data

Aggregating Measurement Data

Aggregating measurement data involves identifying constituent measuring components for an Aggregation Measuring Component, identifying the measurement data that falls within the

Copyright © 2026, Oracle and/or its affiliates.

aggregation period, and aggregating that data. The process varies slightly between interval and scalar measuring components.

Interval Measuring Components

Aggregation of interval measurements is handled by the Aggregate Interval Measurements (D1_AGGINT) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm and populates the GTT1 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT1 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT1, the algorithm inserts its measurements for the aggregation period into the GTT2 global temporary table. If the SPI and UOM are different from that of the Aggregation Measuring Component, the measurements are converted to a common SPI using Axis Conversion.
- It performs aggregation calculations on the measurement data in the GTT2 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

Scalar Measuring Components

Aggregation of scalar measurements is handled by the Aggregate Scalar Measurements (D1_AGGSCL) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm and populates the GTT1 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT1 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT1, the algorithm inserts its measurements for the aggregation period into the GTT2 global temporary table. If the UOM is different from that of the Aggregation Measuring Component, the measurement is converted to a common UOM before being inserted into GTT1.
- It performs aggregation calculations on the measurement data in the GTT2 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It applies a profile to the aggregated scalar values to create interval measurements with the same SPI as the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.

Copyright © 2026, Oracle and/or its affiliates.

- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

Aggregating Item Data

Aggregating item data involves identifying constituent items for an Aggregation Measuring Component, calculate item measurement data for the aggregation period, and aggregating that data.

Aggregation of interval measurements is handled by the Aggregate Item Data (D1_AGGITEM) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Items (D1-FICONITEM) algorithm and populates the GTT3 global temporary table with groups of device types (for badged items) and/or service point types (for unbadged items) (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT3 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT3, the algorithm calculates measurements for the aggregation period and inserts them into the GTT4 global temporary table by applying the item counts for each group to the item interval (profile) data. If the SPI and UOM are different from that of the Aggregation Measuring Component, the measurements are converted to a common SPI using Axis Conversion.
- It performs aggregation calculations on the measurement data in the GTT4 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

Aggregating Billed Quantities

Aggregating billed service quantities involves identifying constituent usage transactions for an Aggregation Measuring Component, retrieving the service quantity data that falls within the aggregation period, and aggregating that data.

Aggregation of billed quantities is handled by the Aggregate Billed Service Quantities (D1_AGGBSQ) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Billing Data (D1-FINDCONSB) algorithm and populates the GTT5 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT5 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT5, the algorithm retrieves billing data for the aggregation period (based on the service quantity configuration on the data source) and inserts it into the GTT6 global temporary table. If the UOM is different from that of the Aggregation Measuring Component, the service quantities are converted to a common UOM.

Copyright © 2026, Oracle and/or its affiliates.

- It performs aggregation calculations on the measurement data in the GTT6 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It applies a profile to the aggregated values to create interval measurements with the same SPI as the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

Aggregating Aggregated Data

Aggregating aggregated data, also known as composite aggregation, involves identifying constituent aggregation measuring components for a composite Aggregation Measuring Component, identifying the measurement data that falls within the aggregation period, and aggregating that data.

Composite aggregation is very similar to aggregating interval measurement data, and uses the Aggregate Interval Measurements (D1_AGGINT) Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm.

Aggregation Measuring Components

- About Aggregation Measuring Components
- Maintaining Aggregation Measuring Components

About Aggregation Measuring Components

Aggregation Measuring Components are measuring components used to store aggregated data. Aggregation measuring components do not use initial measurements. Aggregated measurements are stored in the Measurement or Aggregation Measurement table (based on the Measurement Business Object defined for the aggregation measuring component's type).

Aggregation Measuring Components are defined by the following:

- **Initiating Measuring Component Set**: The measuring component set that triggered creation of the aggregation measuring component via the Dimension Scanning process (see Dimension Scanning for more information).
- **Processing Information**: Details of aggregation processing for the aggregation measuring component, including the date and time through which aggregation has been performed.
- **VEE Groups**: The VEE groups used when performing validation, editing, and estimation for the aggregation measuring component.
- **Measuring Component Attributes**: Attributes that describe the dimensions and criteria for the aggregation measuring component (identified by the Dimension Scanning process)

The base package includes the following aggregation measuring component business objects for use with dynamic aggregation:

Copyright © 2026, Oracle and/or its affiliates.

- Aggregator Interval (D1–AggregatorInterval): Used to aggregate interval measurements from interval measuring components
- Aggregator Scalar (D1–AggregatorScalar): Used to aggregate scalar measurements from scalar measuring components
- Aggregator Items Measuring Component (D1–AggregatorItems): Used to aggregate item data
- Aggregator Billed Service Quantities (D1–AggBilledServiceQuantities): Used to aggregate billed service quantities from usage transactions
- Aggregator Customer Quantities (D1–AggregatorCustomerQuantity): Used to aggregate specific types of customer quantities such as Peak Load Contribution (PLC) and usage factors
- Aggregator Composite (D1–AggregatorComposite): Used to aggregate aggregated data from aggregation measuring components (composite aggregation)

These business objects define the Find Constituents and Aggregation algorithms used for aggregation calculations.

Maintaining Aggregation Measuring Components

You use the **Totals and Trends View** portal to maintain aggregation measuring components.

Refer to About Aggregation Measuring Components for additional information.

You can access this portal by selecting an aggregation measuring component from the **Measuring Component Set Participants** zone in the **Measuring Component Set** portal. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Measuring Component**: This zone displays the basic attributes of the aggregation measuring component
- **Measuring Component Attributes:** This zone displays attributes that describe the dimensions and criteria for the aggregation measuring component
- **Related Measuring Component Sets**: This zone displays the measuring component sets related to the aggregation measuring component
- **Final Values Overlay**: This zone graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components that measuring different quantities, such as temperature.
- **Interval - Audit View**: This zone allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.
- **Measurements - Interval**: This zone displays measurement details for the aggregation measuring component
- **Usage Subscriptions Directly Linked to MC**: This zone displays usage subscriptions that are directly linked to the aggregation measuring component

Copyright © 2026, Oracle and/or its affiliates.

- **Time of Use Overlay**: This zone displays an overlay of the TOU periods on measurement data along with totalized TOU consumption based on a user-defined time period and TOU map

Master Data Synchronization

The synchronization of master data in Customer To Meter is covered in detail in the Customer To Meter Master Data Synchronization section of the *Oracle Utilities Customer to Meter Administrative User Guide*.

Outage Storm Mode

- About Outage Storm Mode
- Reviewing Widespread Outages

About Outage Storm Mode

Unfortunately, utilities and their customers are often subject to natural disasters that cause disruptions to AMI networks. These outage events can be related to hurricanes, tornadoes, ice storms, earthquakes, tsunamis, etc. During these severe outages, AMI systems do not always provide outage and restoration events from the devices. Without these outage and restoration events, Oracle Utilities Meter Data Management processes could estimate measurements that would be billed by the CIS. This will lead to unhappy customers, potentially bad press, and lots of manual cleanup work after the fact.

While this is a problem for major outage events, this can also occur with smaller outages (smaller blackouts, planned outages, etc.). This can potentially become an issue anytime the AMI network has a significant number of devices failing to communicate.

This module includes outage processing to improve estimation accuracy during major outages. The system will automatically stop estimations when the number of meter readings coming is less than a user defined tolerance. This tolerance is based on a significant drop in the read percentage for all devices in a similar postal code, service type, and Head End. Estimations will automatically resume when the device returns to normal operation. This can occur through a number of actions: read percentage for the surrounding area returns to normal, regular data is received from the device, or a power restoration device event is received from the device.

Oracle Utilities Meter Data Management also includes the ability to manually override "storm mode". This override will allow estimations to resume even if the outage hasn't been fully resolved. This will be useful if network fixes take too long and the utility has decided that customers should be billed on an estimate after a few days pass.

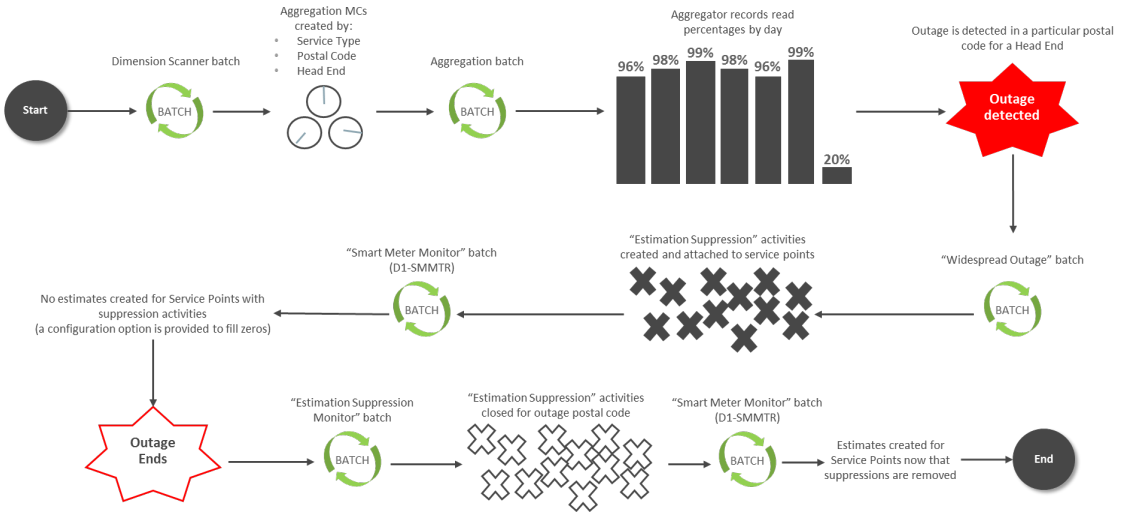
> **Note**
>
> For information on configuration, reference the related section for Outage Storm Mode in the Administrative User Guide.

Copyright © 2026, Oracle and/or its affiliates.

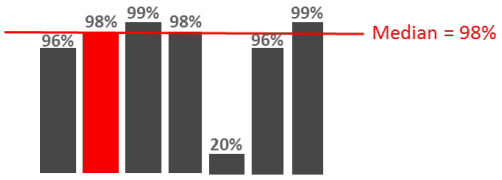
The diagram below illustrates the primary flow for Outage Storm Mode processing. The steps include:

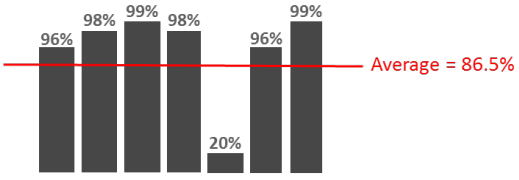
1. The **Dimension Scanner** batch (D1-ADS) monitors the **Meter Communication Tracking Dimension Scanner** business object. New Aggregator Measuring Components will be created for every applicable Service Type, Postal Code, and Head End in the system.
2. The **Aggregation Monitor** batch (D2-AGG) runs to execute the logic for the **Meter Communication Tracking Aggregator** Measuring Component. This logic will aggregate all meters for the defined Service Type, Postal Code, and Head End. Once the read percentage is found for that day then a measurement will be created for the Aggregator Measuring Component.
3. The **Widespread Outage** batch (D2-WSO) runs to determine if any of the "Meter Communication Tracking Aggregator" Measuring Components have dropped below the defined threshold on the MDM Master Configuration. If it has dropped enough, then an **Estimation Suppression** Activity is created for each of the affected Service Points and the "Meter Communication Tracking Aggregator" Measuring Component is set to an "Active Outage" indicator.
4. If the **Device Periodic Estimation** batch (D1-DVCPE) runs to generate estimates while an Estimation Suppression Activity is active for a Service Point, then the Service Point will either be skipped for estimation or will have zero consumption filled for all data with a condition code of "No Read - Outage Fill".
5. There are a number of methods available to resume normal operation and close out the Estimation Suppression Activity. These include the following:
    a. The read percentage reaches a "normal level" again as defined in the MDM Master Configuration. This will be detected by the **Widespread Outage** batch (D2-WSO).
    b. When non-estimated data is received from a device, this will close out an open Estimation Suppression Activity.
    c. If the **Standard Event Name** Extendable Lookup is configured with the End Estimation Suppression algorithm (D2-EN-ESTSUP), then any Power Up event this is configured for will close out an open Estimation Suppression Activity.
    d. Estimation Suppression Activities can also be closed manually be a user by navigating to the Activity, using the "Edit" button", and add an End Date.
6. The **Estimation Suppression Monitor** batch (D2-ESTSU) runs for all Estimation Suppression Activities to determine if the related "Meter Communication Tracking Aggregator" Measuring Component is no longer in a state of "Active Outage". If the widespread outage has ended, then the Estimation Suppression activity will be set with an end date.
7. The next time the **Device Periodic Estimation** batch (D1-DVCPE) runs it will generate estimated data as normally configured.

Copyright © 2026, Oracle and/or its affiliates.



In order to determine if an area is experiencing a widespread outage, the median read percentage of prior days is used as the basis for comparison. The median was chosen instead of the average to help reduce the impact of outlier values. This is illustrated in the example below.





Reviewing Widespread Outages

Anytime a **Meter Communication Tracking Aggregator** Measuring Component is marked with an indicator of "Active Outage", a To Do will be generated for user review. The base To Do Type for this is "Widespread Outage" (D2-WSOTD). This To Do is meant to serve as a warning to users that one of the areas in the service territory is experiencing a widespread outage.

If there is a desire to continue processing as normal and not have estimation suppressed, then you can override the Meter Communication Tracking Aggregator by hitting the "Force Complete Outage" button while reviewing the To Do. This will mark the Widespread Outage Indicator as "User Override" and will allow estimation to continue normally.

Dashboards

- About Dashboards
- Using the Customer Operational Dashboard
- Using the Meter Operational Dashboard
- Using the Service Order Operational Dashboard

Copyright © 2026, Oracle and/or its affiliates.

- Using the Service Order Trends Dashboard

About Dashboards

Dashboards provide a broad view of numerous metrics used with Oracle Utilities Customer To Meter. The data presented on each dashboard is tailored to provide operational as well as analytical information for management. On some graphs, drill down options are provided to retrieve more detailed information about the metrics provided.

For information on configuration, reference the related section for Defining Dashboard Options in the *Oracle Utilities Customer To Meter Administrative User Guide*.

Using the Customer Operational Dashboard

This portal is used to display the Customer Operational Dashboard.

You can access the portal from the **Main > Dashboards > Customer Operational Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The **Billing** tab displays the following zones:

- **Billing Trend**: displays bill completion counts and billed amounts by date.
- **Bill Print Trend**: displays bill print counts by date.
- **Open Bill Issues Trend**: displays open bill issue counts by date.

The **Payment** tab displays the following zones:

- **Payment Trend - All Sources**: displays payment counts and payment amounts from all sources.
- **Payment Trend - Upload**: displays upload payment counts and payment amounts.

The **To Do** tab displays the following zones:

- **To Do Completion By User**: displays To Do completion counts by user.
- **To Do Work Time By User**: displays average To Do work time by user.

The **Batch Performance** tab shows a series of zones related to any defined Performance Targets for batches. More information on Performance Targets can be found in the Framework guide.

Refer to About Dashboards for additional functional information. For information on configuration, refer to the related section for Configuring the Customer Operational Dashboard in the *Oracle Utilities Customer To Meter Administrative User Guide*.

Refer to each zone's embedded help for more information.

Using the Meter Operational Dashboard

This portal is used to display the Meter Operational Dashboard.

You can access the portal by selecting **Main , Dashboards, Meter Operational Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

Copyright © 2026, Oracle and/or its affiliates.

The **Meter Operational Dashboard** contains three tabs:

- Operations
- Batch Performance
- System Health Check

Using the Operations Tab

The **Operations** tab displays the following zones:

- **IMD Loading Trend**: displays the count of initial measurement data that's loaded each day by Head End system.
- **Device Event Loading Trend**: displays the count of device event data that's loaded each day by Head End system.
- **Incomplete IMDs by Status**: displays a set of counts for initial measurement data by status.
- **Incomplete IMDs Trend**: displays a set of counts for initial measurement data for each day by status
- **VEE To Dos by Exception Type**: displays a set of counts for VEE Exceptions by Exception Type that also still have a To Do that's incomplete.
- **VEE To Dos Trend**: displays a set of counts for VEE Exceptions by Exception Type for each day that also still have a To Do that's incomplete.
- **Usage Transactions by Status**: displays a set of counts for usage transactions by status.
- **Usage Transactions Trend**: displays a set of counts for usage transactions for each day by status.
- **Usage Transaction To Dos by Exception**: displays a set of counts for Usage Transaction Exceptions by Exception Type that also still have a To Do that's incomplete.
- **Usage Transaction To Dos Trend**: displays a set of counts for Usage Transaction Exceptions by Exception Type for each day that also still have a To Do that's incomplete.

Click **Configure** to navigate to the MDM Operational Dashboard Configuration master configuration. See **Configuring the Meter Operational Dashboard** in the *Administrative User Guide* for information regarding configuration of the dashboard.

Filtering by Division

Data displayed in the zones on the **Operations** tab (except the **IMD Loading Trend** and **Device Event Loading Trend** zones) can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to “No” the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked “Use As Primary” is the default.

- Using the Batch Performance Tab
- Using the System Health Check Tab

Using the Batch Performance Tab

Copyright © 2026, Oracle and/or its affiliates.

The **Batch Performance** tab shows a series of zones related to any defined batch-based Performance Targets

Click **Configure** to navigate to the **Performance Target** portal. See Measuring Performance in the *Administrative User Guide* for information.

Using the System Health Check Tab

The **System Health Check** tab includes a set of zones displaying data related to initial measurement processing and quality. These include:

- IMD Processing Trend
- IMD Processing Detail
- IMD Quality Trend
- IMD Quality Detail
- IMD Control Seeder Overview
- IMD Control Seeder Trend

Click **Configure** to navigate to the System Health Check Dashboard Configuration master configuration. See **Configuring the System Health Check Dashboard** in the *Administrative User Guide* for information regarding configuration of the dashboard.

IMD Processing Trend

The **IMD Processing Trend** zone displays the number of initial measurements and calculated numbers of constituent measurements for each day in the selected time period. The x-axis displays individual dates during the selected period. The left y-axis shows the average and maximum number of measurements per initial measurement. The right y-axis shows the number of initial measurements. The graph shows the following:

- **IMD Records**: The number of initial measurements processed on each day in the selected time period.
- **Average Measurements**: The average number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.
- **Maximum Measurements**: The maximum number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the largest number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.

You can filter the data displayed using the following controls:

- **Period View**: Specifies the range of data displayed. Options include Thirty Days, Seven Days, Three Days, or One Day. The default is "Seven Days".
- **Division**: Specifies the division of the initial measurements displayed in the graph. The default is "Show All".
- **Interval/Scalar**: Specifies whether the graph should display only interval initial measurements, scalar initial measurements, or both. The default is "Show All".

Copyright © 2026, Oracle and/or its affiliates.

- **Business Object**: Specifies the initial measurement business object(s) related to the initial measurements displayed in the graph. The default is "Show All".
- **Status**: Specifies the status (Pending, VEE Ready, Exception, Error, Finalized, and so on) of the initial measurements displayed in the graph. The default is "Show All".
- **Data Source**: Specifies the data source (if any) of the initial measurements displayed in the graph. The default is "Show All".

Click on the labels on the legend below the graph to hide/show specific metrics.

This zone displays data from statistics snapshots based on the IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly) business object. The **Snapshot Date/Time** drop-down list (in the lower right corner of the zone) lists all currently available snapshots based on this business object. Filtering options are based on data available in the snapshot. For example, if all of the initial measurements processed for a specific snapshot are all interval initial measurements, the "scalar" option wouldn't appear in the **Interval/Scalar** drop-down list.

Other metrics, including File Count (the number of payload files processed) and Total Measurements, Average Age, and Maximum Age can be viewed from the statistics snapshot.

Click **Capture** (next to **Snapshot Date/Time** in the lower right corner of the zone) to navigate to the **Statistics Control** portal where you can view a list of snapshots for the statistics control, view details of a specific snapshot, and create a new snapshot if needed.

IMD Processing Detail

The **IMD Processing Detail** zone displays details from the most recent snapshot based on the IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a unique combination of business object, interval/scalar flag, and status. This table includes the following:

- **Date Time**: The date and time on which the initial measurements captured in the row were processed
- **Business Object**: The business object of the initial measurements captured in the row
- **Interval/Scalar**: A flag designating if the data captured in the row is interval (D1IN) or scalar (D1SC)
- **Status**: The status of the initial measurements captured in the row
- **Description**: A description of the initial measurements captured in the row
- **IMD Records**: The number of initial measurements processed for the unique combination of business object, interval/scalar, and status captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by IMD Records) per initial measurement captured in the row
- **Maximum Measurements**: The maximum number of measurements per initial measurement captured in the row
- **Total Measurements**: The total number of measurements values in the initial measurements captured in the row

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

IMD Quality Trend

The **IMD Quality Trend** zone displays the number of Validation, Editing, and Estimation (VEE) exceptions and calculated numbers of measurements for each day in the selected time period. The x-axis displays individual dates during the selected period. The left y-axis shows the

Copyright © 2026, Oracle and/or its affiliates.

average and maximum number of measurements per initial measurement. The right y-axis shows the number of VEE exceptions. The graph shows the following:

- **VEE Exceptions**: The number of VEE exceptions processed on each day in the selected time period.
- **Average Measurements**: The average number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.
- **Maximum Measurements**: The maximum number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the largest number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.

You can filter the data displayed using the following controls:

- **Period View**: Specifies the range of data displayed. Options include Thirty Days, Seven Days, Three Days, or One Day. The default is "Seven Days".
- **Division**: Specifies the division of the initial measurements displayed in the graph. The default is "Show All".
- **Interval/Scalar**: Specifies whether the graph should display data for only interval initial measurements, scalar initial measurements, or both. The default is "Show All".
- **Business Object**: Specifies the initial measurement business object(s) related to the data displayed in the graph. The default is "Show All".
- **Excepti0on Type**: Specifies the exception type (Insufficient Input Data, Device Identifier Discrepancy, and so on) of the VEE exceptions displayed in the graph. The default is "Show All".
- **Data Source**: Specifies the data source (if any) of the the data displayed in the graph. The default is "Show All".

Click on the labels on the legend below the graph to hide/show specific metrics.

This zone displays data from statistics snapshots based on the IMD Quality Snapshot - Hourly (D1-IMDQualityHourly) business object. The **Snapshot Date/Time** drop-down list (in the lower right corner of the zone) lists all currently available snapshots based on this business object. Filtering options are based on data available in the snapshot. For example, if all of the VEE exceptions processed for a specific snapshot are all based on interval initial measurements, the "scalar" option wouldn't appear in the **Interval/Scalar** drop-down list.

Other metrics, including File Count (the number of payload files processed) and Total Measurements, Average Age, and Maximum Age can be viewed from the statistics snapshot.

Click **Capture** (next to **Snapshot Date/Time** in the lower right corner of the zone) to navigate to the **Statistics Control** portal where you can view a list of snapshots for the statistics control, view details of a specific snapshot, and create a new snapshot if needed.

IMD Quality Details

The **IMD Quality Detail** zone displays details from the most recent snapshot based on the IMD Quality Snapshot - Hourly (D1-IMDQualityHourly) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a

Copyright © 2026, Oracle and/or its affiliates.

unique combination of business object, interval/scalar flag, and status. This table includes the following:

- **Date Time**: The date and time on which the VEE exceptions captured in the row were processed
- **Business Object**: The business object of the initial measurements on which the VEE exceptions captured in the row are based
- **Interval /Scalar**: A flag designating if the VEE exceptions captured in the row are based on interval (D1IN) or scalar (D1SC) initial measurements
- **Exception Type**: The exception type of the VEE exceptions captured in the row
- **Description**: A description of the VEE exception type captured in the row
- **VEE Exceptions**: The number of VEE exceptions processed for the unique combination of business object, interval/scalar, and exception type captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by VEE Exceptions) per initial measurement on which the VEE exceptions captured in the row are based
- **Maximum Measurements**: The maximum number of measurements per initial measurement on which the VEE exceptions captured in the row are based
- **Total Measurements**: The total number of measurements values in the initial measurements on which the VEE exceptions captured in the row are based

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

IMD Control Seeder Overview

The **IMD Control Seeder Trend** zone displays details from the most recent snapshot based on the IMD Control Seeder – Hourly (D1-IMDCtrlSeeder) business object. By default this zone displays a pie chart of the data from the snapshot, with each segment in the chart representing a unique combination of business object, interval/scalar flag, and message category and number.

To view a table version of this data, click the **Grid** icon (to the upper right of the table). The tabular version of this data can also be seen in the **IMD Control Seeder Trend** zone.

IMD Control Seeder Trend

The **IMD Control Seeder Trend** zone displays details from the most recent snapshot based on the IMD Control Seeder – Hourly (D1-IMDCtrlSeeder) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a unique combination of business object, interval/scalar flag, and message category and number. This table includes the following:

- **Date Time**: The date and time on which the initial measurements captured in the row were processed
- **Business Object**: The business object of the initial measurements captured in the row
- **Interval/Scalar**: A flag designating if the data captured in the row is interval (D1IN) or scalar (D1SC)
- **Message Category**: The message category for the exceptions associated with the IMD seeders captured in the row
- **Message Number**: The message number for the exceptions associated with the IMD seeders captured in the row

Copyright © 2026, Oracle and/or its affiliates.

- **Message Text**: The message text for the exceptions associated with the IMD seeders captured in the row
- **IMD Records**: The number of initial measurements processed for the unique combination of business object, interval/scalar, and status captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by IMD Records) per initial measurement captured in the row
- **Maximum Measurements**: The maximum number of measurements per initial measurement captured in the row
- **Total Measurements**: The total number of measurements values in the initial measurements captured in the row

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

Using the Service Order Operational Dashboard

This portal is used to display the Service Order Operational Dashboard.

You can access the portal from the **Main > Dashboards > Service Order Operational Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The following zones may appear as part of the portal's **Main** tab page:

- **In-Flight Activities By Type**: shows the distribution of in-flight Activities (in progress or in error) by Activity Type.
- **In-Flight Activity Trend**: shows the number of activities that are in-flight on a given day.
- **In-Flight Activities with Issues**: presents a bar chart that indicates the number of activities with issues.
- **In-Flight Activity Status Distribution**: presents a pie chart that displays the distribution of statuses of in-flight activities.
- **Efficiency Trend for In-Flight Activities**: presents a stacked bar chart that displays the distribution of in-flight activity processing efficiency relative to configurable tolerance.
- **Service Order To Do Summary**: this zone uses horizontal bars to highlight the relative number of open To Dos for a set of To Do Types related to Service Order Management.

Refer to About Dashboards for additional functional information. For information on configuration, refer to the related section for Configuring the Service Order Operational Dashboard in the Administrative User's Guide.

Filtering by Division

Data displayed in the zones on this dashboard can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to "No" the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked "Use As Primary" is the default.

Using the Service Order Trends Dashboard

Copyright © 2026, Oracle and/or its affiliates.

This portal is used to display the Service Order Operational Dashboard.

You can access the portal from the **Main > Dashboards > Service Order Trends Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The following zones may appear as part of the portal's **Main** tab page:

- **Orchestration Issues Trend**: contains a stacked bar chart showing issues by category and Activity Type.
- **Issue Distribution Over Time**: contains a pie chart showing the distribution of activity issues over a range of days.
- **Activity Success Trend**: shows successful activities over a range of days.
- **Activity Creation Trend**: shows activities created over a range of days.
- **Activity Completion Efficiency Over Time**: shows the efficiency of activity completion over a range of days.

Refer to About Dashboards for additional functional information. For information on configuration, refer to the related section for Configuring the Service Order Trends Dashboard in the Administration Guide.

Filtering by Division

Data displayed in the zones on this dashboard can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to “No” the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked “Use As Primary” is the default.

Totals and Trends

- Processing Statistics
- Consumption Statistics

Processing Statistics

- About Processing Statistics
- About Head-End System Processing Statistics

About Processing Statistics

Upload statistics (or upload stats) are statistics related to the uploading of initial measurement data and device events sent from a head-end system.

Upload statistics are defined as activities in Oracle Utilities Smart Grid Gateway.

There are three types of upload statistics activities:

Copyright © 2026, Oracle and/or its affiliates.

Payload Statistics: Contains statistics related to a specific payload (file) containing one or more initial measurements or device events. Payload Statistics activities contain:

- Basic information about the payload (head-end system, file name, and status
- Middleware statistics including specifics about the file, the total number of initial measurements or device events processed, the number of initial measurement or device events errors, and total processing time
- Initial measurement statistics including the number of initial measurements processed
- Device event statistics including the number of device events processed

Payload Error Notification: Contains details concerning processing errors encountered in an individual payload (file) containing one or more initial measurements or device events. Payload Error Notification activities are related to Payload Statistics activities.

Payload Summary: Contains processing summary statistics for an individual payload (file) containing one or more initial measurements or device events. Payload Summary activities are related to Payload Statistics activities, and are used to update related payload statistics upon the completion of payload processing.

Upload statistics activities are created during processing of payload files as follows:

- When processing begins for a payload, a Payload Statistics activity is created to record the process.
- If an error occurs during processing, a Payload Error Notification activity is created.
- When payload processing is complete, a Payload Summary activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload, including (the start and end time of the processing, the total processing time, the number of initial measurements or device events processed, and the number of initial measurement or device event errors (if any).

About Head-End System Processing Statistics

Head-end system processing statistics are summarized processing statistics for payloads received from a given head-end system.

As Oracle Utilities Smart Grid Gateway processes payloads containing initial measurements or device events, statistics for each payload are captured in Payload Statistics activities. Over time, payload statistics for each head-end system are summarized to allow administrators to view statistics summary for the head-end system.

Head-end system processing statistics are stored as aggregated measurements for aggregator measuring components. A separate aggregator measuring component must be set up for each head-end system for which processing statistics will be aggregated.

Consumption Statistics

Please refer to the About Aggregation section for information on consumption statistics.

Time Zone Support

- Daylight Savings Time Support

Copyright © 2026, Oracle and/or its affiliates.

- Multiple Time Zone Support

Daylight Savings Time Support

This section describes how the Oracle Utilities Service and Measurement Data Foundation and its related products support Daylight Saving Time (DST).

All date/times in the system are displayed adjusted for daylight savings time (when in effect). However, not all date/times in the system are stored in daylight savings time:

- Stored in Standard (not adjusted for DST): data that is highly sensitive to DST shifts are stored in standard time without any shifts, this means they have a consistent 24 hour day each day all year and avoid issues that arise from a skipped hour when transitioning to DST and a duplicate hour when transitioning out of DST. For example: initial measurement data, final measurements, device events, TOU map data.
- These date/times are adjusted for daylight savings time prior to being displayed to a user.
- Stored in Legal (adjusted for DST): data that is not as sensitive to DST time shifts are stored in legal time. For example, device configurations, service points, installation events, usage subscriptions, usage transactions.

Adjusting Date/times Upon Receipt

The two largest sources of imported data to the system can be sent in a mixture of date/time formats some in legal time some in standard time:

- Initial Measurement Data
- Device Events

Since both sets of data are sent by head end systems and each head end system handles date/times differently, potentially even differing by device, the interface must be extremely flexible in how it handles date/times. In order to ensure data is accurately stored and processed the system must know how a given meter sends date/time information. This is defined on the device through the Incoming Data Shift field:

- Always in Standard Time: No special handling is required.
- Always in Legal Time: All date/times for the initial measurement data or device event will be converted from legal time into standard time.
- In circumstances where a date/time is received in legal time and it falls on the duplicate hour that occurs when DST ends special logic will be executed to identify whether that hour is the first (still in DST) or second (no longer in DST) of the duplicate hours. If for some reason it cannot be identified which hour it is the transaction will go into error.

Daylight Savings Time Interval Conversion Examples

The following examples show how data received legal time will be shifted into standard time for an interval meter that is read on an hourly basis.

Daylight Savings not in Effect

Date/Time Received in Legal	Date/Time Stored in Standard
01/14/2010 12:00AM	01/14/2010 12:00AM
01/14/2010 1:00AM	01/14/2010 1:00AM
01/14/2010 2:00AM	01/14/2010 2:00AM

Copyright © 2026, Oracle and/or its affiliates.

01/14/2010 3:00AM	01/14/2010 3:00AM
01/14/2010 4:00AM	01/14/2010 4:00AM
01/14/2010 5:00AM	01/14/2010 5:00AM

Daylight Savings Starts (Spring Forward) - No 2AM interval

Date/Time Received in Legal	Date/Time Stored in Standard
03/14/2010 12:00AM	03/14/2010 12:00AM
03/14/2010 1:00AM	03/14/2010 1:00AM
03/14/2010 3:00AM	**03/14/2010 2:00AM**
03/14/2010 4:00AM	03/14/2010 3:00AM
03/14/2010 5:00AM	03/14/2010 4:00AM
03/14/2010 6:00AM	03/14/2010 5:00AM

Daylight Savings in Effect

Date/Time Received in Legal	Date/Time Stored in Standard
06/14/2010 12:00AM	06/13/2010 11:00PM
06/14/2010 1:00AM	06/14/2010 12:00AM
06/14/2010 2:00AM	06/14/2010 1:00AM
06/14/2010 3:00AM	06/14/2010 2:00AM
06/14/2010 4:00AM	06/14/2010 3:00AM
06/14/2010 5:00AM	06/14/2010 4:00AM

Daylight Savings Ends (Fallback) - Duplicate 1AM

Date/Time Received in Legal	Date/Time Stored in Standard
11/7/2010 12:00AM	11/6/2010 11:00PM
11/7/2010 1:00AM	**11/7/2010 12:00AM**
11/7/2010 1:00AM	**11/7/2010 100AM**
11/7/2010 2:00AM	11/7/2010 2:00AM
11/7/2010 3:00AM	11/7/2010 3:00AM
11/7/2010 4:00AM	11/7/2010 4:00AM

Daylight Savings Impact on Periodic Estimation

In periodic estimation both the interval and scalar versions can be impacted by daylight savings time:

- Interval periodic estimation using the "Predictable Cut-off" method.
- All scalar periodic estimation

Refer to About Periodic Estimation for more information about the impact of Daylight Savings Time on periodic estimation.

Multiple Time Zone Support

Copyright © 2026, Oracle and/or its affiliates.

Some utilities have operations in more than a single time zone. When operating in multiple time zones, Oracle Utilities Meter Data Management and Smart Grid Gateway provide the following high level functionality:

- The ability to receive data in any time zone and convert it to the common time zone used by Oracle Utilities Meter Data Management (as defined by the base time zone)
- The ability to display functional date/times in the time zone of a given device, service point, usage subscription
- The ability to define what time zone a measuring component, device configuration, service point, or usage subscription resides in
- The ability to aggregate data across time zones into a single time zone or without respect to a time zone (see Aggregations on page E-17 for more information).
- The ability to export data to an external system with the appropriate time zone information to enable the external system to consume data from different service points in different time zones

By default, the system assumes only a single time zone is supported. To enable support for multiple time zones it requires that adding the appropriate entry to the general system configuration feature configuration.

Refer to Configuring Feature Configurations for additional on the feature configuration options available.

Date/Time Storage and Display

All date/times in the system are stored in the base time zone. Date/times are stored in either standard time or legal time and this determination is made based on the type of data (e.g. measurements are in standard but installation event date/times are in legal).

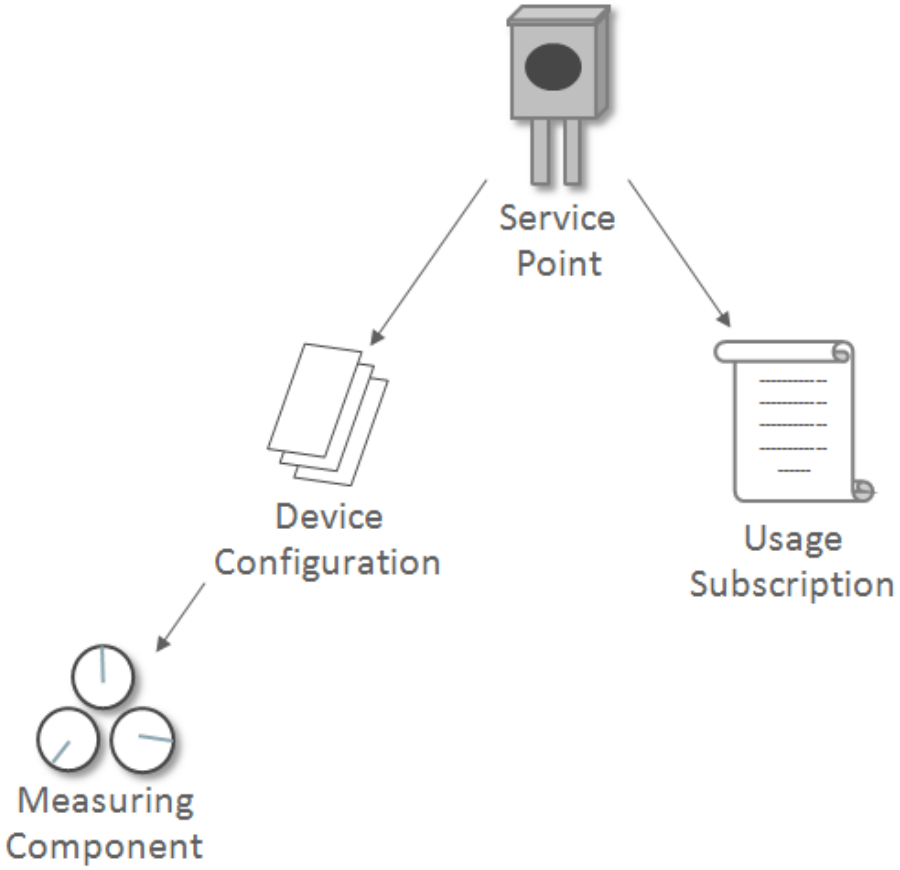
All date/times are displayed in legal time. Those date/times that are key functional date/times related to a device, service point, or usage subscription will be displayed in the legal time of the time zone the device resides in.

Refer to the Glossary of Terms for a more detailed definition of Standard and Legal time.

Time Zone Hierarchy and Maintenance

A time zone can be defined at many points within the device to usage subscription hierarchy. This enables the time zone to be accurately portrayed based on the installation status of that device (e.g. not installed, installed but not billed, billed). At each stage of the hierarchy the time zones must match.

Copyright © 2026, Oracle and/or its affiliates.



Once the hierarchy has been established no one portion of that hierarchy can be changed to a time zone that differs from the other levels of the hierarchy. In order to change the time zone for an established device to usage subscription hierarchy the change must be done to the service point. Once the service point time zone has been changed the system will cascade that change to all other levels of the hierarchy.

If a device has been installed at a service point in order for it to be moved to a different service point in a different time zone. It will require that a new device configuration be created for time zone of the new service point since a device configuration cannot be associated to two different service points each with a different time zone from the device configuration.

Note: If the time zone for a device that has existing measurements has been changed there will be no automatic adjustments made to the underlying measurement data. If the measurement data must be changed it will require manual intervention.

Note: Although measuring component has time zone it is not populated on physical devices that are installed at service points. This time zone is used for standalone measuring components.

Impacts on Aggregations

In aggregation there are two use cases for how data should be aggregated for service points that exist in differing time zones:

- **Absolute Time**: If a time zone is provided on the measuring component or measuring component type, then the aggregation horizon date/times will be used across service points of differing time zones. For example the 4/1/2010 12AM interval in US/Pacific would be combined with the 4/1/2010 3AM interval in US/Eastern. Since all date/times are stored in the base time zone there is no need to do time zone conversions.

Copyright © 2026, Oracle and/or its affiliates.

- **Local Time**: If no time zone is specified, consumption is aggregated by converting the aggregation horizon to the appropriate time zone for each service point. For example the 4/1/2010 12AM interval in US/Pacific would be combined with the 4/1/2010 12AM interval in US/Eastern.

Aggregation Measuring Components in general do not have any time zone related dependencies with other master configuration objects. However, when an aggregator is listed as a direct channel measuring component on a usage subscription it share the same time zone as the Usage Subscription.

Impacts on TOU Map Generation

The date and times defined on a Time of Use Mapping Templates (TOU) do not have a time zone associated to them and will be interpreted based on the time zone of the TOU map type of the specific TOU Map being processed. This allows for a single definition of a TOU schedule that can be applied across many different time zones.

During the generation of the TOU map data for a given TOU map template the date/times will be converted from the time zone of the TOU map to the time zone defined in the base time zone defined on the installation options - framework. For example, if 1/1/2010 5:00PM was considered to be "On Peak" and the system time zone was UTC -08:00 and the TOU map time zone was UTC -05:00 then the TOU map data entry would be written as 1/1/2010 2:00PM.

Impacts on Factors

Factor effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example a factor effective 4/1/2010 12:00AM would be considered to be effective 4/1/2010 12:00AM UTC -08:00 and 4/1/2010 12:00AM UTC -05:00 based on the time zones of the respective meters. This allows a single factor to apply to many time zones and behave uniformly.

Impacts on Holidays

Holiday effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example a holiday on 12/25/2010 would be considered to be effective beginning 12/25/2010 12:00AM UTC -08:00 and 12/25/2010 12:00AM UTC -05:00 based on the time zones of the respective meters.

Impacts on Periodic Estimation

Refer to About Periodic Estimation for more information about the impact of Daylight Savings Time on periodic estimation.

Impacts on Initial Measurement Creation Online

When creating initial measurements online from the 360 Degree View portals, initial measurements are created as appropriate for measuring components with time zones other than the base time zone.

For example, if you copy measurements from a measuring component in one time zone to a measuring component in a different time zone, the measurements are being moved in absolute time (i.e. the 3:00 PM in UTC -08:00 would be changed 6:00PM in UTC -05:00).

Impacts on Items

The consumption for "badged" and "unbadged" items is driven by configuration on the device type service quantity and the multi-item section of the service point:

- **Device Type Service Quantity**: the effective date of the service quantity does not have a time zone and is assumed to be in the local time zone of the service point it is associated

Copyright © 2026, Oracle and/or its affiliates.

to (whether through a install event for "badged" items or by being directly configured on the service point). For example an effective date time of 1/1/2010 5:00PM would be considered to be effective beginning 1/1/2010 5:00PM UTC -08:00 and 1/1/2010 5:00PM UTC -05:00 based on the time zones of the respective meters

- **Service Point Multi-Item**: the start date time and end date time of the multi-item list are stored in legal time of the base time zone and displayed in the legal time of the service point's time zone.

Service Issue Monitors

- About Service Issue Monitors
- About Service Investigative Orders

About Service Issue Monitors

Service Issue Monitors are service tasks that analyze service points to determine if service is needed. If service is determined to be needed, the Service Issue Monitor creates a Service Investigative Order.

Device events, VEE exceptions, failed smart meter commands, and other events can trigger the creation of a service issue monitor (the type of service issue monitor created is based on the Service Issue Monitor Type specified on the device event type, exception type, or activity type). Once created, service issue monitors analyze the service point where the device associated with the device event, VEE exception, or failed command, based on evaluation criteria specified on the service issue monitor's type. If the criteria are met (in other words, if a specified number of command failures, device events, or VEE exceptions are found for the service point), the service issue monitor creates a service investigative order.

Service Issue Monitor Creation

This section provides an overview of the process that takes place when service issue monitors are created.

> **Note**
>
> The process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

1. One (or more) of the following takes place:
   - A device event or reader remark is received, whose type specifies a Service Issues Monitor Type
   - A VEE exception whose type specifies a Service Issues Monitor Type is generated during VEE processing
   - A usage transaction exception generated during usage calculation
   - A smart meter command fails whose type specifies a Service Issues Monitor Type
   - A business flag is received whose type specifies a Service Issue Monitor Type

Copyright © 2026, Oracle and/or its affiliates.

2. The object received in item (1) creates a service issue monitor based on the referenced Service Issue Monitor Type
3. The service issue monitors analyzes the service point based on the Discard Rules and Evaluation Criteria defined on the service issue monitor's type to determine if a service investigative order should be created or if the service issue monitor should be discarded.

   The "Discard Rules" on the service issue monitor's type are evaluated to determine if the new service issue monitor should be discarded (new service issue monitors are always discarded when created if an existing service investigative order initiated by a service issue monitor of the same type exists for the service point.)

   If the service issue monitor type was configured to require approval, the service issue monitor generates a To Do to notify the appropriate individuals that approval is necessary.

   If the service issue monitor is approved or if no approval was necessary the service investigative order will be continue.
4. The service issue monitor then executes the evaluation criteria defined on the service issue monitor's type. If those criteria are met the Service Investigative Order activity is created. If the criteria are not met the service issue monitor is discarded.
5. If the service issue monitor has completion criteria defined the service issue monitor will wait for the service investigative order to complete. Once complete the the service issue monitor will execute the completion criteria to evaluate if the service investigative order achieved the desired results.

   If either there were no completion criteria or they were satisfied the service issue monitor will complete.

   If the completion criteria were not satisfied the service issue monitor enter an inconclusive investigation state from which manual intervention will be required.

About Service Investigative Orders

Service investigative orders are activities created by a service issue monitor when a specified set of events have occurred at a service point. The type of activity created by the service issue monitor is defined on the service issue monitor's type.

Service issue monitors are often configured to create field activities that are in turn sent to an external field work system.

Market Transaction Management Addendum

This chapter is an addendum to the Market Transaction Management chapter. It describes functionality that is specific to Oracle Utilities Customer to Meter.

- Maintaining Market Usages, Market Charges and Market Payments

Maintaining Market Usages, Market Charges and Market Payments

The following sections describe the pages available to view and work on market usage, market charge and market payment.

- Maintaining Market Usages

Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Market Charges
- Maintaining Market Payments

Maintaining Market Usages

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Usage.**You are brought to a query portal with options for searching for market usages.

Once a market usage record has been selected, you are brought to the maintenance portal, to view the selected record.

Maintaining Market Charges

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Charge.**You are brought to a query portal with options for searching for market charges.

Once a market charge record has been selected, you are brought to the maintenance portal, to view the selected record.

Maintaining Market Payments

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Payment.**You are brought to a query portal with options for searching for market payments.

Once a market payment record has been selected, you are brought to the maintenance portal, to view the selected record.

Market Transaction Management - US Distribution

Market Transaction Management - US Distribution refers to functionality distributors can use to exchange standardized electronic business transactions in deregulated US distribution markets. The functionality supports the processing of Electronic Data Interchange (EDI) transactions that enable the distributor to communicate operational, customer, billing, enrollment, and service-related information with market participants in a consistent and reliable way.

This functionality (also referred to as "market transaction messaging") provides a common baseline architecture and transaction-processing framework that can be extended to support the specific rules, transaction sets, and validation requirements of individual deregulated distribution markets. The functionality provided is a framework on which specific market implementations can be based. The business objects and algorithms represent a generic "root" and "lowest common denominator" implementation, meaning they serve as the basis upon which a market-specific versions of those objects and algorithms can be created. For example, market-specific versions of the business objects and algorithms used for a PJM

Copyright © 2026, Oracle and/or its affiliates.

implementation of a market invoice (810) transaction would differ from those used for an ERCOT implementation, even though they would be based on the same root objects.

> **Note**
>
> The US Distribution market transaction messaging functionality is intended for use specifically by **distributors** working in a deregulated market within the United States.
>
> This documentation assumes readers have a basic understanding the market transaction messaging business process, including understanding Electronic Data Interchange (EDI) transactions, different types of market transactions, and the different types of market participants in a deregulated market.
>
> It is CRITICAL that you have a firm understanding of how your market operates before you begin planning your implementation. This includes, but is not limited to, details concerning the types of market participants involved, the billing relationship between the distributor and other market participants, and the types and formats of the market messages the distributor will send to and/or receive from other market participants.
>
> Objects provided with MTM are prefixed with "U2". The business objects provided are intended to serve as parent business objects for market-specific versions. None of the business objects provided can be instantiated.
>
> Note that the information provided in this is not sufficient for configuring any specific market. Use of this functionality requires implementers and users have an understanding of their own market.

- Overview
- The Importance of Understanding Your Market
- Customer Lifecycle Example
- Market Transaction Management and Customer Cloud Service
- Implementing an MTM US Distribution Transaction

Overview

In deregulated utility markets, distributors exchange large volumes of business transactions with retail energy providers, market operators, and other participants to support customer choice and delivery operations.

Market transaction management helps the distributor:

- Exchange standardized EDI market transactions
- Automate communication with external market participants
- Support customer enrollment, switching, and service processing
- Manage service requests and account updates
- Process billing and usage information
- Comply with regional market rules and transaction standards
- Reduce manual processing and operational errors

Copyright © 2026, Oracle and/or its affiliates.

The framework provided offers reusable core functionality while allowing market-specific customization and configuration.

Supported transaction types can vary by market implementation, but commonly include:

- Customer enrollments and drops
- Service starts, stops, and transfers
- Meter usage and consumption data
- Billing and payment information
- Customer account updates
- Service order requests
- Move-in and move-out requests
- Transaction acknowledgments and responses

Market-Specific Implementations

Although deregulated markets share many common business processes, each market operates under its own regulatory requirements, transaction standards, and implementation guidelines.

Market transaction messaging provides a common foundational framework that supports:

- Reusable transaction-processing functionality
- Shared integration patterns
- Standardized monitoring and error handling
- Configurable validation and business rules

This approach enables distributors to implement market-specific requirements while maintaining consistency across multiple deregulated markets.

Glossary

The table below lists a number of terms used in market transaction messaging.

Term	Definition
Supplier	The utility responsible for supplying power to the grid Also referred to as: • Electric Generation Suppliers (EGS)
Distributor	The utility responsible for distribution of power within the market. Also referred to as: • Electric Distribution Companies (EDCs) • Local Distribution Company (LDC) • Transmission and Distribution Service Provider (TDSP)

Copyright © 2026, Oracle and/or its affiliates.

Term	Definition
Retailer	The company with which customers enroll to purchase electricity Also referred to as: • Competitive Retailer (CR) • Energy Service Company • Energy Service Provider (ESP) • Retail Electric Provider (REP)
Market Operator	The organization in charge or managing and operating the market. Also referred to as: • Independent System Operator (ISO) • Regional Transmission Organization (RTO)
Service Provider	A generic term for an entity within a market.
Market Participant	A company participating in the electricity market.
Transaction	A message sent between market participants to perform business processes, such as customer enrollment or switching, sending usage, sending charges, sending payments, etc. See Supported Transactions for information about the specific types of transactions provided with MTM USD.

Market Transaction Management Data and Objects

This section provides a high-level overview of the data and objects used by the Market Transaction Management US Distribution functionality.

Markets

The Market object is used to define a deregulated market which is a geographical area operating with a set of rules to manage premises and/or customers for a service type. Separate market instances are created for each geographic area and service type.

Market Participants

Market Participants are participants in a deregulated environment. Relationships between market participants are defined in a particular market record.

Inbound Market Messages

Inbound Market Messages capture instances of market messages received from an external source.

Outbound Market Messages

Outbound Market Messages capture instances of market messages sent from the system to an external source.

Market Message Type

Market Message Type are used to define the different types of market messages, both inbound and outbound, and to define the appropriate market message business object to use when creating objects of this type.

Market Processes

Market Processes capture instances of market processes that may be created to carry out business logic related to a specific inbound or an outbound market message.

Copyright © 2026, Oracle and/or its affiliates.

Market Process Types

Market Process Type are used to define configuration related to processes that may be initiated to support steps that should be execute to support an inbound or an outbound market message. It defines the market process business object to use when creating market processes of this type. It may also define one or more market process event business objects to initiate events that are generated to support steps within the market process.

Market Process Events

Market Process Events are used to orchestrate individual events that may occur to support steps in a given market process, such as creating specific records to define a customer (account, premise, service point, service agreement, and so on).

Market Usage

Market Usage records are used to store usage based on market codes and values. This is typically for current data, as it's received into the system. Historical Market Usage records are created only on request as needed.

Market Charges

Market Charges are used to store charges based on market codes and values. These are typically the charges that appear on a customer bill. When a bill is canceled, the charges are pulled from here.

Market Payments

Market Payments are used to store and capture individual payments received/sent in an "820 Payment Order/Remittance Advice" transaction,

Market Configuration

Market Configurations are used to capture configuration needed to support specific types of market transactions. Each type of configuration use a specific business object that defines the information captured. Each type of configuration may define up to five key fields that may be used to uniquely identify an individual configuration instance.

Registration Points

A registration point is the single point at which a premise is registered in the system. Registration points operate at different levels in different markets. Some are based on the premise, some on the combination of premise and person, some on the meter ID, and so on.

Supported Transactions

MTM USD supports the following types of market transactions.

Copyright © 2026, Oracle and/or its affiliates.

EDI Transaction Number	Name	Description
EDI 814	General Request/Response	Used to manage customer enrollment, account maintenance, drops, move-ins, and supplier switches. Supported request types include: • Historical Usage Request • Enrollment Request • Change Request • Drop Request • Service Order Request • Move Out Request • Price Code Request
EDI 824	Application/Advice	Used to transmit notifications or rejections regarding the status of a previously sent transaction (such as accepting or rejecting an 814 enrollment).
EDI 867	Product Transfer Account Adjustment / Usage	Used to transfer historical customer load data (when requested by a supplier) and routine, actual or estimated monthly metered consumption to the Energy Service Provider (ESP).
EDI 810	Invoice	Used to transmit bill amounts, monthly usage, and charge components. Depending on the market, it is sent from the utility to the supplier or vice versa.
EDI 820	Payment Order/Remittance Advice	Used to transfer payment and allocation details between utilities, retail suppliers, or financial institutions, particularly in dual-billing scenarios or consolidated billing.
EDI 650	Maintenance Service Order	Used to manage, request, and report maintenance-related actions
EDI 248	Account Assignment/Inquiry and Service/Status	Used to assign accounts, update account details, and report on service or collection statuses

The Importance of Understanding Your Market

As noted previously, it is CRITICAL that you have a firm understanding of how your market operates before you begin planning your implementation.

MTM is designed as a configurable framework rather than a market-specific application. Although it provides common processes for enrollment, billing, payments, customer changes, and retailer communications, the business rules governing those processes vary considerably from one market to another.

Copyright © 2026, Oracle and/or its affiliates.

Many implementation decisions—including customer setup, transaction processing, billing behavior, and registration point design—depend on market requirements rather than product configuration alone.

Successful MTM implementations begin with an understanding of the market's business rules. The application provides the framework for implementing those rules but does not prescribe how every market should operate.

Types of Market Participants

Within most markets there are three primary types of participants:

- **Distributor**: This is you, the utility implementing MTM US Distribution. In most markets, the Distributors own the transmission system (including substations, transformers, and wires) as well as the meters installed at customer service points.

  > **Note**
  >
  > When defining Billing Options, "We" and "Us" refers to the Distributor. In the system, the Distributor us defined as a Market Participant service provider.

- **Retailers**: Retailers are organizations that sell electricity to customers. In the system, retailers are defined as Market Participant service providers.

  > **Note**
  >
  > When defining Billing Options, "Them" and "They" refers to Retailers.

- **Customers**: Customers are the residential and commercial/industrial entities that purchase power from a retailer. In the system Customers are represented by a Registration point.

Note that many markets have other market participants, such as Independent System Operators (ISOs) or Regional Transmissions Organizations (RTOs), but the focus of market transaction messaging for US Distribution is between the Distributor and Retailers.

Why Market Knowledge Matters

Unlike many enterprise applications, MTM does not assume a single business model. Different markets define:

- Customer ownership
- Retailer responsibilities
- Distributor responsibilities
- Billing relationships
- Enrollment procedures
- Transaction timing
- Customer switching rules
- Payment settlement
- Registration point identification

These differences influence almost every aspect of an implementation. For example, one market may require the distributor to create customer records during enrollment, while another assumes the customer already exists. Likewise, some markets require retailer settlement

Copyright © 2026, Oracle and/or its affiliates.

based on customer payment, while others require settlement based on bill due dates. Understanding these differences before implementation helps ensure that the application is configured to support the correct business processes.

Markets Differ in More Than Geography

Markets should not be viewed simply as geographic regions. A market defines the business rules under which distributors, retailers, and customers interact. Those rules determine:

- Which organization owns customer information
- Which organization bills the customer
- Which transactions are exchanged
- When transactions are exchanged
- Which participant is considered the system of record for different types of information

Even distributors operating within the same state may implement different business practices to satisfy their market obligations.

Examples of Market Differences

The following examples illustrate how business rules vary from market to market.

Texas

Texas represents a fully deregulated retail market. Characteristics of this market include:

- Retailers own the customer relationship.
- Customers may switch retailers outside normal billing cycles.
- Enrollment often creates the customer within the distributor's system.
- Retailers generally bill customers directly.
- Market processes emphasize frequent enrollment and switching activity.

PJM and Similar East Coast Markets

Many PJM-style markets operate differently.

Characteristics of these markets include:

- Customers typically already exist within the distributor's customer information system.
- Distributor billing is common.
- Retailer relationships are established without creating new customer records.
- Switching frequently occurs at billing-cycle boundaries.

Limited Deregulation Markets

Some markets support only limited customer choice. For example, some markets:

- Restrict the number of customers that may enroll with retailers.
- Allow retailer enrollment only during defined enrollment periods.
- Require customers to return temporarily to distributor service before enrolling with another retailer.

These markets often require different enrollment and customer lifecycle processing than fully deregulated markets.

Copyright © 2026, Oracle and/or its affiliates.

Market Rules Affect the Entire Customer Lifecycle

Market requirements influence every stage of the customer lifecycle. The table below outlines how the market rules affect various processes in the customer lifecycle.

Lifecycle Activity	Market Rules Influence
Enrollment	Customer creation, retailer relationships, effective dates
Billing	Billing responsibility, bill-ready or rate-ready processing
Payments	Retailer settlement and receivables processing
Customer Changes	Which participant owns customer information
Retailer Switching	Transaction timing and conflict resolution
Customer Drop	Manual drop processing and move-out behavior

Because each of these processes depends on market rules, understanding those rules should be an early implementation activity.

Configuration Reflects Market Rules

Most MTM configuration exists to represent market behavior rather than application behavior. Examples include:

- Service Provider Configuration
- Billing Relationships
- Registration Point models
- Customer Registration Process implementations
- Validation rules
- Transaction processing
- Completion processing

Configuration should therefore be driven by documented market requirements rather than assumptions based on previous implementations.

Do Not Assume One Market Represents Another

Because market rules can vary so widely, solutions developed for one market should not automatically be applied to another. For example:

- A Texas implementation may rely heavily on customer creation during enrollment, while another market assumes customers already exist.
- Billing and payment processing may differ depending on whether the distributor or retailer bills the customer.
- Registration points may identify a premise, a service point, or a customer at a premise depending on market requirements.

Although the MTM framework supports all of these approaches, each implementation must determine which model is appropriate for its market.

Implementation Guidance

Before configuring MTM:

Copyright © 2026, Oracle and/or its affiliates.

- Understand the regulatory structure of the market.
- Identify the responsibilities of distributors and retailers.
- Determine who owns customer information.
- Understand the billing model used by the market.
- Identify the registration point model.
- Review enrollment, switching, and drop requirements.
- Understand payment settlement rules.
- Identify any market-specific transaction requirements.

These activities provide the foundation for configuring MTM to support your organization's business processes.

Customer Lifecycle Example

Let's look at a simplified example of the lifecycle of a customer from the perspective of market transaction management. This example involves the following participants:

- The Distributor
- Two Retailers (referred to as Retailer A and Retailer B)
- A Customer (defined in the Distributor's Customer Information System (CIS))

This example is based on a sample market. In this market, the Distributor sends bills to customers on behalf of retailers. In this market, customers pay the distributors, which in turns pays the Retailers via "Purchase of Receivables".

The market used in this example is not based directly on any existing market. It is provided for illustrative purposes only. As noted previously, it is CRITICAL that you have a firm understanding of how your market operates before you begin planning your implementation.

Customer Enrollment

The Customer chooses to enroll with Retailer A.

Retailer A sends a Historical Usage Request transaction (814) to the Distributor.

The Distributor receives the Historical Usage Request transaction, and validates that it has all the information needed to process the request.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer A, and the Distributor sends a Usage transaction (867) to Retailer A.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer A.

Retailer A sends an Enrollment transaction (814) to the Distributor.

The Distributor receives the Enrollment transaction, and validates that it has all the information needed to enroll the customer.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer A, and the customer information system is updated as needed to reflect the customer's enrollment with Retailer A and enable the customer for billing.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer A.

Copyright © 2026, Oracle and/or its affiliates.

Billing

The customer's bill is calculated each billing period based on the customer's billing cycle (typically monthly).

When a bill is completed, the Distributor sends a Usage transaction (867) and an Invoice transaction (810) to Retailer A.

- If either or both are invalid for some reason, Retailer A sends a rejection transaction (824) to the Distributor.

Payments

The billing system sends a AP Request Adjustment to the Financial System, which is routed to the bank.

Once the payment has cleared, the Distributor sends a Payment transaction (820) to Retailer.

The Distributor receives the Payment transaction, and validates that it has all the information needed to process the payment.

- If the transaction passes validation, the Distributor sends an acceptance transaction (824) to Retailer A.
- If the transaction fails the validation, the Distributor sends a rejection transaction (824) to Retailer A.

Switching Retailers

The Customer decides to switch retailers from Retailer A to Retailer B.

Retailer B sends a Historical Usage Request transaction (814) to the Distributor.

The Distributor receives the Historical Usage Request transaction, and validates that it has all the information needed to process the request.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the Distributor sends a Usage transaction (867) to Retailer B.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.

Retailer B sends an Enrollment transaction (814) to the Distributor.

The Distributor receives the Enrollment transaction, and validates that it has all the information needed to enroll the customer.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the customer information system is updated as needed to reflect the customer's enrollment with Retailer B and enable the customer for billing.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.

The Distributor sends a Drop transaction (814) to Retailer A.

The Distributor sends a Confirmation transaction (814) to Retailer B.

Customer Information Changes

The customer's phone number changes.

Retailer B sends a Change Request transaction (814) to the Distributor.

Copyright © 2026, Oracle and/or its affiliates.

The Distributor receives the Change Request transaction, and validates that it has all the information needed to process the request.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the customer information system is updated as applicable.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.

Meter Exchange

The distributor changes the meter at the customer's premise.

Distributor sends a Change Request transaction (814) to Retailer B.

- If the transaction is invalid for some reason, Retailer B sends a rejection transaction (824) to the Distributor.

Customer Drops Retailer

The customer chooses to drop their retailer and purchase direct from the distributor.

Retailer B sends a Drop transaction (814) to the Distributor.

The Distributor receives the Drop transaction, and validates that it has all the information needed to drop the customer.

- If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the customer information system is updated as needed to enable the customer for billing by the distributor.
- If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.

The Distributor sends a Drop transaction (814) to Retailer B.

- Customer Lifecycle Example - Another Look

Customer Lifecycle Example - Another Look

This section expands on the example above, and provides specific references to the system data and objects used for each transaction.

- The Distributor is set up as a Service Provider and Market Participant in Customer Cloud Service
  - Billing Options for the Distributor are defined as "We Bill For Them, Rate Ready"
- Both Retailers are set up as Service Providers and Market Participants in Customer Cloud Service
- The Customer is defined as a Customer/Account/Premise/Service Point/Meter and a Registration Point in Customer Cloud Service

Business Process / Step	MTM USD Data and Objects
Customer Enrollment	A customer enrolls with a retailer.
Retailer A sends a Historical Usage Request transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.

Copyright © 2026, Oracle and/or its affiliates.

Business Process / Step	MTM USD Data and Objects
The Distributor receives the Historical Usage Request transaction, and validates that it has all the information needed to process the request. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer A, and the Distributor sends an Historical Usage transaction (867) to Retailer A. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer A.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Historical Usage Request Root (U2-In814HUReqRoot) parent business object. • Validation is performed based on the validation defined for the 814 Historical Usage Request **Market Message Type**, based on the In 814 Historical Usage Req Admin Root (U2-In814HUReqAdminRoot) parent business object. The Acceptance/Rejection transactions are **Outbound Market Messages**. The business object used for these is based on the Outbound 814 Historical Usage Resp Root (U2-Out814HUResponseRoot) parent business object. The Historical Usage transaction is an **Outbound Market Message**. The business object used for this is based on either the Outbound 867 Historical Usage Intrvl Root (U2-Out867HUIntervalRoot) or the Outbound 867 Historical Usage Scalar Root (U2-Out867HUScalarRoot) parent business object (depending on whether the customer is an interval or scalar customer).
Retailer A sends an Enrollment transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.
The Distributor receives the Enrollment transaction, and validates that it has all the information needed to enroll the customer. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer A, and the customer information system is updated as needed to reflect the customer's enrollment with Retailer A and enables the customer for billing. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer A.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Enrollment Request Root (U2-In814EnrolReqRoot) parent business object. • Validation is performed based on the validation defined for the 814 Enrollment Request **Market Message Type**, based on the Inbound 814 Enrollment Request Admin Root (U2-In814EnrolReqAdminRoot) parent business object. The Acceptance/Rejection transactions are **Outbound Market Messages**. The business object used for these is based on the Outbound 814 Enrollment Response Root (U2-Out814EnrlNtfcnRspRoot) parent business object. Enabling the customer billing is performed by a Market Process that creates the data needed, including creating a Service Agreement and Starting service for the customer. The business object for the Market Process would be based on the Customer Registration Process Root (U2-CRPRoot) example business object.
Billing	The customer is billed for their service.

Copyright © 2026, Oracle and/or its affiliates.

Business Process / Step	MTM USD Data and Objects
The customer's bill is calculated each billing period based on the customer's billing cycle (typically monthly).	This is done via the standard billing functionality provided with Customer Cloud Service.
When a bill is completed, the Distributor sends a Usage transaction (867) and an Invoice transaction (810) to Retailer A. • If either or both are invalid for some reason, Retailer A sends a rejection transaction (824) to the Distributor.	The Add Bill Details to GenProc on Bill Completion (U2BILLGENPRC) Bill Completion algorithm defined on the SA Type creates records in the General Processing table containing bill details. The following batch processes create **Outbound Market Messages**: • Create Outbound 867 Monthly Usage (U2CRE867): The business object for this message is based on the Outbound 867 Monthly Interval Usage Root (U2-Out867MUIntervalRoot) or Outbound 867 Monthly Usage Scalar (U2-Out867MUScalarRoot) parent business objects. • Create Outbound 810 Charges (U2CRE810): The business object for this message is based on the Outbound 810 Invoice Charges Root (U2-Out810InvChargesRoot) parent business object. The specific business objects created by these batch processes are defined in corresponding **Master Configurations**. The messages are created in the "Pending" state. A Monitor batch process moves them into the "Created" state where a series of algorithms populates the messages with appropriate data. The Rejection transaction is an **Outbound Market Message**. The business object for this is based on the Outbound 824 Application Advice Root (U2-Out824AppAdviceRoot) parent business object.
Payment	The customer pays their bill.
The billing system sends a AP Request Adjustment to the Financial System, which is routed to the bank.	This is done via the standard billing functionality provided with Customer Cloud Service.
Once the payment has cleared, the Distributor sends a Payment transaction (820) to Retailer.	The Payment transactions is an **Outbound Market Message**. The business object used for this is based on the Outbound 820 Remittance Advice Root (U2-Out820RemittanceAdviceRoot) parent business object.

Copyright © 2026, Oracle and/or its affiliates.

Business Process / Step	MTM USD Data and Objects
The Distributor receives the Payment transaction, and validates that it has all the information needed to process the payment. • If the transaction passes validation, the Distributor sends an acceptance transaction (824) to Retailer A. • If the transaction fails the validation, the Distributor sends a rejection transaction (824) to Retailer A.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object. As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 820 Remittance Pay Advice Root (U2-IN820RemittanceAdvRoot) parent business object. • Validation is performed based on the validation defined for the 820 Remittance Pay Advice **Market Message Type**, based on the Inbound 820 Remittance Advice Admin Root (U2-IN820RemittanceAdviceAdmin) parent business object. The Acceptance/Rejection transactions are **Outbound Market Messages**. The business objects used for these are based on the Outbound 824 Application Advice Root (U2-Out824AppAdviceRoot) parent business object.
Switching Retailers	The customer decides to switch retailers from Retailer A to Retailer B.
Retailer B sends a Historical Usage Request transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.
The Distributor receives the Historical Usage Request transaction, and validates that it has all the information needed to process the request. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the Distributor sends a Usage transaction (867) to Retailer B. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Historical Usage Request Root (U2-In814HUReqRoot) parent business object. • Validation is performed based on the validation defined for the 814 Historical Usage Request **Market Message Type**, based on the In 814 Historical Usage Req Admin Root (U2-In814HUReqAdminRoot) parent business object. The Acceptance/Rejection transactions are **Outbound Market Messages**. The business object used for these is based on the Outbound 814 Historical Usage Resp Root (U2-Out814HUResponseRoot) parent business object. The Historical Usage transaction is an **Outbound Market Message**. The business object used for this is based on either the Outbound 867 Historical Usage Intrvl Root (U2-Out867HUIntervalRoot) or the Outbound 867 Historical Usage Scalar Root (U2-Out867HUScalarRoot) parent business object (depending on whether the customer is an interval or scalar customer).

Copyright © 2026, Oracle and/or its affiliates.

Business Process / Step	MTM USD Data and Objects
Retailer B sends an Enrollment transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.
The Distributor receives the Enrollment transaction, and validates that it has all the information needed to enroll the customer. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the customer information system is updated as needed to reflect the customer's enrollment with Retailer B and enable the customer for billing. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Enrollment Request Root (U2-In814EnrolReqRoot) parent business object. • Validation is performed based on the validation defined for the 814 Enrollment Request **Market Message Type**, based on the Inbound 814 Enrollment Request Admin Root (U2-In814EnrolReqAdminRoot) parent business object. The Acceptance/Rejection transactions are **Outbound Market Messages**. The business object used for these is based on the Outbound 814 Enrollment Response Root (U2-Out814EnrlNtfcnRspRoot) parent business object. Enabling the customer billing is performed by a Market Process that creates the data needed, including creating a Service Agreement and Starting service for the customer. The business object for the Market Process is based on the Customer Registration Process Root (U2-CRPRoot) parent business object.
The Distributor sends a Drop transaction (814) to Retailer A.	The Drop transaction is an **Outbound Market Message**. The business object used for this is based on the Outbound 814 Drop Req Switch Root (U2-Out814DropReqSwRoot) parent business object.
The Distributor sends a Confirmation transaction (814) to Retailer B.	The Confirmation transaction is an **Outbound Market Message**. The business object used for this is based on the Outbound 814 Enrollment Response Root (U2-Out814EnrlNtfcnRspRoot) parent business object.
Customer Information Changes	The customer's phone number changes.
Retailer B sends a Change Request transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.

Copyright © 2026, Oracle and/or its affiliates.

Business Process / Step	MTM USD Data and Objects
The Distributor receives the Change Request transaction, and validates that it has all the information needed to process the request. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Change Request Root (U2-In814ChangeReqRoot) parent business object. • Validation is performed based on the validation defined for the 814 Change Request **Market Message Type**, based on the Inbound 814 Change Request Admin Root (U2-In814ChangeReqAdminRoot) parent business object. Updating the customer's phone number is performed by a Market Process that updates the data as applicable. The business object for the Market Process is based on the Change Process Root (U2-CPRoot) parent business object.
Meter Exchange	The distributor changes the meter at the customer's premise.
Distributor sends a Change Request transaction (814) to Retailer B. • If the transaction is invalid for some reason, Retailer B sends a rejection transaction (824) to the Distributor.	The Change Request transaction is an **Outbound Market Message**. The business object used for this is based on the Outbound 814 Change Request Root (U2-PLOut814CRRoot) parent business object.
Customer Drops Retailer	The customer chooses to drop their retailer and purchase direct from the distributor.
Retailer B sends a Drop transaction (814) to the Distributor.	The transaction is sent in standard EDI format. A Electronic Service Bus (ESB) or other integration component converts the message into the "MTM XML" format, based on the business object schema of the appropriate business object.
The Distributor receives the Drop transaction, and validates that it has all the information needed to drop the customer. • If the transaction passes validation, the Distributor sends an acceptance transaction (814) to Retailer B, and the customer information system is updated as needed to enable the customer for billing by the distributor. • If the transaction fails the validation, the Distributor sends a rejection transaction (814) to Retailer B.	As the message is received, an **Inbound Market Message** is created. • The business object for this message is based on the Inbound 814 Manual Drop Response Root (U2-In814DropRspRoot) parent business object. • Validation is performed based on the validation defined for the 814 Change Request **Market Message Type**, based on the Inbound 814 Drop Request Admin Root (U2-In814DropReqAdminRoot) parent business object. Updating the customer's billing setup is performed by a Market Process that updates the data as applicable. The business object for the Market Process is based on the Change Process Root (U2-CPRoot) parent business object.
The Distributor sends a Drop transaction (814) to Retailer B.	The Drop transaction is an **Outbound Market Message**. The business object used for this is based on the Outbound 814 Manual Drop Req Root (U2-Out814MDropRequestRoot) parent business object.

Copyright © 2026, Oracle and/or its affiliates.

Market Transaction Management and Customer Cloud Service

This section provides specific details concerning how Market Transaction Management (MTM) processes interact with Customer Cloud Service (CCS).

- Customer Enrollment / Registration / Move Out
- Billing and Usage
- Payments
- Change Requests

Customer Enrollment / Registration / Move Out

MTM and CCS coordinate on customer registrations/enrollments. MTM receives and processes the market enrollment transaction. The Customer Registration Process (CRP) then creates or updates the CCS customer relationship that supports the market model.

The registration process is highly market- and distributor-specific.

MTM manages the market transaction and its market-specific business rules. CCS maintains the customer data that results from the transaction, including:

- The 'V' - Person, Account, Premise, SP and master SA
- SA Relationship
- Sub-SA
- Characteristics

'We Bill For Them' Enrollments / Registrations

In a 'We Bill for Them' model, the distributor generally already maintains the customer relationship in CCS. The 'V' already exists. The CRP makes only the changes needed to support the retailer and market transaction.

For example, the CRP can locate the existing master SA, create the required SA relationship, create a retailer's sub-SA when applicable, and update characteristics. The rest of the 'V' is untouched.

'They Bill For Us' Enrollments / Registrations

In a 'They Bill for Us' model, the distributor does not own the customer. The 'V' is not set up in CCS. CCS is only responsible for sending device/usage details and charges to the retailer that is responsible for billing. CCS will have minimal information such as phone number (for outage notifications), life support / sensitive load indicator on person or premise.

Enrollment / Registration Processing

MTM does most of work in enrollment / registration processing:

1. MTM receives the inbound enrollment, move-in, or switch transaction and applies the configured market rules.
2. MTM identifies the applicable CRP and evaluates existing customer and service agreement information.
3. MTM asks for a meter read.

Copyright © 2026, Oracle and/or its affiliates.

4. MTM invokes the CRP to create the required CCS records, including Person, Account, SA, SA relationship, sub-SA, and characteristics required by the market configuration.

> **Note**
>
> The Person is always created because there is no Person Identifier to match to existing Persons.

MTM completes the 'V'. This can also include Rate Schedule update - for example, the default or the previous customer's rate.

5. MTM uses Completion Events to complete the 'V'. Completion Events provide reusable, configurable points for MTM to coordinate the resulting updates to the 'V'.

> **Note**
>
> The Customer Registration root business objects in MTM USD are intended as examples only. Implementations should create their specific business objects based on their requirements and business rules.

Stops / Move Outs

When a customer relationship ends, MTM evaluates the stop or move-out transaction and invokes the configured completion event. MTM sets the SA's End Date. There may be other information to update - e.g., switch off life support, copy characteristics (based on configuration). CCS stops the SA.

In a 'We Bill for Them' scenario, MTM also flags the bill as final for market processing.

Transfers

MTM US Distribution can support the transfer of a customer from one premise to another with the retailer. However, the solution is implementation-specific, due to varying rules.

Billing and Usage

CCS calculates and completes bills. MTM uses configured billing events and bill calculation data to create the required market usage and market charge transactions.

The billing relationship determines when CCS notifies MTM that bill data is ready for market processing. The notification is configured as an algorithm on the Service Agreement Type. When the configured billing event occurs, CCS creates a General Process that MTM uses to build the applicable market transaction.

The required billing event (trigger) depends on the billing relationship:

Billing Relationship	CCS object that triggers MTM	Purpose
We Bill for Them - Bill-Ready	Frozen Bill Segment	Sends bill-ready charge information when the bill segment is frozen.
We Bill for Them - Rate Ready	Completed Bill	Sends the completed bill after the distributor has rated the charges.
They Bill for Us	Completed Bill	Sends the completed billing information for market processing.

Copyright © 2026, Oracle and/or its affiliates.

The General Process algorithm on the Service Agreement Type is configured for the applicable trigger:

- In a 'We Bill for Them - Bill Ready' scenario, the FT Freeze algorithm starts the process.
- In a 'We Bill for Them - Rate-Ready' or 'They Bill for Us' scenario, bill completion starts the process.

CCS bill calculation lines can contain charges that must be reported to a service provider in an 810 Invoice Charge transaction. The configured Rate Schedule and its Calculation Rules are used to add the required Bill Segment Calculation Line characteristics.

The calculation line characteristic is the key link between CCS billing and the MTM market message. MTM uses it to identify which charges or bill segments must be included in the outbound transaction. Only the market-required calculation line characteristics are included.

For 'We Bill for Them - Bill Ready' processing, the applicable billable charge template is also used.

Usage

Usage transactions can require market-specific data that is not required to calculate the CCS bill. The usage transaction contains the information required by the monthly billed-usage market transaction.

For standard utility processing, the configured Cyclical Billing Usage Request business object is used where appropriate.

Payments

Market Transaction Management (MTM) works with Customer Cloud Service (CCS) to process outbound and inbound payments.

MTM applies market-specific validation and settlement rules; CCS provides billing, payment staging, adjustment, adjustment approval, and standard payment processing.

Payments are processed depending on the billing relationship:

- **We Bill For Them.** The distributor bills the customer on behalf of a retailer and settles the retailer's share. The payments are outbound.
- **They Bill For Us**. An inbound market payment is matched to CCS bills before standard payment processing distributes it.

Outbound Payments

Outbound payments use the established A/P request process in CCS to interact with the financial system, to pay the retailers.

Outbound payments use one of two timing models:

Copyright © 2026, Oracle and/or its affiliates.

Payment Timing	When payment occurs	Processing
Pay at bill time	At bill creation or when bill is due	Used for purchase-of-receivables arrangements. CCS records the settlement activity, less applicable administration and fees. When the 810 Invoice Charge transaction (inbound or outbound) is created, an Invoice Charge Payment Process is created, to manage the payment. The Invoice Charge Payment Process (ICPP) in MTM creates a payment that moves debt (i.e., amount to be paid to the retailer) from the holding SA to the billing/settlement SA when due. There is a one-to-one relationship between ICPP and Invoice Charge Payment because it's the process that triggers when to pay - i.e., on the bill due date. The retailer receives its share, while administration and delinquency fees are directed to the appropriate general ledger accounts. The Service Provider Payment Process (SPPP) in MTM: • Takes the individual adjustments on the holding SA and puts them through the required approval levels. • Creates an A/P adjustment for the settlement SA and bills it. – This involves a rate-able adjustment that performs the calculations and an offsetting adjustment for the settlement SA. – The reason for billing is to have financial transactions for arrears handling and GL processing.
Pay at payment time	When customer payment is received	Used when receivables are not purchased. The Invoice Charge Payment Process in MTM monitors payments that are applied to the SA or sub-SA and creates Invoice Charge Payments based on those payments. Similar processing via Service Provider Payment Process (mentioned above) applies.

The Invoice Charge Payment Process and Service Provider Payment Process are based on the Market Process object in CCS. MTM US Distribution provides a parent business object (BO) for Invoice Charge Payment Process. Refer to the `U2-ICPPRoot` business object in CCS, for more details. Note that there are no parent BOs for Service Provider Payment Process, as these processes are distributor-based.

Inbound Payments

Each retailer typically sends one 820 payment transaction daily.

An Inbound 820 Payment transaction is processed as follows:

1. MTM receives market payment data through the Market Payment maintenance object. The Market Payment Detail records contain the individual payments.
2. MTM performs record-level validation - for example, matching payment to a bill (810 invoice charge).
3. The Market Payment Detail record has the information to indicate whether it is a payment or an overpayment:
    a. For a payment. A payment upload staging record is created. The Bill ID is included in the staging record, for CCS payment upload (PUPL batch process) to match the payment to that particular bill.
    b. For an overpayment, the detail records include one record for the payment amount and another record for an adjustment (for the overpaid amount). MTM will create an

Copyright © 2026, Oracle and/or its affiliates.

adjustment for the overpaid amount. This is not refunded, but rather recorded as credit that a subsequent 820 would typically offset. The Overpayment and Suspense SAs on Market Provider Configuration manage overpayments.

4. PUPL runs as usual. No further matching/validation needed.

Change Requests

CCS is the system of record for the business data that changes. MTM detects, validates, and translates those changes into market transactions, or applies validated inbound market changes through the appropriate CCS processes.

Change requests follow one of these paths:

- **Outbound changes.** MTM monitors data changes in CCS. MTM compares the states before and after the change and, when applicable, creates an outbound market message.
- **Inbound changes.** MTM receives a market change request, validates it against market rules, and creates a change process that updates the appropriate CCS data.

The market configuration determines which changes are supported, which data is included, and whether or not the change is processed immediately.

Outbound 814 Change Requests

Outbound changes consist of CCS changes that must be reported to a market participant, such as a retailer. Examples include changes to service point, device, billing, rate, and/or customer data, depending on the market implementation.

The maintenance objects are configured with audit algorithms that identify the CCS data to monitor.

The Change Event Master Configuration defines the Market, Service Type, Change Type, affected Entity, Business Object, etc. The Change Type is key because it's information that the market looks for. Market-required characteristics are also configured. For example, a meter multiplier audit can identify a change to a device measurement scale. Device installations, removals, exchanges, and effective-dated meter events can also affect the values that must be reported. Some Change Types have derivation scripts that perform additional processing before the change request transaction is created.

Change Event Process & Market Message

When an audited change occurs, MTM creates a Change Event Process in Pending status and captures an initial snapshot. This gives a user an opportunity to reverse change(s) before the final snapshot is evaluated.

The Change Event Process batch then runs. When the batch process sees that there is no change (e.g., in the case where the user reversed the change), it will remove that change from processing and a transaction is not created. On the other hand, if the change is identified, the Change Event Process is linked to the Registration Point, Person, Account, Premise, Service Agreement and Service Point(s). When the change needs to go to more than one retailer (for example, inbound and incumbent), the Service Provider is left blank.

When the Change Event Process goes to the Completed state and a final snapshot is taken, the Outbound 814 Change Request market message is created in Pending status and is populated based on the settings in the Change Event Master Configuration and in the Market Message Type (for example, Market Participant Reference that indicates distributor, market recipient, and/or retailer). Depending on the change, the Outbound 814 Change Request is populated with any of the following: Meter Details, Unmetered Details, Contact Record, etc..

Copyright © 2026, Oracle and/or its affiliates.

Outbound market messages support throttling, which is used to control message volume and protect downstream integrations from a sudden surge of transactions. When throttling is in place, the Outbound 814 Change Request goes to the Message Hold state. Otherwise. it goes through to the Completed state.

When the Outbound 814 Change Request goes to the Completed state, MTM sends the outbound message to the market participant.

Inbound 814 Change Requests

Inbound changes come from a market participant (such as a retailer) or a market hub that sends a change request to the distributor.

The number of inbound changes that a retailer can do is typically small. Some common types of changes are:

- Billing relationship change - such as changing from 'We Bill For Them - Rate Ready' to 'Dual' billing relationship (or vice versa)
- Rate-related changes - such as a rate schedule or bill factor changes

Market Message & Change Event Process

MTM receives the Inbound 814 Change Request market message in the Pending state. The transaction is validated based on the validation rules configured on the related Market Message Type that defines: validation scripts, required fields, applicable To Do Type and To Do Role, Status to Validate, Next Status, Status/Reject Record, Status Reject Reason, etc.. Market Message Type also includes UI Controls that define what users can edit / view.

If validation is successful, the Inbound 814 Change Request goes to the Process Event state, where the Inbound 814 Change Event Process is created.

The Inbound 814 Change Event Process determines which CCS data is changed and how it is updated. Depending on the configuration, the change could be: immediate/automatic (contact details change), bill-based or registration process-based (for example in a billing relationship change, the needed changes to the master SA, SA Relationship and sub-SAs need to be handled by the registration process). For an immediate change, the Change Event Process takes care of updating the data. The common use case for an immediate change is contact information changes. The Market Process Type controls the data that can be changed, such as person details, rates, bill factors, or billing relationships. The change process updates the data based on the configured Change Types and implementation-specific rules.

The applicable market response is also sent. If validation fails, MTM creates an outbound reject response. Otherwise, an accept inbound market response is sent. Note: An accept response does not necessarily mean that the CCS update completed. In most cases, the data coming in from the market is valid. If, for some reason, CCS cannot apply an update, it's likely due to some internal issue (such as a missing/changed configuration) that needs to be resolved.

Implementing an MTM US Distribution Transaction

This section walks through the process involved in implementing a transaction used with MTM US Distribution.

Typical Implementation Steps

The implementation steps commonly include the following:

1. Setting up and/or verifying prerequisite general configuration

Copyright © 2026, Oracle and/or its affiliates.

2. Configuring the Market Message Type
    a. Creating the Market Message transactional Business Object
    b. Creating the Market Message Type admin Business Object
    c. Creating the Outbound Message Type
    d. Creating the Market Message Type
3. Configuring the Change Event Process
    a. Creating the Parent Process transactional Business Object and Parent Process Admin Business Object
    b. Creating the Market Process Type
    c. Configuring implementation-specific change data
4. Setting up Master Configuration
5. Setting up MO Audit Algorithms

Example: Outbound 814 Change Request

The following illustrates the above-mentioned steps using an Outbound 814 Change Request transaction.

General Configuration

Assuming that CIS Division(s), Service Types, SA Types and SA Relationship Types are already set up, configure the following:

1. External Systems, including one for MTM
2. Markets
3. Provider Configuration Options, which define billing relationships
4. Service Providers for the distributor and one or more retailers
5. Market Provider Configuration Options and Service Provider Market Options
6. SA Type SA Relationship Types. including any default sub-SA Types as applicable

> **Note**
>
> Outbound Message Types are configured as part of the outbound Market Message Type configuration

Configure the Outbound 814 Change Request

Create the Market Message Transaction and Administration Business Objects (BOs)

1. Create a parent transaction BO for the Market Message MO using a parent/lifecycle BO such as `U2-Out814ESIMaintReqRoot`.
2. Add market-specific customization or extensions, such as algorithms that populate custom details or values.
3. Create a parent admin BO for the Market Message Type MO, using a parent/lifecycle BO such as `U2-Out814ESIMaintReqAdminRoot`.

Create the Outbound Message Type

1. Add an Outbound Message Type and link it to the new transaction Business Object.

Copyright © 2026, Oracle and/or its affiliates.

2. Add the Outbound Message Type to the applicable External System.

Create the Market Message Type

1. Using the new admin Business Object, configure the Related Transaction Business Object, Service Type, Market, External System, Outbound Message Type, To Do Types and Roles, Derivation Scripts, and UI Controls.
2. Use the delivered MTM USD To Do Types or custom To Do Types. Configure derivation scripts only when overriding MTM USD behavior. UI controls can set read-only behavior, display suppression, and auditing; by default, fields are editable and auditing is off.

Configure the Change Event Process

Create the Change Event Process and Administration business objects

1. Create a Parent Process Business Object, using a parent/lifecycle BO such as `U2-ChgEvtProcAdvRoot`.
2. Add market-specific customization or extensions, such as algorithms that populate custom details or values.
3. Create a Parent Process Admin Business Object using a parent/lifecycle BO such as `U2-ChgEvtProcAdvAdminRoot`.

Create the Market Process Type

1. Configure a Market Process Type using the new admin Business Object with the related Market Process Business Object, Market, and the appropriate delivered or custom To Do Types and Roles.
2. Configure EDI-specific change types and values, including any market-specific or implementation-specific characteristic values.

Set up Master Configuration

For each Market:

1. Duplicate a Master Configuration Business Object such as U2-814ESIMasterConfig
2. Add a Master Configuration for the new Business Object, and set up the Change Configuration fields.

Configure MO Audit Algorithms

Configure the Maintenance Objects with the audit algorithms required by the implementation.

Customer Program Management

- Overview
- About Customer Program Management
- Maintaining Controllable Assets and Devices
- Maintaining Device Locations
- Maintaining Facilities
- Maintaining Metered Service
- Maintaining Metered Service Points
- Maintaining Network Locations

Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Programs and Related Records
- Maintaining Event Impact Measurement & Verification Data
- Maintaining Customer Program Service Tasks
- Enrolling and Unenrolling Customers Using Process Flow
- Creating Program Service Calls Using Process Flow
- Manually Completing Work Activities Using Process Flow
- Using the Customer Program Summary Portal
- Using the Program Dashboard
- Using Unified Search with Customer Program Management
- Using Service Order Management for Customer Program-Related Field Activities
- Other Related Functionality

Overview

Customer Program Management refers to the core functionality for utilities to create, run, and maintain demand response and distributed energy resource programs as part of a larger distributed energy resource management system (DERMS).

Customer Program Management functionality is offered via the following products:

- Oracle Utilities Custom Program Management Cloud Service
- Oracle Utilities Customer Program Management (when implemented on-premises)
- Oracle Utilities Customer Cloud Service

> **Note**
>
> Customer Program Management functionality in Oracle Utilities Customer Cloud Service must be licensed separately.

Demand response programs are programs in which customers agree to let their utility service provider temporarily alter the settings of specific devices in their homes (referred to as controllable devices or controllable assets) in response to periods of high demand. For instance, customers might enroll in a program that allows their utility to temporarily adjust their thermostat during a heat wave, or disable an electric vehicle charger during a period a of peak energy usage in the area. Instances such as these are known as "program events".

Customer Program Management also supports behavior-driven demand management programs, such as Critical Peak Pricing Programs or Peak Time Rebate programs that do not involve controllable devices.

Customer program eligibility criteria can include specific types of controllable devices, current utility rates, past and present program participation, customer opt-out history, heating/cooling technology, and specific customer questions (such as "Does your house have an internet connection?").

Customer enrollment involves identifying a specific customer that is eligible for a particular program. Some programs also involve and one or more controllable devices installed at a specific device location that are applicable to a specific program, but not all programs involve devices. Devices can include those that the customer purchases and installs themselves,

Copyright © 2026, Oracle and/or its affiliates.

those that the utility provides to the customer to install, and those that the utility provides and installs at the customer's location, and can include "smart" thermostats, electric vehicle chargers, storage batteries, and other types of devices. Programs that don't involve devices include home energy assessment programs, critical peak pricing, programs, among others.

As devices are installed and registered during the enrollment process, device information is provided to an advanced distribution management system (ADMS) such as Oracle Utilities Network Management System, a distributed energy resource management system, or a supervisory control and data acquisition (SCADA) system. These systems are sometimes referred to as "control systems" and are used to manage communication with and control of individual controllable devices during program events.

Device control scores - numerical ratings that help utilities determine if a device should participate in a program event - are calculated on a daily basis and sent to the distributed energy resource management system. When a utility determines that it should run a program event, control system operators use control score information (in addition to other data) to identify a set of devices that should participate in the program event, and the control system sends details of the event to Customer Program Management. Program event details include the date and time of the event, as well as a list of devices that will participate in the event, and if appropriate, specific dates and times during which each device will participate in the program event. This information is used to create "event participation" records for each device.

After the event has completed, the control system sends a list of devices that participated in the event, and the outcome of each device's participation. Outcomes can include successful participation in the event, cancelation of the event for the device, an "in-flight" override in which the customer overrides the control of the device at its location, or an error. The results of the program event are used to update the event participation records for each device, which are subsequently used in future control score calculations.

Calculation of device control scores is performed through configuration of program rules and score calculation rules that use pre-configured values and customer-defined formulas. Program rule parameters can include the number of times within a specific period (such as a week or month) that an individual device can participate in an event, the frequency of these cyclical periods, the minimum number of days that must transpire between events, and more. Control scores are intended to capture the contract rules the device operates under, both from an operational perspective as well as a financial perspective.

Customer measurement data can used to calculate a customer's "baseline" usage used when calculating reductions in customer usage related to demand response programs.

As device-related events are completed, program event settlement transactions are created that represent credits that will be applied to a customer's bill based on participation in an individual event. Program event settlement transactions can be created either from data received from an external analytics application or via a batch process. When created via batch processing, data regarding reductions in consumption or demand for the event is calculated via Program Settlement Calculation Rules which perform a comparison of actual consumption or demand to a "baseline" that represents an estimate of the customer's consumption or demand for the event period had they not participated in the event. When created from an external analytics application, the application sends files containing details regarding consumption or demand reduction achieved for the duration of an event for each customer. These reductions are multiplied by a price to determine the total credit to be applied to the customer's bill for the event.

Program event settlement transactions aren't applied directly to a customer's bill. Instead, they are collected in customer settlement transactions that summarize all of the events in which the customer participated during a given time period. In other words, customer settlement transactions represent the total of all of the customer's program event settlement transactions for a given time period. Customer settlement transactions are created upon requests from a

Copyright © 2026, Oracle and/or its affiliates.

customer information system. These requests are typically sent on an annual basis (such as at the beginning or end of the year), but customer settlement transactions are also created in the other specific situations such as when a customer moves out or stops their service (in which case the customer settlement is applied to the customer's final bill), when a customer is re-billed (in which case the customer settlement is applied to the customer's new bill), or when a customer unenrolls for a program (in which case the customer settlement is applied to the customer's next bill). Once calculated. customer settlement transactions can be downloaded and sent to the customer information system for use in billing.

Oracle Utilities Customer Program Management Cloud Service also includes other asset management functionality, including inventory, purchasing, financials, work orders and more.

About Customer Program Management

This section describes customer program management objects and data used with Customer Program Management.

- About Controllable Assets and Devices
- About Device Locations
- About Facilities
- About Metered Service
- About Metered Service Points
- About Network Locations
- Programs
- Event Impact Measurement and Verification
- About Customer Program Service Tasks
- About Enrollment and Unenrollment
- About Service Calls

About Controllable Assets and Devices

Controllable assets and devices are the individual devices installed at a device location that can participate in demand response programs and program events. Devices can include those that the customer purchases and installs themselves, those that the utility provides to the customer to install, and those that the utility provides and installs at the customer's location, and include devices and assets such as thermostats, batteries, photo-voltaic solar panels, and other devices.

Controllable assets are represented by two records in the system: a controllable asset, and a controllable device. When a new controllable asset is created, a corresponding controllable device record is created automatically.

Controllable assets are defined by the following:

- **Asset Type**: The asset type record on which the asset is based. Asset types define certain parameters and attributes of a class of assets.
- **Status**: The current status of the asset. Only assets in the "Active" state can participate in program events.

Copyright © 2026, Oracle and/or its affiliates.

- **Identifiers**: The Badge Number, Serial Number, Network Identifier, and Network Address used to identify the asset in the system and in external systems.
- **Specification**: The specification for the asset. See Controllable Asset Specifications below for more information.
- **Asset Disposition Detail**: The current disposition of the asset (such as "In Receipt", "In Store", "Installed", "Retired", and so on), including the asset's location and the date and time the asset was moved to that location.
- **Access Control**: Details about the organizations which have access to data related to the asset.
- **Control Score**: Details about the asset's control score, calculated via program calculation rules, including:
  - **Score**: The most recent control score calculated for the controllable asset or device.
  - **Score Date/Time**: The date and time at which the control score was calculated.
  - **Asset Availability**: A flag that specifies ("Yes" or "No") whether or not the asset is available for events based on the last score calculation.

Controllable devices are defined by the following:

- **Device Type**: The device type record on which the device is based. Device types define certain parameters and attributes of a class of devices.
- **Status**: The current status of the device. Only devices in the "Active" state can participate in program events. A device's status always matches the status of its corresponding asset.
- **Identifiers**: The Badge Number, Serial Number, Network Identifier, and Network Address used to identify the device in the system and in external systems.
- **Manufacturer** and **Model**: The manufacturer and model of the device.
- **Head-End System**: The head end system associated with the controllable device
- **Head-End Registration Status**: The registration status of the controllable device on the head end system. This is updated via data synchronization as appropriate based on data in a related Distributed Energy Resources Management System
- **DER Aggregator Service Provider**: The aggregator with which the device is registered

System Data

Controllable assets are based on the following system data:

Data Type	Description
Maintenance Object	Asset (W1-ASSET)
Business Object(s)	Controllable Asset (X1-ControllableAsset)

Controllable devices are based on the following system data:

Data Type	Description
Maintenance Object	Device (D1-DEVICE)
Business Object(s)	Item (D1-Item)

Controllable Asset Specifications

Controllable asset specifications describe design details and functional performance criteria for a set of controllable assets. This can include manufacturer, model, and other information

Copyright © 2026, Oracle and/or its affiliates.

specified for the assets. Specifications may also contain written procedures, careful handling instructions for hazardous or delicate materials, or any other set of attributes needed to provide information about a set of assets. Controllable asset specifications should be created using the Controllable Asset Specification (X1-ControllableAssetSpec) business object.

See About Specifications for more information about specifications.

System Data

Controllable asset specifications are based on the following system data:

Data Type	Description
Maintenance Object	Specification (W1-SPEC)
Business Object(s)	Controllable Asset Specification (X1-ControllableAssetSpec)

Distributed Energy Resource (DER) Device Attributes

Registration of DER devices includes populating various attributes of the device and related specifications. These DER attributes are captured as characteristics whose Characteristic Types are defined for the "Asset Attribute" and "Specification Attribute" characteristics entities.

See Distributed Energy Resource (DER) Device Attributes in for more information about configuring these attributes and attribute types.

About Device Locations

Device locations represent locations where controllable devices are installed and used. A device location can have multiple controllable devices installed at a time.

Device locations are often created as a result of the customer enrollment process, but can be created on an ad-hoc basis as needed.

Device locations are defined by the following:

- **Device Location Type**: The device location type record on which the device location is based. Device location types define certain parameters and attributes of a class of device locations.
- **Distribution Network Facility**: The network node closest to the device location. In terms of electric networks, facilities typically represent transformers.
- **Division**: The operating company or operating division within a large conglomerate of utilities associated with the device location.
- **Status**: The current status of the device location.
- **Address and Location Information**: The address (including city, state, postal code, and country) and geographic location of the device location.
- **Controllable Devices**: A list of one ore more controllable devices installed at the device location, including the date and time the device was installed or removed (as applicable), and whether or not the device is active. When a controllable device has been removed, this list also indicates a crew (used to identity the Out of Service Location where the the asset is moved to upon removal).
- **Life Support / Sensitive Load Information**: Information regarding life support equipment installed at the device location, or if the load supplied to the device location is considered sensitive.

Copyright © 2026, Oracle and/or its affiliates.

- **Field Information**: Information related to field work performed at the device location, including any applicable warnings or instructions.

System Data

Device locations are based on the following system data:

Data Type	Description
Maintenance Object	Service Point - MDM (D1-SP)
Business Object(s)	Device Location (X1-DeviceLocation)

About Facilities

Facilities represent the network node level closest to the device location. In terms of electric networks, facilities typically represent transformers, but can also be feeders and substations.

Facilities are defined by the following:

- **Facility Type**: The type of facility (derived from the business object). Facilities used with Customer Program Management are typically transformers, but can also be feeders and substations.
- **Service Type**: The type of service provided by the facility. This is most often "Electric".
- **Status**: The current status of the facility.
- **External ID**: An ID used to identify the facility in external systems.
- **Time Zone**: The time zone in which the facility resides.

System Data

Facilities are based on the following system data:

Data Type	Description
Maintenance Object	Facility (D1-FACILITY)
Business Object(s)	Feeder (D1-Feeder) Substation (D1-Substation) Transformer (D1-Transformer)

Defining Facility Types

Different types of facilities can be defined using the Facility Type (D1-FacilityTypeLookup) extendable lookup. This lookup can be used to define the Network Node Type for each type of facility (such as transformer, feeder, or substation), as well as whether or not facilities of a given type are eligible to be connected to metered service points (via the **SP Connection Eligibility** flag).

About Metered Service

Metered service records represent a customer's service agreements and usage subscriptions in Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter),

Metered service records are created via a data synchronization process between Customer Program Management Cloud Service and Customer Cloud Service.

Copyright © 2026, Oracle and/or its affiliates.

Metered Service records are used as Master Subscriptions when creating Program Subscriptions.

Metered service records are defined by the following:

- **Metered Service Type**: The metered service type record on which the metered service is based. Metered service types define certain parameters and attributes of a class of metered services.
- **Status**: The current status of the metered service record.
- **Start and End Date/Time**: The date and time on which the metered service record started, and ended (if applicable).
- **External ID**: An ID used to identify the metered service record in an external system.
- **External Account ID**: An ID used to identify the account associated with the metered service record in an external system.
- **Main Customer**: The program contact associated with the metered service record.
- **Time Zone**: The time zone associated with the metered service record.
- **Bill Cycle**: The bill cycle associated with the metered service record, used when calculating program incentives and fees via batch processing. If defined here, the Bill Cycle is defined by default on all program subscriptions for the metered service.
- **Rate History**: A list of one or more effective-dated Oracle Utilities Customer Care and Billing or Oracle Utilities Customer to Meter (or Customer Cloud Service) rates for the metered service record. These are used to determine a customer's eligibility to participate in a program. Rates can be defined using the Customer Rate Schedule extendable lookup.
- **Metered Service Points**: A list of one or more metered service points associated with the metered service record, including the start and stop dates and times for each.
- **Program Subscriptions**: One or more program subscriptions that have the metered service defined as their Settlement Group.
- **Attribute Values**: Attribute values whose characteristic type are defined as valid characteristic types on the metered service type.
- **Settlement Group Program Subscriptions**: Program subscriptions that have the metered service defined as their Settlement Group.

System Data

Metered service records are based on the following system data:

Data Type	Description
Maintenance Object	Usage Subscription (D1-US)
Business Object(s)	Metered Service (X1-MeteredService)

About Metered Service Points

Metered service points represent service points in Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter).

Metered service points are created via a data synchronization process between Customer Program Management Cloud Service and Customer Cloud Service.

Metered service points are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Metered Service Point Type**: The metered service point type record on which the metered service point is based. Service points types define certain parameters and attributes of a class of metered service points.
- **Division**: The operating company or operating division within a large conglomerate of utilities associated with the metered service point.
- **Weather Station**: The weather station associated with the metered service point. Weather stations are mapped to metered service points based on geographical location. When a metered service point is added or updated, an algorithm maps the metered service point to the nearest weather station using the **Location Information** (specifically the geographic latitude and longitude). If longitude and latitude are not available, mapping is done based on the metered service point's postal code and assigns the weather station defined for the Postal Code Default that includes the service point's postal code.
- **Status**: The current status of the metered service point.
- **External Service Point ID**: An ID used to identify the metered service point in an external system.
- **External Premise ID**: An ID used to identify the premise associated with the metered service point in an external system.
- **Main Contact**: The program contact associated with the metered service point.
- **Address and Location Information**: The address (including city, state, postal code, and country) and geographic location of the metered service point.
- **Field Information**: Information related to field work performed at the metered service point, including any applicable warnings or instructions
- **Distribution Network Facility**: The network node closest to the metered service point. In terms of electric networks, facilities typically represent transformers.
- **Access Group**: The access group used to control which groups of users have access to the metered service point (if applicable). See Defining Access Groups in the *Administrative User Guide* for more information.

System Data

Metered service points are based on the following system data:

Data Type	Description
Maintenance Object	Service Point - MDM (D1-SP)
Business Object(s)	Metered Service Point (X1-MeteredServicePoint)

About Network Locations

Network locations define the location of a facility within a larger network.

Network locations hold the network nodes that provide service to a facility. In the case of electric networks, network locations define the feeder and substation associated with a transformer (facility). Because these network nodes can change over time, a facility can have many network locations (this is an effective-dated relationship). For example, for electric service, whenever the substation or feeder that provides power to a given transformer is changed, a new record that references the latest substation and feeder is typically created.

Network Locations are defined by the following:

- **Facility ID**: The facility associated with the network location.

Copyright © 2026, Oracle and/or its affiliates.

- **Start** and **End Date/Time**: The date and time range during which the facility was connected to the feeder and/or substation.
- **Feeder**: An ID for the feeder related to the facility.
- **Substation**: An ID for the substation related to the facility.
- **External ID**: An ID used to identify the network location in external systems.

System Data

Network locations are based on the following system data:

Data Type	Description
Maintenance Object	Network Location (D1-NWLOC)
Business Object(s)	Electricity Network Location (D1-ElectricityNetworkLocation) Nominal Electricity Network Location (D1-NominalElecNetworkLocation)

Programs

This section describes components and data related to demand response programs in Customer Program Management.

Information about programs is maintained in Customer Program Management, but other systems are responsible for initiating and creating program events.

- About Programs
- About Program Contacts
- About Contact Communications
- About Program Subscriptions
- About Program Control Scores
- About Program Events
- About Event Participation
- About Program Event Settlement Transactions
- About Program Incentives and Fees
- About Customer Settlement Transactions
- About Interval Prices and Price Sets

About Programs

Programs represent specific demand response programs in which customers agree to let their utility service provider temporarily take control of specific devices in their homes (referred to as controllable devices or controllable assets) in response to periods of high demand. For instance, customers might enroll in a program that allows their utility to temporarily adjust their thermostat during a heat wave, or disable an electric vehicle charger during a period a of peak energy usage in the area. Instances such as these are known as "program events".

Programs are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Program Type**: The program type associated with the program. Program types define certain parameters and attributes of a class of programs.
- **Status**: The current status of the program. Only programs in the "Active" state are eligible for customer enrollment and events.
- **Eligibility**: A program initiative associated with the program, used to determine eligibility for customers to enroll in the program.
- **Approval Profile**: The process used for approval of the program (see Approvals and Approval Profiles below for more information).
- **Suspend**: A flag that indicates if the program is currently suspended (the default is No).
- **Start Date**: The date on which the program is active and available.
- **End Date**: The date after which the program is no longer active.
- **Division**: The operating company or operating division within a large conglomerate of utilities associated with the program.
- **Program External ID**: The ID for the program used by external systems.
- **Allow Customer Unavailability**:: A flag that indicates (Yes or No) if customers are allowed to make themselves unavailable for a period of time for program events for the program. Customer unavailability for a period of time is captured using Customer Unavailability device events.
- **Program Quantities**: One or more quantities associated with the program such as the default reduction capacity for demand response programs.
  - **Start Date/Time**: The start date and time for the time period during which the quantity applies
  - **End Date/Time**: The end date and time for the time period during which the quantity applies
  - **Program Quantity Type**: The type of quantity (based on the **Program Quantity Types** defined on the program type)
  - **Program Quantity**: The quantity, based on the UOM, TOU, and SQI defined for the program quantity type

In addition, programs can also have the following related objects and data:

- **Specifications**: One or more specifications for devices eligible for the program, used to determine eligibility for customers to enroll in the program (the customer must have devices based on the listed specifications to join the program).
  - Programs based on a program type whose **Program Category** is "DER Registration" must have a specification and attributes defined (see Maintaining Specifications for more information). When registering and enrolling a DER device, the device's specification attributes must match the specification's attributes in order for the DER device to be registered and enrolled.
- **Rates**: One or more rates used to determine eligibility for customers to enroll in the program (the customer must be on one of the listed rates to join the program)
- **Rules**: One or more program rules associated with the program. If a program has more than one program rule, the rule with the latest effective date/time is used. See Understanding Program Rules in the *Administrative User Guide* for more information about program rules.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> : If a Device Score rule is added to program, or an existing Device Score Program Rule associated with a program is edited, a synchronization request and outbound message are created to update external DERMS systems such an Oracle Utilities Grid Edge DERMS.

- **Qualifications**: One or more qualification questions used to determine eligibility for customers to enroll in the program. The **Response Required** flag can be used to indiciate that an answer must be provided for the question. See Understanding Customer Questions in the *Administrative User Guide* for more information about questions.
- **Informational**: One or more informational questions used to capture additional information about customers when enrolling in the program. The **Response Required** flag can be used to indiciate that an answer must be provided for the question. See Understanding Customer Questions in the *Administrative User Guide* for more information about questions.
- **Events**: A list of program events associated with the program. See About Program Events for more information about program events.
- **Program Settlement Calculation Groups** (on the **Calculation Groups** tab): One or more program settlement calculation groups that contain settlement calculation rules used to calculate program settlements for devices associated with the program. When more than one group is defined on the program, the appropriate calculation group is executed based on the **Event Settlement Method**. If more than one group is defined for the same Event Settlement Method, a program settlement transaction is created for each group. If a **Calculation Group Determination** is defined on the program's type, that algorithm may override which calculation group is used. Program Settlement Calculation Groups are defined by the following:
  - **Eent Settlement Method**: A flag that indicates when program settlement events are calculated after participating in an event. Options include:
    - **After Event Participation**: Calculations are performed as each program settlement event is created. This option is used with settlement calculations based on non-peak or consumption measurements (referred to as "kWh Avoided").
    - **End of Season**: Calculations are performed at the end of a specified time period (a year, a season, and so on). This option is used with settlement calculations based on peak or demand measurements ("kW Drop").
  - **Transaction Type**: The type of transaction . Options for this are defined as Characteristic Values for the Settlement Transaction Type (X1-STTRT) Characteristic Type.
  - **Calculation Group**: The program settlement calculation group that contains settlement calculation rules used to calculate program settlements for the selected **Event Settlement Method** and **Transaction Type**.
  - **Effective Date/Time**: The date and time at which the calculation group is in effect.
  - **Expiration Date/Time**: The date and time at which the calculation group expires and is no longer in effect.

System Data

Programs are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Dynamic Option (D1-DOP)
Business Object(s)	Program (X1-Program) Program Characteristics (X1-ProgramCharacteristics)

Calculate Control Score via Program Transaction Algorithm

The Calculate Control Score via Program Transaction (X1-CLCNTPRTR) algorithm, defined on the "Program (BO) - Control Score Calculation" System Event creates program transactions that are used to calculate control scores for controllable devices associated with a program. This algorithm also defines the following variables that can be used in Program Calculation Rules:

- **Control Score Variable Name**: Defines the variable used to represent control scores. When Program Calculation Rules return a value for this variable (defined as the Result Variable on the rule), the score is saved to the controllable asset (along with the date and time at which the score was calculated). The default value for this variable is "SC".
- **Availability Mark**: Defines the variable used to represent the availability of the device's participation in program events. If Program Calculation Rules return a value of 0 for this variable, the Asset Availability flag in the controllable asset is set to "No". If Program Calculation Rules return a non-zero value for this variable or the rule doesn't return a value for this variable, the Asset Availability flag in the controllable asset is set to "Yes". The default value for this variable is "AV".

Approvals and Approval Profiles

Programs and program types can specify an Approval Profile used to define a process for approval of programs. Programs with an Approval Profile enter the "Approval in Progress" state after being created, and must be approved before moving into the "Active" state. An Enter algorithm on the "Approval in Progress" state creates an approval request based on the specified approval profile.

See About Approvals for more information about creating and maintaining approvals and approval profiles.

Program Incentives and Fees

Beyond incentives based on reductions in consumption (kWh Avoided) demand (kW Drop) via program settlements, programs can be configured to include other customer incentives, such as sign-up incentives for enrollments, referral rebates, and incentives for remaining enrolled during the peak season. Customers may also receive penalties or fees in some circumstance, for instance if they terminate their enrollments before the contracted period, or when the utility incurs charges related to device repairs, device upgrades, premium device fees. The incentive and fees are sent to the customer information system to be passed on to the customer.

Program incentives and fees are configured as Program Rules that define how and when calculations are performed, with corresponding Program Settlement Calculation Rules that define the specific calculations performed for each type of incentive or fee.

Customer Unavailability Events

Customer unavailability events define periods of time during which customers make themselves (and controllable devices at the customer's device location) as unavailable for program events for a particular program. For example, if a customer will be out of town for a period of time, they may choose to make themselves unavailable to participate in program

Copyright © 2026, Oracle and/or its affiliates.

events during that time. Customer unavailability periods for the same device cannot overlap. Customer unavailability events are based on device events.

Customer unavailability events are defined by the following:

- **Customer Unavailability Type**: The customer unavailability device event type. See Understanding Device Event Types for more information about creating device event types.
- **Status**: The current status of the event. Possible statuses include "Active", "Inactive", and "Cancel".
- **Device**: The controllable device marked as unavailable.
- **Device Location**: The device location for the controllable device marked as unavailable.
- **Unavailability Start Date/Time**: The start date and time of the unavailability period for the customer and controllable device
- **Unavailability End Date/Time**: The end date and time of the unavailability period for the customer and controllable device
- **Reason**: The reason the customer is unavailable, from the Device Event Reason (D1-DeviceEventReasonLookup) extendable lookup

System Data

Customer Unavailability events are based on the following system data:

Data Type	Description
Maintenance Object	Device Event (D1-DVCEVENT)
Business Object(s)	Customer Unavailability (X1-CustomerUnavailability)

About Program Contacts

Program contacts represent customers enrolled in programs, and are associated with programs via program subscriptions.

Program contacts are defined by the following:

- **Contact Type**: The program contact type associated with the contact . Contact types define certain parameters and attributes of a class of contacts.
- **Name**: The contact's name.
- **Contact Information**: Contact information for the contact, including phone number and/or email address (used with customer contacts),
- **Contact External ID**: An ID used to identify the contact in external systems (used with customer contacts).
- **Life Support / Sensitive Load Information**: Indicates if the contact is considered to be on life support (used with customer contacts).
- **Asset System Contact ID**: An ID used to identify the contact in external asset systems (used with customer contacts).
- **Address**: The address for a point of contact.
- **Phones**: One or more phone numbers for a point of contact.
- **Emails**: One or more email addresses for a point of contact.

Copyright © 2026, Oracle and/or its affiliates.

System Data

Program contacts are based on the following system data:

Data Type	Description
Maintenance Object	Contact (D1-CONTACT)
Business Object(s)	Business (D1-Business) Person (D1-Person) Point of Contact (D1-PointOfContact)

About Contact Communications

Contact communications are interactions with the customer, most often notes captured during an enrollment or unenrollment process.

Contact communications are defined by the following:

- **Create Date/Time**: The date and time the communication was created
- **Contact ID**: The contact associated with the communication
- **Communication Channel**: The channel used for the communication, such as email or phone. Values for this are defined in the Customer Communication Channel (X1-CustCommChnLookup) extendable lookup.
- **Contact Communication Class**: The class of the communication, such as enrollment, unenrollment, trouble report, and so on. Each class can have one more associated communication categories. Values for this are defined in the Customer Communication Class (X1-CustCommClassLookup) extendable lookup.
- **Contact Communication Category**: The category of the communication, such as written or verbal. Values for this are defined in the Customer Communication Category (X1-CustCommCatLookup) extendable lookup.
- **Metered Service**: The metered service associated with the communication (from the enrollment or unenrollment service task)
- **Metered Service Point ID**: The metered service point associated with the communication (from the enrollment or unenrollment service task)
- **Program ID**: The program associated with the communication (from the enrollment or unenrollment service task)
- **Service Task ID**: The service task associated with the communication (from the enrollment or unenrollment service task)
- **WO Activity**: The work order activity associated with the communication (from the enrollment or unenrollment service task)
- **Field Activity ID**: The field activity associated with the communication (from the enrollment or unenrollment service task)
- **Notes**: Notes associated with the communication (from the enrollment or unenrollment service task)

System Data

Contact communications are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Contact Communication (D1-CONTCOMM)
Business Object(s)	Customer Program Communications (X1-CustomerComm)

About Program Subscriptions

Program subscriptions represent a customer's enrollment in a program, and link a customer with a program and (in most cases) a device location. Customers can have separate program subscriptions for different device locations and programs.

Metered service records serve as a parent to individual program subscriptions for each of a customer's programs and/or device location. If multiple program subscriptions are associated to the same device location, one can be defined as a "parent" program subscription. Program subscriptions are most often created as a result of the customer enrollment process, but can be created on an ad-hoc basis if needed.

Program subscriptions are defined by the following:

- **Program Subscription Type**: The program subscription type associated with the program. Program subscription types define certain parameters and attributes of a class of program subscriptions.
- **Metered Service**: The metered service record that serves as a "parent" for the program subscription.
- **Parent Program Subscription**: (Optional) A program subscription that serves as a "parent" for the program subscription.
- **Settlement Group**: A metered service for the Main Customer associated with the program subscription, used to group program subscriptions for customer settlement transactions.
- **External Account ID**: An ID used to identify the program subscription in external systems.
- **Main Customer**: The program contact for the customer associated with the program subscription.
- **Status**: The current status of the program subscription.
- **Start** and **End Date/Time**: The date and time on which the program subscription record started, and ended (if applicable).
- **Time Zone**: The time zone associated with the program subscription.
- **Device Location**: The device location associated with the program subscription. This defines the devices for which the customer is enrolled in the program.
- **Program**: The program in which the customer is enrolled.
- **Program Enrollment Source**: The source of the customer's enrollment in the program. Options incude:
  - **Bring Your Own**: The enrollment was created as a result of the customer installing their own controllable device.
  - **Contact Center**: The enrollment was created by the customer contacting the utility's contact center. This includes enrollments created via the Customer Enrollment Process Flow.
  - **Self Service**: The enrollment was created via an online customer self service portal.

Copyright © 2026, Oracle and/or its affiliates.

- **Bill Cycle**: The bill cycle associated with the program subscription, used when calculating program incentives and fees via batch processing. If defined for the parent metered service, the Bill Cycle is defined by default on all program subscriptions for the metered service.
- **Transaction Calculation Groups**: One or more effective dated program calculation groups that contain program calculation rules used to calculate scores for devices associated with the program subscription. If more than one group is defined on the program subscription, the group with the latest effective date/time is used.
- **Fallback Transaction Calculation Groups**: A list of one or more effective dated program calculation groups used if a program calculation group is not defined on the program subscription (from the program subscription type).

System Data

Program subscriptions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Subscription (D1-US)
Business Object(s)	Program Subscription (X1-ProgramSubscription)

Creating Program Event Settlement Transactions

Program event settlement transactions can be created via the "Initiate program event settlement for completed program events" (X1-INIPES) monitor algorithm defined on the program subscription business object. This algorithm is triggered by the Program Subscription BO Periodic Monitor Process (X1-PSPMB) batch process (see below).

This algorithm scans program events and event participation records related to the program subscription looking for program events that have been in the "Complete" status for a number days (defined as a parameter on the algorithm) with related event participation records that are in either the "Successful participation" or "In-flight override" states. The algorithm creates program event settlement transactions for each program event/event participation found.

Program event settlement transactions are created in the "Pending" state, and are processed by the Process Program Settlement (X1-PREVS) batch process which transitions the transaction to a specific state depending on the type of program event settlement transaction. See About Program Event Settlement Transactions for more information.

Batch Processing

Program subscription processing is supported by the following monitor batch processes:

- Program Subscription BO Periodic Monitor Process (X1-PSPMB): Used to trigger monitor algorithms on the program subscription.

Use the **Batch Control** portal to learn more about the parameters used by these batch processes.

About Program Control Scores

Program control scores are created via a regularly scheduled process that performs control score calculations for each device associated with a program subscription, based on the transaction calculation groups defined for the program subscription. A separate program control scores is created for each device associated with the program subscription.

Copyright © 2026, Oracle and/or its affiliates.

Program control scores are created by the Calculate Asset Score (X1-CALAS) batch process , which should be run on a daily basis to monitor controllable assets that are currently installed on a device location and enrolled in a device control program. This batch process invokes the Calculate Control Score (X1-CLCNTSCR) algorithm (a Monitor algorithm on the "Installed" state of the Controllable Asset (X1-ControllableAsset) business object). This in turn invokes the Calculate Control Score via Program Transaction (X1-CLCNTPRTR) algorithm defined on the "Program (BO) - Control Score Calculation" System Event on the Program, which creates the program control score. When the program control score enters the "Calculate" state, the Identify Program Calculation Group and Apply its Rules (X1-CALCPT) algorithm executes the Program Calculation Group defined on the Program Subscription (if the Program Subscription has multiple calculation groups, the group with the latest effective date/time is used).

Program transactions contain the following data:

- **Program Subscription**: The program subscription for which the program control score was run.
- **Calculation Group**: The transaction calculation group containing the transaction calculation rules executed as part of the program control score.
- **Transaction Date/Time**: The date and time when the program transaction was run.
- **Device**: The device for which the program control score was run.
- **Status**: The status of the program control score. Statuses include Pending, Error, Discarded, and Complete.
- **Calculation Results**: The results of the calculations performed by the transaction calculation rules. This includes the program control score's start and end dates and times, the Program Calculation Rules and variables used to calculate the results, and the individual quantities and formulas for each variable used in the calculation.
- **Issues**: One or more messages related to the program control score.

System Data

Program control scores are based on the following system data:

Data Type	Description
Maintenance Object	Usage Transaction (D1-USAGETRAN)
Business Object(s)	Program Control Score (X1-ProgramTransaction)

About Program Events

Program events represent instances when a utility takes control of a set of controllable assets in response to demand. For example, during a heat wave, a utility may wish to adjust the thermostats (controllable assets) during peak usage times to help alleviate demand.

Program events are performed by an advanced distribution management system (ADMS) or a supervisory control and data acquisition (SCADA) system. When an event is to occur, the ADMS or SCADA system information about the program (including date/time information), and a "Scheduled" event record is created. The same system then creates event participation records for each device to be included in the event. Once the event has been processed, the external system updates the event to the "Complete" status, and then sends list of impacted devices and the status of each. These details are captured in program event participation records.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Program Events and Program Event Stages**
> Program Events in Customer Program Management *program event stages* of program events defined and managed in external systems such as ADMS or SCADA. A program event in ADMS or SCADA can comprise multiple program event stages. For example, a program event in an ADMA or SCADA might be 8 hours in total duration, comprising 4 program event stages of 2 hours each. Each of these 2-hour program event stages would be represented in CPMS/CPM as a program event. Program events in external systems are designated by the **External Event ID** field on each program event.

Program events are defined by the following:

- **Program**: The program associated with the event.
- **Status**: The current status of the event (Pending, Complete, or Canceled).
- **Program Category**: The category of the event, such as Device Test, Economic, Reliability, and so on.
- **Start** and **End Date/Time**: The date and time when the event started and ended.
- **External ID**: An ID used to identify the program event stage in an external DERM system.
- **External Event ID**: An ID used to identify the program event in an external DERM system.

System Data

Program events are based on the following system data:

Data Type	Description
Maintenance Object	Dynamic Option Event (D1-DOPEVT)
Business Object(s)	Program Event (X1-ProgramEvent)

About Event Participation

Event participation records capture the participation of a controllable device in a program event. When a program event is to occur, the ADMS or SCADA system sends a list of devices and date/time information to "reserve" the devices for the event, and event participation records are created. Once the event is complete, the external system sends a list of impacted devices and the status of each. These details are captured in event participation records.

Event participations are defined by the following:

- **Status**: The current status of the event participation record.
- **Participation Start** and **End Date/Time**: The start and end date and time of the event.
- **Reservation Start** and **End Date/Time**: The start and end date and time of the device's participation in the event, Some devices may participate in only a portion of an event. If not applicable, this field is blank.
- **Device ID**: The information string for the controllable device which participated in the event.
- **Program Subscription**: The program subscription associated with the device and program associated with the event.

Copyright © 2026, Oracle and/or its affiliates.

- **Program**: The program for which the event took place.
- **Program Event**: The event associated with the event participation record.
- **Reservation**: A flag that indicates if the event participation record represents a device reservation. This flag is typically set to "Yes" when the system receives a list of devices for a planned event.
- **Outcome**: The outcome of the device's participation in the event. Possibly outcomes include "Successful participation", "Non-responsive device", "In-flight override", "Error", and "Canceled".
- **Preemptive Opt-Out** : A flag which indicates if the device preemptively opted out of the event. This flag is most often set to "Yes" when a device is removed from an event after an event participation record has been created.
- **External ID**: An ID used to identify the event participation record in external systems.

System Data

Event participation records are based on the following system data:

Data Type	Description
Maintenance Object	Event Participation (D1-EVENTPART)
Business Object(s)	Event Participation (X1-EventParticipation)

Successful Event Participations

The number of successful event participations for a device is used when calculating control scores for the device (via Program Transactions). A device's event participation is considered successful in the following circumstances:

- The Event Participation has an **Outcome** of "Successful Participation".
- The Event Participation has an **Outcome** of "Canceled". This happens when a device has been reserved but the event stopped halfway through due to reasons such as, target load control reached. It is considered as a success in customer perspective, since it is the utility that stopped the event which resulted to a cancelled outcome.
- The Event Participation has a blank **Outcome** and the device was not preemptively opted out of the event (**Preemptive Opt-Out** is "No" or blank). Posting of an event's outcome may be delayed, and until then the event participation of a reserved device is considered as a success as long as the customer did not opt out before the event transpires.

About Program Event Settlement Transactions

Program event settlement transactions represent credits that will be applied to a customer's bill based on participation in an individual event. Note that program event settlement transactions aren't applied directly to a customer's bill. A customer's program event settlement transactions are totalized into Customer Settlement Transactions (see About Customer Settlements Transactions) which are applied to the customer's bill.

Program event settlement transactions are calculated in different ways depending on the type of event and the program on which they're based. Typically programs are based on either consumption saved or demand reduction events.

- Program event settlement transactions for consumption-based program events can be calculated as they created. The consumption saved per hour (represented as hourly interval values, typically in kWh) are multiplied by hourly prices for the same date and time

Copyright © 2026, Oracle and/or its affiliates.

and the results are totaled to create an settlement amount that will be applied to the customer's bill. Program settlement transactions for consumption-based programs are referred to "kWh Avoided" program settlement transactions.

- Program event settlement transactions for demand-based program events can be calculated as they created or are calculated based on a calculated demand value for a specified period of time, such as a year, a season, and so on (these are referred to as "end of season"). For these demand-based events, program event settlement transactions capture individual demand reduction interval value as well as a the "Maximum Drop" for an event (the largest demand reduction value for the event) and are saved in a "Calculation Deferred" state until a Customer Settlement Transaction is created for the period. Program event settlement transactions for demand-based programs are referred to "kW Drop" program settlement transactions.
- When a Customer Settlement transaction is created for "end of season" demand-based programs, a new program event settlement transaction is created for the entire time period which calculates the average of a set of the "Maximum Drop" values from the "Calculation Deferred" transactions created previously. These program settlement transactions are referred to as "Combined kW Drop" program settlement transactions.

The **Program Settlement Method** attribute on the Program indicates when settlement is calculated for participating in an event. Options include:

- **After Event Participation**: Calculations are performed as each program event settlement is created. This option is most often used with settlement calculations based on non-peak or consumption measurements ("kWh Avoided"), but can be used with settlement calculations based on peak or demand measurements ("kW Drop").
- **End of Season**: Calculations are performed at the end of a specified time period (a year, a season, and so on). This option is used with settlement calculations based on peak or demand measurements ("kW Drop").

Program event settlement transactions can be created either from data received from an external analytics application or via a monitor batch process run for program subscriptions (see Creating Program Event Settlement Transactions in About Program Subscriptions).

- When created from data received from an external analytics application, the external analtyics applicationsends files containing details regarding consumption or demand reduction achieved for the duration of an event for each customer. A batch process imports these files and creates business flags that capture the data for each metered service point. These business flags in turn create program event settlement transactions containing data regarding reductions in consumption or demand for the event.
- When created via batch processing, data regarding reductions in consumption or demand for the event is calculated via Program Settlement Calculation Rules which perform a comparison of actual consumption or demand to a "baseline" that represents an estimate of the customer's consumption or demand for the event period had they not participated in the event. See **Baseline Calculations** for more information about this calculation.

There are three types of program settlement transactions:

- "kWh avoided" program event settlement transactions are created based on reductions in consumptions achieved for the duration of an event for each customer. See Processing KWH Avoided Files for more information about how "kWh Avoided" program event settlement transactions are created from business flags received from an external analytics application.
- "kW Drop" program event settlement transactions are created based on reductions in demand achieved for the duration of an event for each customer. See Processing KW Drop Files for more information about how "kW Drop" program event settlement transactions are created from business flags received from an external analytics application.

Copyright © 2026, Oracle and/or its affiliates.

- "Combined kW Drop" program event settlement transactions are created by customer settlement transactions and contain demand reduction values derived from individual "kW Drop" program event settlement transactions in the "Calculation Deferred" state that fall within the date range of the customer settlement transaction.

Program event settlement transactions are created in the "Pending" state, and are processed by the Process Program Settlement (X1-PREVS) batch process which transitions the transaction to a specific state depending on the type of program settlement transaction:

- Pending "kWh Avoided" program event settlement transactions (for programs with a **Program Settlement Method** of "After Event Participation") are transitioned to the "Calculate" state where Consumption-based Program Settlement Calculation Rules defined for the program related to the program settlement transactions (See Understanding Program Settlement Calculation Rules for more information) are executed. The amount of a "kWh Avoided" program settlement transaction is calculated by applying the Program Settlement Calculation Rules defined on the metered service's program to the "kWh avoided" values from the program settlement transaction. These rules apply interval prices to the "kWh avoided" interval values and total the results for the event. Once the program settlement transaction calculations are complete, it is transitioned to the "Calculated" state.
- Pending "kW Drop" program event settlement transactions (for programs with a **Program Settlement Method** of "End of Season") are transitioned to the "Calculation Deferred" state. These are later used by customer settlement transactions and "Combined kW Drop" program settlement transactions.
- Pending "Combined kW Drop" program event settlement transactions are transitioned to the "Calculate" state where demand-based Program Settlement Calculation Rules defined for the program related to the program event settlement transactions (See Understanding Program Settlement Calculation Rules for more information) are executed. The amount of a "Combined kW Drop" program event settlement transaction is calculated by applying the Program Settlement Calculation Rules defined on the metered service's program to the "kW Drop" values derived from the program event settlement transactions in the "Calculation Deferred" state to calculate the overall reduction in demand and apply a price to calculate the settlement amount for the event. Once the program event settlement transaction calculations are complete, it is transitioned to the "Calculated" state.

Program event settlement transactions contain the following data:

- **Program Subscription**: The program subscription associated with the event for which the program settlement transaction was created
- **Start Date/Time**: The start date and time of the event
- **End Date/Time**: The end date and time of the event
- **Parent Customer Settlement Transaction**: The customer settlement transaction that is considered the "parent" of the program event settlement transaction. (Note: Only populated after after customer settlement transaction request is received).
- **Used on Bill**: A flag that indicates (Yes or No) if the program event settlement transaction has been included in a customer settlement transaction.
- **Status**: The current status of the program event settlement transaction.
- **Calculation Details**:
  - **Program Event**: The program event for which the program event settlement transaction was created.
  - **Price**: The price used with "kW Drop" program event settlement transactions

Copyright © 2026, Oracle and/or its affiliates.

- **Program Settlement Calculation Group**: The program settlement calculation group containing the program settlement calculation rules executed as part of the program event settlement transaction.
- **Line Description**: A single line description of the program event settlement transaction, which can be used when displaying the transaction on bills or reports.
- **UOM/TOU/SQI**: The unit of measure, time of use period, and service quantity identifier combination used to define the interval usage values used in the program event settlement calculation. These are derived from the UOM/TOU/SQI defined on the business flag type of the business flag that created the program event settlement transaction.
- **Interval Size**: The size of the interval usage values, represented as hours:minutes:seconds (HH:MI:SS). This is derived from the Interval Size defined on the business flag type of the business flag that created the program event settlement transaction. This should match the interval size used for interval price sets.
- **Maximum Drop**: The largest demand reduction interval value for the event
- **Minimum Drop**: The smallest demand reduction interval value for the event
- **Average Drop**: The average demand reduction interval value for the event
- **Settlement Quantity**: The total quantity used for the program event settlement.
  * For "kWh Avoided" program settlements, this is the "kwh avoided" consumption saved for the event period, calculated by adding together the individual "kWh avoided" values for the event period
  * For "Combined kW Drop" program settlements, this is the average of the "Maximum Drop" values selected by the Program Settlement Rule - Demand Based program settlement calculation rule.
- **Actual Consumption**: The actual consumption for the customer for the event period (applicable only for "kWh Avoided" program event settlement transactions)
- **Program Settlement Amount**: The program event settlement transaction's monetary amount, calculated by multiplying interval values by prices that apply to the event.
  * In the case of "kWh Avoided" program settlements, the interval values are individual "kWh avoided" interval values (from the "kWh avoided" business flags received from an external analytics application)
  * In the case of "kW Drop" program settlements, the interval values are those derived from individual "kW Drop" program settlements (based on the Program Settlement Calculation Rule configuration).
- **Intervals**: Details of the interval values and prices used in the program settlement transaction calculation, including:
  * **Service Quantity Data Date/Time**: The date and time of the interval value (per the UOM/TOU/SQI and price (if applicable)
  * **Quantity**: The interval value (per the UOM/TOU/SQI) for the interval date and time
  * **Service Quantity Interval Condition**: The condition code of the interval value (per the UOM/TOU/SQI) for the interval date and time
  * **Price**: The interval price for the interval date and time (applicable only for "kWh Avoided" program settlement transactions)
  * **Program Settlement Amount**: The monetary amount for the date and time, calculated by multiplying the "kWh avoided" interval value times the interval price at the interval date and time (applicable only for "kWh Avoided" program event settlement transactions)

Copyright © 2026, Oracle and/or its affiliates.

- **Issues**: One or more messages related to the program event settlement transaction.

Example - KW Avoided

An event scheduled for 12:00 PM to 6:00 PM on February 11, 2023 results in the following "kWh avoided" values for a particular customer:

Date/Time	kWh Avoided
02-11-2023 12:00:00PM PST	5.00
02-11-2023 01:00:00PM PST	3.00
02-11-2023 02:00:00PM PST	2.00
02-11-2023 03:00:00PM PST	2.00
02-11-2023 04:00:00PM PST	3.00
02-11-2023 05:00:00PM PST	8.00
02-11-2023 06:00:00PM PST	6.00

This represents a total "Consumption Saved" of 29.00.

Later, the system receives the following price data for this event period:

Date/Time	Price
02-11-2023 12:00:00PM PST	0.45
02-11-2023 01:00:00PM PST	0.45
02-11-2023 02:00:00PM PST	0.35
02-11-2023 03:00:00PM PST	0.25
02-11-2023 04:00:00PM PST	0.35
02-11-2023 05:00:00PM PST	0.45
02-11-2023 06:00:00PM PST	0.45

When a program event settlement transaction is calculated for this event, the details of the program event settlement transaction would be as follows:

- UOM/TOU/SQI: kWh
- Interval Size: 01:00:00
- Consumption Saved: 29.00
- Actual Consumption: 430
- Program Settlement Amount: 12.15
- Intervals:

Service Quantity Data Date/Time	Quantity	Service Quantity Interval Condition	Price	Program Settlement Amount
02-11-2023 12:00:00PM PST	5.00	999999	0.45	2.25
02-11-2023 01:00:00PM PST	3.00	999999	0.45	1.35
02-11-2023 02:00:00PM PST	2.00	999999	0.35	0.70

Copyright © 2026, Oracle and/or its affiliates.

Service Quantity Data Date/Time	Quantity	Service Quantity Interval Condition	Price	Program Settlement Amount
02-11-2023 03:00:00PM PST	2.00	999999	0.25	0.50
02-11-2023 04:00:00PM PST	3.00	999999	0.35	1.05
02-11-2023 05:00:00PM PST	8.00	999999	0.45	3.60
02-11-2023 06:00:00PM PST	6.00	999999	0.45	2.70

System Data

Program event settlement transactions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Transaction (D1-USAGETRAN)
Business Object(s)	Program Event Settlement Transaction (X1-EventSettlementTransaction) Program Event Settlement Transaction (X1-ProgramEventSettlementTr)

Batch Processing

Program event settlement transaction processing is supported by the following monitor batch processes:

- Process Program Settlement (X1-PREVS): Used to transition program event settlement transactions in the "Pending" state to the "Calculate" state.
- Retry Program Settlement (X1-RTEVS): Used to retry program event settlement transactions in the "Issue Detected" state.
- Recalculate Program Settlement (X1-RCEVS): Used to recalculate program event settlement transactions in the "Calculated" state. This is used in the case of a rebill, when updated measurement data is received from the customer information system, or when updated "kWh avoided" data is received from an external analytics application.

Use the **Batch Control** portal to learn more about the parameters used by these batch processes.

Correcting Calculated Program Event Settlement Transactions When Measurement Data Changes

In the event that updated measurement data is received after a program event settlement transaction has been calculated, but before it has been included in a customer settlement transaction, the program event settlement transactions must be recalculated and corrected. For example:

- A customer participates in a program event in May, and a program event settlement transaction is created for the event in "Pending" status.
- Measurement data for the customer for the month of May is received in June, and the program event settlement transaction is calculated and updated to the "Calculated" state.

Copyright © 2026, Oracle and/or its affiliates.

- Before the program event settlement transaction been included in a Customer Settlement. updated measurement data for the customer is received.
- The program event settlement transaction must be recalculated to account for the change in measurement data (which could impact the baseline calculation results used in the transaction).

When updated measurement data is received and processed (via Direct Measurement Processing) the Create UT Correction for Baseline Derived SQs (X1-CRUTCORRB) Post-VEE algorithm on the IMD Type creates a Usage Transaction Correction activity (details regarding parameters for this algorithm can be found in the **Detailed Description** of the X1-CRUTCORRB Algorithm Type). When this activity is processed, the Recalculate Program Event Transaction Due to Measurement Change (X1-RECLPETMC) algorithm recalculates the program event settlement transaction and marks the transaction with a reason code of "Measurement Change".

Additionally, program event settlement transactions can be recalculated and corrected using the following options:

- Using the Recalculate Program Settlement (X1-RCEVS) batch process (see Batch Processing). When using this option, the Recalculation Reason parameter must be set to "X1MC" (Measurement Change).
- Using the "Re-Calculate" button in the **Record Actions** section of the program event settlement transaction. When using this option, you must select the "Measurement Change" reason for the recalculation.

About Program Incentives and Fees

Program incentives and fees are program settlement transactions used to calculate incentives and fees for program subscriptions, based on rules configured on the program subscription's program. Incentives include sign-up incentives for enrollments, referral rebates, and incentives for remaining enrolled during the peak season. Penalties or fees are applied in certain circumstance, for instance if they terminate their enrollments before the contracted period, or when the utility incurs charges related to device repairs, device upgrades, premium device fees.

Incentives and fees transactions are created either ad-hoc from the **Program Subscription** portal, or via a regularly scheduled batch process. In both cases, the process performs calculations for program subscriptions based on the program rule(s) configured on the program subscription's program. The Calculate Incentives and Fees (X1-CALIF) batch process should be run on a regular basis to monitor program subscriptions enrolled on in a device control program with an Incentive and Fees program rule in effect as of the batch run date. The Calculate Incentives and Fees (X1-CALIF) batch process creates transactions for all eligible rule types configured on the program rule.

When run either manually or via batch, the process creates the incentives and fees transaction, and when the program transaction enters the "Calculate" state, the Calculate Incentives and Fees (X1-CALCIFT) algorithm executes the Program Settlement Calculation Group defined on the Program Rule.

Program incentive and fee transactions contain the following data:

- **Program Subscription**: The program subscription for which the transaction was run.
- **Start Date/Time and End Date/Time**: The date and time range when the subscription was run.
- **Status**: The status of the program transaction. Statuses include Pending, Error, Discarded, and Complete.

Copyright © 2026, Oracle and/or its affiliates.

- **Program Rule**: The program rule on which the transaction is based.
- **Rule Type Category**: The category for the incentive and fee, from the Incentive and Fee Type Category (X1-IncentiveFeeTypeCatLookup) extendable lookup
- **Calculation Group**: The program settlement calculation group containing the program settlement calculation rules executed as part of the transaction.
- **Interim Calculations**: The interim results of the calculations performed by the program settlement calculation rules, including the variables used in the calculations and the value of each. These are results are based on variables marked as "FPCO" (For Calculation Purposes Only) on the program settlement calculation rules.
- **Calculated Amounts**: The interim results of the calculations performed by the program settlement calculation rules, including the variables used in the calculations and the line description and calculated amount for each.

System Data

Incentives and Fees transactions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Transaction (D1-USAGETRAN)
Business Object(s)	Incentives and Fees Transaction (X1-IncentivesFeesTransaction)

About Customer Settlement Transactions

Customer settlement transactions represent credits that will be applied to customer's bill based on all of the events in which the customer participated during a given time period. In other words, customer settlement transactions represent the total of all of the customer's program settlement calculations for a given time period.

Customer settlement transactions are created upon requests from a customer information system (CIS). Customer settlement transaction requests are sent in a file which is processed by a batch process which creates the individual customer settlement transactions.Customer settlement transaction requests include the customer contact or account or metered service, the program, the start and end date for the requested time period, and the type of request. These requests are typically sent on an annual basis (such as at the beginning or end of the year), but customer settlement transactions are also created in the following specific situations:

- Final Bill: When a customer moves out or stops their service, a customer settlement transaction is created and applied to their final bill.
- Re-Bill: When a customer is re-billed, the customer settlement transaction is recalculated for inclusion on the new bill.
- Customer Unenrollment: When a customer unenrolls from a program, a customer settlement transaction is created and applied to their next bill.

Customer settlement transactions are created in the "Pending" state. Pending customer settlement transactions are processed by the Process Customer Settlement (X1-PRCS) batch process which transitions them the "Calculate" state, in which all of the customer's program settlements for the requested time period are collated and totalized. These are then sent back to the customer information system for application on the customer's bill.

For customer settlement transactions for programs with an **Program Settlement Method** of "End of Season" (typically associated with demand-based programs), the Calculate Customer Settlement (X1-CALCCST) algorithm creates a "Combined kW Drop" program settlement

Copyright © 2026, Oracle and/or its affiliates.

transaction for the customer settlement transaction time period which calculates the average of a set of the "Maximum Drop" values from "Calculation Deferred" program settlement transactions that fall within the customer settlement transaction time period. The customer settlement transaction uses the results of the "Combined kW Drop" program settlement transaction. See About Program Settlement Transactions for more information about "Combined kW Drop" program settlements.

Once customer settlement transaction calculations are complete, they can be sent back to the customer information system for application on the customer's bill. See Downloading Customer Settlement Transactions for more information about how customer settlement transaction information is sent to the customer information system.

By default, the credits from customer settlement transactions are applied to the metered service for which they were calculated. However, in the case of a group of related metered services (such as a set of stores belonging to a common franchise), customer settlement transactions can be applied to a single "parent subscription" via the **Settlement Aggregation Method** (see below). "Parent" subscriptions are defined using the **Settlement Group** attribute on the Program Subscription. A program subscription's **Settlement Group** is a metered service associated with the program subscription's Main Customer, and is used to group program subscriptions for customer settlement transactions (see About Program Subscriptions for more information). "Parent" subscriptions are often used with demand-based programs for commercial customers where credits for overall demand reduction (based on "kW Drop" program settlement transactions) for a group of related accounts (metered services) are applied to the "parent" account.

Customer settlement transactions contain the following data:

- **Metered Service ID**: The metered service (customer) for which the customer settlement transaction was run.
- **Start and End Date**: The start and end date for the customer settlement transaction period.
- **Status**: The current status of the customer settlement transaction.
- **Used on Bill**: A flag that indicates (Yes or No) if the customer settlement transaction has been calculated.
- **Extract**: A flag that indicates (Yes or No) if the customer settlement transaction has been extracted.
- **Customer Settlement Request Type**: The type of customer settlement request. Valid options include Valid options include, End of Season, Final, Rebill, Regular, and Unenrollment.
- **Included Transaction**: The types of transactions included in the customer settlement. This is supplied in the customer settlement request received from the customer information system. Valid options include:
  - **Both**: Include both program event settlement transactions and incentive and fee transactions in the customer settlement transaction.
  - **Event Settlement**: Include only program event settlement transactions in the customer settlement transaction.
  - **Incentive and Fee**: Include only incentive and fee transactions in the customer settlement transaction.
- **Program**: The program associated with the customer settlement.
- **Method**: A flag that indicates the method by which customer settlements are applied to customer accounts. Options include:

Copyright © 2026, Oracle and/or its affiliates.

- **Individual**: Indicates that customer settlements should be applied to the individual customer accounts (represented by metered services).
- **Grouped**: Indicates that customer settlements should be applied to "Settlement Group" customer accounts (metered services specified as a **Settlement Group** on a program subscription).

- **Calculation Details**:
  - **UOM/TOU/SQI**: The unit of measure, time of use period, and service quantity identifier combination used to define the interval usage values used in the program settlement transactions that comprise the customer settlement calculation
  - **Customer Settlement Quantity**: The total quantity on which the customer settlement is base
    - For customer settlements based on "kwh Avoided" program settlements, this is the consumption saved for the customer settlement, calculated by adding together the individual program settlement transaction "Consumption Saved" values
    - For customer settlements based on "kW Drop" program settlements, this is the average of the "Maximum Drop" values from the individual program settlement transactions
  - **Total Actual Consumption**: The total actual consumption for the customer for the customer settlement, calculated by adding together the individual program settlement transaction "Actual Consumption" values (applies only to customer settlements for consumption-based programs)
  - **Customer Settlement Amount**: The monetary amount for the customer settlement.
    - For customer settlements based on "kwh Avoided" program settlements, this is calculated by adding together the individual program settlement transaction "Program Settlement Amount" values.
    - For customer settlements based on "kW Drop" program settlements, this is the amount from the "Combined kW Drop" program settlement transaction.
  - **Issues**: One or more messages related to the customer settlement transaction.
- **Program Settlements**: The program settlement transactions used for the the customer settlement transaction calculations

System Data

Customer settlement transactions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Transaction (D1-USAGETRAN)
Business Object(s)	Customer Settlement Transaction (X1-CustomerSettlementTransaction)

Batch Processing

Customer settlement transaction processing is supported by the following monitor batch processes:

- Process Customer Settlement (X1-PRCS): Used to transition customer settlement transactions in the "Pending" state to the "Calculate" state.
- Retry Customer Settlement (X1-RTCS): Used to retry customer settlement transactions in the "Error" state.

Copyright © 2026, Oracle and/or its affiliates.

Use the **Batch Control** portal to learn more about the parameters used by these batch processes.

About Interval Prices and Price Sets

Interval price sets represent set of interval pricing values used in various types of calculations, including program settlement transactions used with Customer Program Management, settlement calculations used with Market Settlements Management, or pricing calculations used in billing or rating. Interval price values can include market prices or real-time prices. When using multiple sources of interval prices, each source might be represented by a different interval price set.

Interval prices are individual prices based on a defined interval size (for example, hourly prices).

Interval price are defined by the following data:

- **Interval Price Set Type**: The interval price set type on which the interval price set is based. The interval price set type defines a number of attributes of the price set, including the interval size for the price intervals (such as 1 hour, 15 minutes, and so on), and the unit of measure, time of use period, and service quantity identifier used to define the price values.
- **External Identifier**: An identifier for the interval price set used in external systems
- **Time Zone**: The time zone associated with the interval price set
- **Interval Prices**: A set of interval price values belonging to the interval price set, including the following:
  - **Price Date/Time**: The date and time of the interval price
  - **Price**: The price value (based on the unit of measure defined for the interval price set's type)

System Data

Interval Price Sets are based on the following system data:

Data Type	Description
Maintenance Object	Measuring Component (D1-MEASRCOMP)
Business Object(s)	Interval Price Set (D1-IntervalPriceSet)

Interval Prices are based on the following system data:

Data Type	Description
Maintenance Object	Measurement (D1-MSRMT)
Business Object(s)	Interval Prices (D1-IntervalPrices)

Event Impact Measurement and Verification

Event Impact Measurement and Verification (M&V) for Distributed Energy Resource (DER) and Demand Response (DR) programs refers to the overall process of accurately measuring and verifying the actual energy savings or demand reductions achieved by implementing programs of these types, such as rooftop solar panels or battery storage, OR load control devices, like

Copyright © 2026, Oracle and/or its affiliates.

Smart Thermostats or Smart Water Heaters, where customers eventually end up reducing their electricity usage during peak demand periods.

The determination of energy savings achieved for an individual program event using baseline calculation methodologies allows utilities to perform the corresponding financial settlements. Refer to About Program Event Settlement Transactions for more information about how these settlements are calculated.

Event Impact Measurement and Verification (M&V) can be used to determine program-level demand reductions (either achieved or projected to be achieved) and ensure the effectiveness and reliability of these initiatives and can provide transparent data for market participants and regulators. This plays a key role in ongoing assessment and improvement in the program development cycle. At a high-level, this means evaluating the overall effects of a program (load reductions or load increases) related to a particular event or set of events, energy savings (positive or negative) and also cost effectiveness among others. The effects may be determined at the program level or at any level of granularity.

M&V calculations are performed using a type of regression analysis known as "Difference-in-Differences" (or DiD). At a high level the DiD methodology (as defined by Wikipedia), is "a statistical technique used in quantitative research that attempts to mimic an experimental research design using observational study data, by studying the differential effect of a treatment on a 'treatment group' versus a 'control group' in a natural experiment". When applied to demand response events, this analyzes the change in electricity consumption between a group of customers participating in a demand response program (the *treatment group*) compared to a similar group not participating (*control group*), both before and after the demand response event, allowing the outcome to isolate the causal effect of the program on electricity usage by accounting for any pre-existing trends in consumption patterns across the groups. This method using *Treatment* and *Control* groups is effective for impact estimation of relatively similar or uniform groups of customers, such as residential or small commercial customers , where several hundred or several thousand customers might participate in a program. The method is less effective for evaluating smaller numbers of customers or large commercial or industrial customers, because the treatment-control differences will have too much random error to be reliable.

As noted above, M&V calculations are performed for two different types of groups of customers:

- **Treatment Groups:** Groups of customers enrolled in a demand response program, receiving price signals or direct load control during the event. Treatment groups are also referred to as **Device Population Groups**.
- **Control Groups:** Groups of similar customers who are not part of the program and experience normal electricity prices during the event.

Both Device Population (or Treatment) groups and Control groups are defined using Entity Groups. See Understanding Device Population and Control Groups in the *Adminitrative User Guide* for more information.

As part of the DiD regression calculations, average hourly load is calculated for the following two regression periods:

- **Pre-Treatment**: A single period of time, typically 3 hours immediately before and adjacent (without gaps or overlaps) to the **Treatment** period.
- **Treatment**: A period comprising Pre-Event, Event, and Post-Event periods. Pre-Event and Post-Event periods are typically 2-3 hours each, while the Event period itself can up to 8 hours.

More specifically, a load model is constructed for each participant in both the **Device Population** group and the **Control** group using the interval meter reads for the **Pre-Treatment**

Copyright © 2026, Oracle and/or its affiliates.

period and the **Treatment** period. Subsequently a regression analysis is performed across the data to compute a ratio of the estimated **Treatment** period average hourly load of the **Device Population** group assuming no event to the **Control** group **Treatment** period average hourly load, and eventually to the **Device Population** group hourly (kW) Load Impact.

This ratio and other outputs from the regression analysis are captured in M&V Transactions, and additional calculations can be performed using M&V Calculation Rules.

M&V Processing and Dynamic Aggregation

M&V processing leverages functionality Dynamic Aggregation in a number of specific ways.

First, it uses Dimension Scanning to identify changes to Device Population and Control groups, and creates aggregation measuring components for each device population group. This process is run on a regular basis.

Secondly, it uses Find Constituents processing as part of the M&V calculation and regression process to identify the measuring components for both Device Population and Control groups. However, instead of aggregating measurement data for these measuring components, instead it stores the measuring components in a global temporary table called the Entity Group Constituents Temporary Table (D1_GTT11). The algorithms that execute the regression calculations read the measuring components from this table and aggregates the data for the Device Population group and the Control group.

Both of these processes requires configuration of Data Sources, Dynamic Aggregation Measuring Component Types, Aggregation Groups, and Measuring Component Sets.

- About Regression Aggregation Measuring Components
- About M&V Subscriptions
- About M&V Transactions
- About Service Point Event Outcome Snapshots

About Regression Aggregation Measuring Components

Regression aggregation measuring components are Aggregation Measuring Components used to represent Device Population groups. Unlike standard aggregation measuring components, these do NOT store aggregated measurement data. Instead, they are used to orchestrate M&V processing for their associated Device Population groups and their corresponding Control Groups (see Understanding Device Population and Control Groups in the *Administrative Use Guide* for more information about the relationship between Device Population and Control groups).

Regression aggregation measuring components are created via the Dimension Scanning process based on changes to Devce Populaion groups.

Regression aggregation measuring components are defined by the following:

- **Initiating Measuring Component Set**: The measuring component set that triggered creation of the aggregation measuring component via the Dimension Scanning process (see Dimension Scanning for more information).
- **Status**: The current status of the measuring component. Valid values are "Active", "Aggregate", and "Inactive".
- **Service Type**: The type of service (electric, water, and so on) the measuring component supports
- **Time Zone**: The time zone in which the measuring component operates

Copyright © 2026, Oracle and/or its affiliates.

- **Processing Information**: Details of aggregation processing for the aggregation measuring component, including the date and time through which aggregation has been performed.
- **Measuring Component Attributes**: Attributes that describe the dimensions and criteria for the aggregation measuring component (used by the Dimension Scanning process)

System Data

Regression aggregation measuring components are based on the following system data:

Data Type	Description
Maintenance Object	Measuring Component (D1-MEASRCOMP))
Business Object(s)	Regression Aggregator for M&V (X1-RegressionAggregatorMV)

Regression Aggregation Measuring Component Processing

As noted above, regression aggregation measuring components orchestrate M&V processing for their associated Device Population groups and their corresponding Control Groups.

Over time, as program events take place, Event Participation records record which customers participate in which events and the outcome of those events (such as "Successful", "Error", and so on). These are used to create Service Point Event Outcome Staging records.

On a regular basis, the Aggregation by MC Set Monitor (D1-AMCS) batch process triggers processing for aggregation measuring components in the "Active" state. This initiates a series of algorithms which do the following:

- Check for participation in program events
- Build a list of customers from the Device Population group who successfully participated in the event (Find Constituents) in the Entity Group Constituents Temporary Table (D1_GTT11).
- Trim the list of constituents by retrieving related Service Point Event Outcome records and deleting records from the Entity Group Constituents Temporary Table (D1_GTT11) in the event of any of the following:
  - The record indicated an outcome of non-responsive device, or canceled
  - The record has the Multiple Entity Group flag set to "Yes"
  - Any customer where another thermostat for the service point has participated in a different event for the same external event ID.
  - If no Service Point Event Outcome record is found for a customer in the Device Population group
- Perform a "Difference in Differences" regression analysis, aggregating and comparing data from the Device Population group to corresponding data from the Control group, and create a M&V transaction, which uses M&V Calculation Rules to perform calculations based on the regression outcome quantities
- Check for participation in any remaining unprocessed qualifying events
  - If no event occurred, the measuring component remains in the "Active" state
  - If another event is found, the measuring component is set to the "Aggregate" state (and can be transitioned back to the "Active" state via the Transition to Default Next Status (Java) (F1-AT-RQJ) batch process.

Copyright © 2026, Oracle and/or its affiliates.

About M&V Subscriptions

M&V subscriptions represent the enrollment of a group of customers (defined by a "Device Population" entity group) in a program. The same group of customers can have separate subscriptions for different programs.

M&V subscriptions are created whenever a new Device Population group is created or changed. A M&V Subscription's Device Population group defines the group of customers whose data is aggregated and processed when Measurement and Verification Transactions are created.

M&V subscriptions are defined by the following:

- **M&V Subscription Type**: The M&V subscription type associated with the Device Population group and program. M& V subscription types define certain parameters and attributes of a class of M&V subscriptions.
- **Division**: The division in which the customers in the Device Population group reside.
- **Start** and **End Date/Time**: The date and time on which the subscription record started, and ended (if applicable).
- **Entity Group**: The Device Population group
- **Program**: The program in which the Device Population group customers are enrolled.
- **Time Zone**: The time zone associated with the subscription.
- **M&V Calculation Groups**: The calculation groups defined for the subscription and used when creating M&V transactions for the Device Population group, including:
  - **M& V Calculation Group**: The M&V Calculation Group
  - **Effective Date/Time**: The date and time at which the calculation group is in effect.
  - **Expiration Date/Time**: The date and time at which the calculation group expires.

System Data

M&V subscriptions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Subscription (D1-US)
Business Object(s)	M&V Subscription (X1-MsrmtVerifSubscription)

About M&V Transactions

M&V transactions capture the results of the regression analysis and calculations preformed on those results based on data for the Device Population (treatment) group and the Control group. This includes the following coefficients and calculated values:

- Intercept
- Coefficient for Time
- Coefficient for Treatment
- Coefficient for Time * Treatment
- Control group average hourly load over the pre-event reference period (scalar value)

Copyright © 2026, Oracle and/or its affiliates.

- Treatment group average hourly load over the pre-event reference period (scalar value)
- Treatment group hourly average load over the event period (interval values)
- Control group hourly average load over the event period (interval values)

M&V transactions are created by monitor algorithms on aggregator measuring components associated with Device Population groups.

M&V transactions contain the following data:

- **M&V Subscription**: The M&V subscription associated with the event for which the M&V transaction was created
- **Program Event**: The event for which the M&V transaction was created
- **Start Date/Time**: The start date and time of the event
- **End Date/Time**: The end date and time of the event
- **Regression Statistics**: Statistics from the regression calculation, including:
  - **Coefficient of Determination**
  - **Multiple R**
  - **Standard Error**
  - **Number of Observations**
- **Coefficient Statistics**: Statistics for each of the following coefficients: Intercept (b0), Coefficient for Time (b1), Coefficient for Treatment (b2), and Coefficient for Time * Treatment (b3), including:
  - **Standard Error**
  - **Low Confidence Interval**
  - **High Confidence Interval**
- **Issues**: One or more messages related to the transaction.

System Data

M&V transactions are based on the following system data:

Data Type	Description
Maintenance Object	Usage Transaction (D1-USAGETRAN)
Business Object(s)	M&V Transaction (X1-MsrmtVerifTransaction)

Batch Processing

M&V transaction processing is supported by the following monitor batch processes:

- Retry M&V Transaction (X1-RTMVT): This batch process monitors M&V Transactions and advances them to the next state, and will retry M&V Transactions events in error.
- Recalculate M&V Transaction (X1-RCMVT): This batch process monitors M&V Transactions to recalculate and advances them to the next state.

Use the **Batch Control** portal to learn more about the parameters used by these batch processes.

About Service Point Event Outcome Snapshots

Copyright © 2026, Oracle and/or its affiliates.

Service Point Event Outcome Snapshots capture a summary of event data at a service point level for program participants that have multiple controllable devices, such as thermostats, including both residential and commercial customers.

Service Point Event Outcome Snapshots are defined by the following:

- **Metered Service Point**: The metered service point for which the event outcome snapshot was created
- **Treatment Group**: The Device Population group to which the devices that participated in the event belong
- **Program Event**: The program event for which the outcome snapshot was created
- **Program Subscription**: The program subscription associated with the event outcome snapshot
- **Service Point Event Outcome**: The outcome of the service point's participation in the event. Possibly outcomes include "Successful participation", "Non-responsive device", "In-flight override", "Mixed", and "Canceled".
- **Control Group**: The control group to be used for M&V processing for the event
- **Parent**: A flag that indicates (Yes or No) if the program subscription associated with the event outcome snapshot is a parent subscription
- **Device Count**: The number of devices that participated in the event
- **Multiple Entity Group**: A flag that indicates (Yes or No) if the event outcome snapshot is based on multiple entity groups
- **Multiple Program Event**: A flag that indicates (Yes or No) if the event outcome snapshot is based on multiple program events
- **Service Point Event Outcome Quantity**: A list of quantities captured in the event outcome snapshot, including the following for each:
  - **Sequence**: The sequence of the quantity
  - **SQI**: The service quantity identifier for the quantity
  - **Date/Time**: The date and time of the quantity
  - **UOM**: The unit of measure for the quantity
  - **Quantity**: The quantity

System Data

Service Point Event Outcome Snapshots are based on the following system data:

Data Type	Description
Maintenance Object	Service Point Event Outcome (D1-SPEVTOUTC)
Business Object(s)	Service Point Event Outcome (X1-ServicePointEventOutcome)

Batch Processing

Service Point Event Outcome snapshot processing is supported by the following monitor batch processes which create Service Point Event Outcome Snapshots from Event Participation and M&V transaction records:

- Create Service Point Event Outcome Staging (X1-SPEOC): This batch process creates Service Point Event Outcome Staging records from Event Participation records that contain the basic data used for the snapshot

Copyright © 2026, Oracle and/or its affiliates.

- Create SP Event Outcome Quantity Staging (X1-SPEOQ): This batch process creates Service Point Event Outcome Quantity Staging records from M&V Transaction records that contain Service Point Event Outcome Quantities

Use the **Batch Control** portal to learn more about the parameters used by these batch processes.

About Customer Program Service Tasks

Customer program service tasks are used to manage various customer-focused processes such as enrollment, unenrollment, and service calls.. A service task is created for each customer process and orchestrates the various steps and tasks involved in that process.

System Data

Customer program service tasks are based on the following system data:

Data Type	Description
Maintenance Object	Service Task (F1-SVCTASK)
Business Object(s)	Customer Program Enrollment Task (X1-DigitalAssetEnrollTask)
	Customer Program Unenrollment Task (X1-DigitalAssetUnenrollTask)
	Program Service Call (X1-ProgramServiceCall)
	DER Interconnection Request Task (X1-DERInterconnectionTask)
	Customer Program Enroll Additional Device Task (X1-DAAddDeviceToEnrollTask)

About Enrollment and Unenrollment

Enrollment involves identifying a specific customer that is eligible for a particular program, and establishing a relationship between the customer, the program, and if appropriate, a device location where one more controllable devices are (or have been) installed. The primary result of the enrollment process is a program subscription that links a customer with a program and (and in most cases) a device location. The process can also involve one or more actions performed to set up the controllable devices. For example, in the case of devices which the utility provides and installs at the customer's location, the enrollment process creates a work activity used to schedule a crew to install the device. Another common device setup action involves issuing a "device status check" command request in which the control system validates that the device can receive messages from the control system.

Unenrollment involves removing a previously enrolled customer from a program, and deactivating program subscriptions and related controllable devices.

The enrollment and unenrollment processes are managed by the following service task business objects:

- Customer Program Enrollment Task (X1-DigitalAssetEnrollTask)
- Customer Program Unenrollment Task (X1-DigitalAssetUnenrollTask)

To understand the enrollment and unenrollment processes, it’s important understand the lifecycle of these service task business objects. The table below outlines the lifecycle for the above business objects. Specifics for Enrollment and Unenrollment are noted in the descriptions below.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Pending	The initial state for service tasks activities. The service task is transitioned to the next state via a monitor process.
Validate	Enrollment: an Enter algorithm validates the enrollment request, including: • If an existing customer can be retrieved for the customer program enrollment request and the request contains the necessary enrollment details to continue processing. • If the program is accepting enrollments • If the metered service points division is same as the programs division • If asset information is provided when the Installation Method is "Bring Your Own" • If an in store asset exists that is valid for the program when the Installation Method is "Bring Your Own" and Asset Ownership is "Utility Owned" Unenrollment: an Enter algorithm validates that the unenrollment request contains the necessary details to derive the program subscription record to be unenrolled. The service task is transitioned to the next state via a monitor process.
Validation Error	If the service task fails any of the validations in the Pending state, it enters this state. Service tasks in this state can be corrected and retried.
Validated	If the service task passes the validations in the Validate state, it enters this state.
Check Eligibility	Enter algorithms validate the the customer is eligible for the program (based on the Eligibility Criteria defined for the program's Initiative), and that the customer meets the program qualifications (based on the program's qualification questions). Applies to Enrollment service tasks only.
Eligibility Error	If the customer is either not eligible for the program or doesn't meet the program qualifications, the service task fails and enters this state. Service tasks in this state can be corrected and retried. Applies to Enrollment service tasks only.
Re-Evaluate Eligibility	If the service task failed and during the Check Eligbility state, it can enter this state temporarily before returning to the the Check Eligbility state. Applies to Enrollment service tasks only.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Setup Enrollment	An Enter algorithm creates a Program Subscription (in the "Inactive" state) and a Device Location (in the "Active" state). **Note** If the **Location Required Source** field is set to "No" on the program's program type, creation of the device location is skipped. If the Installation Method is "Bring Your Own" and Asset Ownership is "Customer Owned", a controllable device is also created and attached to the device location. Applies to Enrollment service tasks only.
Enrollment Setup Error	If an error occurs in the Setup Enrollment state, it enters this state. Service tasks in this state can be corrected and retried. Applies to Enrollment service tasks only.
Setup Unenrollment	An Enter algorithm deactivates the Program Subscription. Applies to Unenrollment service tasks only.
Unenrollment Setup Error	If an error occurs in the Setup Unenrollment state, it enters this state. Service tasks in this state can be corrected and retried. Applies to Uennrollment service tasks only.
Setup Device	Enrollment: An Enter algorithm determines the device setup actions configured under customer program enrollment service task type and creates the relevant setup actions required to setup the controllable device. Each setup action is captured in the Device Setup section of the service request (as well as in the service task's log). **Note** If the **Location Required Source** field is set to "No" on the program's program type, creation of the device is skipped. Unenrollment: An Enter algorithm determines the device setup actions configured under customer program unenrollment service task type and creates the relevant setup actions required to deactivate the controllable device. Each setup action is captured in the Device Setup section of the service request (as well as in the service task's log). See Device Setup Actions, below for more information.
Device Setup Error	If one or more device setup actions cannot be determined, the request is transitioned to this state. Service tasks in this state can be corrected and retried.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Awaiting Device Setup	The service task remains in this state until all device setup actions are complete. The service task is transitioned to the Device Setup Actions stage via a monitor process until all of the actions configured on the service task type have been completed.
Completed	The service task enter this state when all device setup actions have successfully completed. Enrollment: An Enter algorithm sets the Program Subscription's Status to "Active". Unenrollment: An Enter algorithm populates the "End Date/Time" on the Program Subscription and sets its Status to "Inactive", and any controllable devices on the Program Subscription are deactivated.
Discarded	Service tasks discarded in previous states enter this state.
Canceled	Service tasks canceled in previous states enter this state.

Specifics of the service task, such as the type of program subscription and device location to creates and the types of device setup actions required to complete the enrollment are defined by the service task type associated with the program in which the customer is enrolling (or from which the customer is unenrolling). See Understanding Customer Program Service Task Types in the *Administrative User Guide* for more information.

Device Setup Actions

Enrollment or unenrollment processes can include a number of device setup actions performed to install, investigate, or test controllable devices as part of the enrollment or unenrollment process. Supported device setup actions include:

- **Create Activity**: Creates Service Order Management field activities to perform device installation or investigation.
- **Create To Do**: Creates To Do Entries.
- **Create Work Order**: Create Work and Asset Management work orders to perform device installation or investigation.
- **Initiate Controllable Device Command**: Create and send commands to controllable devices to test if they're capable of communicating with the control system.

See Understanding Customer Program Service Task Types in the *Administrative User Guide* for more information about configuring device setup actions.

Enrollment/Unenrollment Methods

Oracle Utilities Customer Program Management Cloud Service provides two methods of initiating the customer enrollment or unenrollment processes.

The first is through use of the **Process Flow** portal and functionality. The system provides two process flow types to support the enrollment and unenrollment processes:

- Program Enrollment Process (X1-DIGITALASSETENROLLMENT)
- Program Unenrollment Process (X1-DIGITALASSETUNENROLLMENT)

Enrollment/unenrollment service tasks are created by the process flow based on data entered by the user. See Enrolling and Unenrolling Customers using Process Flow for more information about using the **Process Flow** portal for enrollment and unenrollment. See Process Flows in the *Administrative User Guide* for general information the process flow functionality.

Copyright © 2026, Oracle and/or its affiliates.

Customer Program Management also provides an inbound web service to enable submission of enrollment and unenrollment requests from external systems such as a customer information system. The Program Enrollments (X1-ProgramEnrollment) inbound web service provides the following operations:

- Create Program Enrollment (enrollProgram): Used to create an enrollment request to enroll a customer in a program
- Create Program Unenrollment (unenrollProgram): - Used to create an unenrollment request to unenroll (remove) a customer from a program
- Get Program Enrollment (getProgramEnrollment): - Used to retrieve enrollment information
- Get Program Unenrollment (getProgramUnenrollment): - Used to retrieve unenrollment information

About Service Calls

Service calls are created when a customer reports an incident of some kind related to a controllable device installed at their location. Service calls can have different levels of urgency (Emergency or Non-Emergency) and can be used to create a work order to address the incident, or can resolve or escalate a previous service call.

The system provides the following process flow type to support service calls:

- Program Service Call Process (X1-DAPROGRAMSERVICECALL)

Service calls are initiated from the **Current Enrollment** zone in the **Customer Program Summary** portal, and make use of the **Process Flow** portal and functionality. When the user clicks the **Service Call** button on the **Current Enrollment** zone, they are brought to the **Process Flow** portal where they can enter details of the service call. Service call service tasks are created by the process flow based on data entered by the user. See Creating Program Service Calls Using Process Flow for more information about using the **Process Flow** portal for service calls. See Process Flows in the *Administrative User Guide* for general information the process flow functionality.

The service call process is managed by the following service task business object:

- Program Service Call (X1-ProgramServiceCall)

To understand the service call process, it's important understand the lifecycle of this service task business object. The table below outlines the lifecycle for the above business object.

State	Description
Pending	The initial state for service tasks activities. The service task is transitioned to the next state via a monitor process (or as part process flow processing).
Validate	The service task is transitioned to the next state via a monitor process.
Validation Error	If the service task fails any of the validations in the Pending state, it enters this state. Service tasks in this state can be corrected and retried.
Escalated	This state applies only to service calls with an "Escalation" action. An Enter algorithm creates an "Escalated" To Do entry, based on the **Escalation To Do Type** defined on the service task type.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Work in Progress	This state applies only to service calls with a "Create a Work Order" action. An Enter algorithm creates a work order based on the selected template work order. The service task remains in this state until the word order and its related work activities are completed (via a Monitor algorithm).
Completed	The service task enters this state following either the "Escalated" or "Work in Progress" states.
Discarded	Service tasks discarded in previous states enter this state.

Specifics of the service task, such as the type of work order to creates are defined by the service task type associated with the program for which the customer is initiating the service call. See Understanding Customer Program Service Task Types in the *Administrative User Guide* for more information.

Maintaining Controllable Assets and Devices

The **Asset** and **Controllable Asset** portals are used to display and maintain controllable asset and device records.

Refer to About Controllable Assets and Devices for more information.

You can access the portal from **Menu**, select **Asset Management**, then select **Asset**. The **Asset Query** portal and **Asset Search** zone open which you can use to search for the controllable asset/device you wish to view. When you select an asset, the **Controllable Asset** portal opens displaying the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Controllable Asset**: This zone displays basic information about the asset record associated with the controllable asset.
  - Click **Calculate Score** to manually calculate the control score for the asset. The score is calculated based on the Score Calculation Rule configured for the program subscription or program subscription type associated with the program in which the asset is enrolled. See Score Calculation Rule for more information the Score Calculation Rule. If no rule is configured, the **Calculate Score** button does nothing. This option is only available if the asset's **Status** and **Asset Disposition** are "Installed".
- **Attachments**: This zone lists any attachments related to the asset record associated with the controllable asset.
- **Asset Service History**: This zone displays a log of the service history that has been recorded for the life of the asset.
- **Notes**: This zone lists any notes for the asset record associated with the controllable asset.
- **Disposition History**: This zone displays a log of the status and location changes for the life of the asset

The following zones may appear as part of the portal's **Device** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Controllable Device**: This zone displays the device record associated with the controllable asset. (Controllable assets/devices have both asset and device records)
- **Device Attachments**: This zone lists any attachments related to the device record associated with the controllable asset.
- **Device Log**: This zone display log information for the device record associated with the controllable asset.
- **Device Entity Groups**: This zone lists the entity groups to assigned to the device. To add an entity group to the device, click **Add Entity Group to Device**. A dialog opens where you can select the entity group.

The following zones may appear as part of the portal's **Log** tab page:

- **Asset Log**: This zone display log information for the asset record associated with the controllable asset.
- **Asset Sync Requests**: This zone lists data synchronization requests associated with the asset record associated with the controllable asset.
- **Asset Shipping Memos**: This zone lists shipping memos associated with the asset record associated with the controllable asset.

Maintaining Device Locations

The **Device Location** portal is used to display and maintain device locations.

Refer to About Device Locations for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Device Location**. You are brought to a query portal with options for searching for a specific record. Once a device location has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Location**: This zone displays location information for this device location record.
- **Customer Unavailability**: This zone lists periods of time during which one more controllable devices are unavailable to participate in program events. Click **Add** to define a new unavailability period for a controllable device at this location. To edit an existing unavailability period for a controllable device at this location, select the period from the list and click **Edit**. To cancel an existing unavailability period for a controllable device at this location, select the period from the list and click **Cancel**.

Maintaining Facilities

The **Facilities** portal is used to display and maintain facility records.

Refer to About Facilities for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Facility**. You are brought to a query portal with options for searching for a specific record. Once a facility has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Facility**: This zone displays general information about the facility.

Copyright © 2026, Oracle and/or its affiliates.

- **Network Locations Related to Facility**: This zone lists the network locations associated with the facility record.
- **Downstream Service Points Related to Facility**: This zone lists the downstream service points associated with the facility record.

Maintaining Metered Service

The **Metered Service** portal is used to display and maintain metered service records.

Refer to About Metered Service for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Metered Service**. You are brought to a query portal with options for searching for a specific record. Once a metered service records has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Metered Service**: This zone displays general information about the metered service records.
- **Program Subscriptions**: This zone displays a list of "child" program subscriptions associated with the metered service record.
- **Attribute Values**: This zone displays a list of attribute values whose characteristic type are defined as valid characteristic types on the Metered Service Type.
- **Settlement Group Program Subscriptions**: This zone lists program subscriptions that have the metered service defined as their Settlement Group.

Maintaining Metered Service Points

The **Metered Service Point** portal is used to display and maintain metered service points.

Refer to About Metered Service Pointsfor more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Metered Service Point**. You are brought to a query portal with options for searching for a specific record. Once a metered service point has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Metered Service Point**: This zone displays basic information for this metered service point.
- **Service Point Attachments**: This zone lists attachments associated with the metered service point.
- **Metered Services**: This zone lists metered services associated with the metered service point.
- **Device Locations**: This zone lists device locations associated with the metered service point.
- **SP Network Locations**: This zone lists all network locations linked to any facility associated with the metered service point.

Copyright © 2026, Oracle and/or its affiliates.

- **SP Business Flags**: This zone lists the 25 most recent business flags associated with the metered service point.

Maintaining Network Locations

The **Network Location** portal is used to display and maintain network locations.

Refer to About Network Locations for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Network Location**. You are brought to a query portal with options for searching for a specific record. Once a network location has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Network Location**: This zone displays general information about the network location, such as Facility ID, Start and End Date/Time, Feeder, and Substation.
- **SP Related to Network Location**: This zone lists the service points associated with the network location record.

Maintaining Programs and Related Records

This section describes how demand response programs and related data are created and maintained in Customer Program Management.

Information about programs is maintained in Customer Program Management, but other systems such as Oracle Utilities Network Management System are responsible for initiating and creating program events.

- Maintaining Programs
- Maintaining Program Contacts
- Maintaining Contact Communications
- Maintaining Program Subscriptions
- Maintaining Program Control Scores
- Maintaining Program Events
- Maintaining Event Participation
- Maintaining Program and Customer Settlement Transactions
- Maintaining Interval Price Sets

Maintaining Programs

The **Program** portal is used to display and maintain programs.

Refer to About Programs for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program**. You are brought to a query portal with options for searching for a specific record. Once a program has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Program**: This zone displays general information about the program, such as program type, status, and approval profile.
- **Specifications**: This zone lists one or more specifications for devices eligible for the program.
- **Rates**: This zone lists one or more rates used to determine eligibility for customers to enroll in the program (customer must be on one of the listed rates to join the program).
- **Rules**: This zone lists program rules associated with the program.
- **Qualifications**: This zone lists one or more qualification questions used to determine eligibility for customers to enroll in the program.
- **Informational**: This zone lists one or more informational questions used to determine eligibility for customers to enroll in the program.
- **Events**: This zone lists events associated with the program.

The following zones may appear as part of the portal's **Calculation Groups** tab page:

- **Program Settlement Calculation Groups**: This zone lists effective-dated program settlement calculation groups associated with the program.

Approving Program Approval Requests

Programs with an Approval Profile must be approved before they are consider active and eligible for use. Users can use with the **Approval Dashboard** or the **Approval Requests** portal to view and approve program approval requests.

Maintaining Program Contacts

The **Contact** portal is used to display and maintain program contacts.

Refer to About Program Contacts for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program Contact**. You are brought to a query portal with options for searching for a specific record. Once a program contact has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact**: This zone displays general information about the contact, such as the contact's name and contact information (phone number, email address, and so on).
- **Contact Communication List**: This zone lists customer communications for the contact.
- **Point of Contact Service Point**: This zone lists metered service points associated with the contact. Used with Customer Program Management.

Maintaining Contact Communications

You use the **Contact Communication** portal to display and maintain contact communications and interactions.

Refer to About Contact Communications for more information.

Copyright © 2026, Oracle and/or its affiliates.

To access the portal select **Menu**, select **Customer Program Management**, then select **Contact Communication**. You are brought to a query portal with options for searching for a specific record. Once a contact communication has been selected, you are brought to the maintenance portal to view and maintain the selected record.

You can also access the portal from the **Contact Communication List** zone on the **Contact** portal. When you select the communication you wish to view you are brought to a maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact Communication**: This zone displays the selected communication.

Maintaining Program Subscriptions

The **Program Subscription** portal is used to display and maintain program subscriptions.

Refer to About Program Subscriptions for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program Subscription**. You are brought to a query portal with options for searching for a specific record. Once a program subscription has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Subscription**: This zone displays general information about the program subscription, such as the customer, status, and event calculation groups.
  - If the program subscription is for a device control program and has Incentive and Fees Program Rules if effect, you can use the **Calculate** button to initiate calculations of program incentives and fees. A window opens enabling the user to enter a **Cutoff Date** and select one or more eligible **Incentive and Fee Types** (based on the Rule Parameters defined on the Program Rule) to process.
- **Event Participation List**: This zone lists the events in which program subscription was a participant.
  - To opt out of one more more scheduled event, select the events and click **Opt Out**.
- **Program Settlements**: This zone lists program settlements calculated for the program subscription.
- **Incentives and Fees Settlements**: This zone lists incentives and fees calculated for the program subscription.
- **Attribute Values**: This zone lists attribute values whose characteristic type have Program Subscription as characteristic entities and are defined as valid Program Subscription characteristic types on the Program Subscription Type.

Maintaining Program Control Scores

The **Program Control Score** portal is used to display and maintain program control scores.

Refer to About Program Control Scores for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program Control Score**. You are brought to a query portal with options for searching for a specific record. Once a program control score has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Control Score**: This zone displays the selected program control score.

Maintaining Program Events

The **Program Event** portal is used to display and maintain program events.

Refer to About Program Events for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program Event**, or select **Program Event** from the **Program** context menu when viewing either a Program Subscription or Program. When using either approach users are brought to a query portal with options for searching for a specific record. Once a program event has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Event**: This zone displays general information about the program event, such as the program, status, program category, the start and end date/time of the event, and an external event identifier (if applicable).

Maintaining Event Participation

The **Event Participation** portal is used to display and maintain event participation records.

Refer to About Event Participation for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Event Participation**. You are brought to a query portal with options for searching for a specific record. Once an event participation has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Event Participation**: This zone displays general information, such as Program Subscription, Dynamic Option ID, Reservation, Outcome, and so on, about the event participation record.

Maintaining Program and Customer Settlement Transactions

The **Program Settlement** portal is used to display and maintain program and customer settlement transactions, , including program incentives and fees.

Refer to About Program Settlement Transactions, About Program Incentives and Fees, and About Customer Settlement Transactions for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Program Settlement**. You are brought to a query portal with options for searching for a specific record. Once a settlement transaction has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Event Settlement**: This zone displays the selected settlement transaction.
- **Event Settlements**: This zone displays a list the program event settlements totalized in the selected customer settlement transaction.

Copyright © 2026, Oracle and/or its affiliates.

- **Incentives and Fees Settlements**: This zone displays a list the incentives and fees settlements totalized in the selected customer settlement transaction.
- **Baseline Measurements**: This zone displays a list of baseline measurements used in calculating the program event settlement transaction (if applicable).

Maintaining Interval Price Sets

You use the **Interval Price Set** portal to display and maintain interval price sets and interval prices

Refer to About Interval Price Sets for more information.

You can access the portal from **Menu**, select **Interval Price**, then select **Interval Price Set**. You are brought to a query portal with options for searching for a specific record. Once an interval price set has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set**: This zone displays the selected interval price set.
- **Interval Prices**: This zone displays the individual interval prices for the interval price set.

Maintaining Interval Prices

You use the **Interval Prices** zone to maintain interval prices for an interval price set. You can either manually add or upload prices.

To manually add prices, click **Add** in the upper right corner of the zone. The **Interval Prices Add** dialog opens. The **Interval Price Set ID** is prepopulated based on the interval price set.

- Enter the **Price Date/Time** and **Price** values for each interval price in the **Interval Prices** section.
- Click **Save** to save the interval prices.

To upload prices, click **Upload** in the upper right corner of the zone. The **Interval Prices Upload** dialog opens. The **Interval Price Set ID** and **Target Time Zone** are prepopulated based on the interval price set.

- Select the **Source Time Format** (Local or Standard) for the price data.
  - **Local** : Date/time values adjusted to "Standard" time (reflecting DST if applicable) and both are captured on the Measurement table
  - **Standard**: Date/time values are captured in both Standard and Local time on the Measurement table
- Specify the **Source Time Zone** for the price data. This is the time zone from which the price data originated. This allows for conversion to the time zone defined for the interval price set.
- To upload prices, click **Choose File** to search for and select a comma separated values (CSV) file containing price data and click **Submit**. The **Interval Prices Upload** dialog will display the price data contained in the file. Note that you can edit the price data before saving it.
- Click **Save**to save the interval prices.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can add prices using both manual entry and upload, but uploading data will overwrite existing prices for the same date/time.

To update one or more prices, select the price(s) to update and click **Edit**. The **Price** dialog opens displaying the selected prices. Edit the prices as needed and click **OK**. Click **Cancel** to close the dialog without making any changes.

To delete one or more prices, select the price(s) to delete and click **Delete**. The **Price** dialog opens asking you to confirm the deletion. Click **OK** to delete the selected prices. Click **Cancel** to close the dialog without deleting the prices.

Interval Prices File Format

The file format used to upload interval prices are comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column must included in the first row of the input file.

Field	Data Type	Sample Content	Description
dt	DATE	2023-02-11-12.00.00	Date and time for the interval price
value	NUMBER	0.35	Interval price amount

Sample File

The following is a sample file for interval prices spanning a six hour event.

```
dt,value 2023-02-11-12.00.00,0.35 2023-02-11-13.00.00,0.35
2023-02-11-14.00.00,0.35 2023-02-11-15.00.00,0.35 2023-02-11-16.00.00,0.35
2023-02-11-17.00.00,0.35
```

Maintaining Event Impact Measurement & Verification Data

This section describes how records and data used with Event Impact Measurement & Verification (M&V) processing are created and maintained.

- Maintaining M&V Subscriptions
- Maintaining M&V Transactions
- Maintaining Service Point Event Outcome Snapshots

Maintaining M&V Subscriptions

You use the **M&V Subscription** portal to display and maintain M&V subscriptions.

Refer to About M&V Subscriptionsfor more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **M&V Subscription**. You are brought to a query portal with options for searching for a specific record. Once an M&V subscription has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **M&V Subscription**: This zone displays general information about the M&V subscription, such as the customer, status, and event calculation groups.

Maintaining M&V Transactions

You use the **M&V Transaction** portal to display and maintain M&V transactions.

Refer to About M&V Transactions for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **M&V Transaction**. You are brought to a query portal with options for searching for a specific record. Once an M&V transaction has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **M&V Transaction**: This zone displays the selected M&V transaction.
- **SQ Details**: This zone displays service quantity details for the M&V transaction.
- **Baseline Measurements**: This zone displays interval data used in calculating the M&V transaction (if applicable).

Maintaining Service Point Event Outcome Snapshots

You use the Service Point Event Outcome portal to display and maintain Service Point Event Outcome snapshots.

Refer to About Service Point Event Outcome Snapshots for more information.

You can access the portal from **Menu**, select **Customer Program Management**, then select **Service Point Event Outcome**. You are brought to a query portal with options for searching for a specific record. Once a record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Event Outcome**: This zone displays general information about the Service Point Event Outcome.

Maintaining Customer Program Service Tasks

The **Service Task** portal is used to display and maintain customer program service tasks.

Refer to About Customer Program Service Tasks for more information.

You can access the portal from **Menu**, then select **Service Task**. You are brought to a query portal with options for searching for a specific record. Once a service task has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Task**: This zone displays general information about the service task.

Copyright © 2026, Oracle and/or its affiliates.

- **DER Attributes**: This lists Distributed Energy Resource (DER) device attributes captured for DER devices related to the service task. Applicable only for enrollment service tasks involving DER devices.

Distributed Energy Resource (DER) Device Service Tasks

The DER Device Management (X1-DERDeviceManagement) inbound web service is used to perform registration and/or enrollment of Distributed Energy Resource (DER) devices. This inbound web service includes the following operations:

- Register DER Device (registerDERDevice): Used by an external application to register and enroll a DER device. This operation creates a DER asset and device, as well as an enrollment service task to enroll the device in a program specified in the inbound web service request, or if a program is not provided based on the specification defined for the asset. The enrollment service task is created in the "Pending" state. The service task should be processed either manually or via the Service Task Monitor (Deferred) (F1-STKDF) batch process.
- DER Interconnection (interconnectDER): Used by an external application to send interconnection attributes of a DER asset from an external system that handles the interconnection process for the utility. The incoming data is validated against existing asset manufacturer and model combinations, leading to the creation of an interconnection service task to allow the initial creation of the DER asset. Enrollment for assets that require interconnection attributes must be performed separately via the Register DER Device operation. The interconnection service task is created in the "Pending" state. The service task should be processed either manually or via the Service Task Monitor (Deferred) (F1-STKDF) batch process.

The **DER Registration Flag** on the controllable asset type specifies whether registration of DER assets require both an interconnection request (using the DER Interconnection operation) and a DER registration request (using the Register DER Device operation), or just a DER registration request.

Enrolling and Unenrolling Customers Using Process Flow

Enrollment and unenrollment processing can be initiated, monitored, and maintained using the **Process Flow** portal and functionality. This allows users to step through a series of panels on which they enter information about the customer, their location, and the controllable device. See **About Enrollment and Unenrollment** for more information.

Enrollment and unenrollment process flows are initiated from the **Eligible Programs** zone of the **Customer View** portal.

When navigating between panels on the **Process Flow** portal, the Next and Previous buttons allow you to move to the next or previous panels in the process respectively. To save your progress and settings before moving on to the next panel, click **Save**. Click **Finish** on the **Summary** panel to mark the process flow as Closed. To cancel the process, click **Cancel Process**. This transitions the service task to the "Canceled" state. See Action Method in the *Administrative User Guide*for more information the process flow functionality.

The following sections describe the content on the different panels used in enrollment and unenrollment process flows.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the **Location Required Source** field is set to "No" on the program's program type, some enrollment steps are skipped.

Customer

The **Customer** panel is used to specify data about the customer, associated with the enrollment or unenrollment process. This includes the following:

Customer Information: This section displays information about the customer, including:

- **Metered Service**: The customer's metered service record
- **Name**: The customer's name
- **Phone**: The customer's phone number
- **Email**: The customer's email address

Location Information: This section displays information about the customer's location, if applicable, including:

- **Metered Service Point**: The customer's metered service point.

Interaction Details: This section is used to capture details around customer interaction. The information in this section is used to create Customer Communications. See About Customer Communications for more information.

- **Channel**: The channel used for the communication, such as email or phone
- **Class**: The class of the communication, such as enrollment, unenrollment, trouble report, and so on
- **Category**: The category of the communication, such as written or verbal
- **Notes**: Notes associated with the communication

Program Enrollment / Program Unenrollment

The **Program Enrollment** panel is used to determine if the customer is eligible for the program, and includes the following: the program,

Program Information: This section displays information about the program, including:

- **Program**: The program in which the customer will be enrolled (or from which the customer will be unenrolled).
- **Start Date**: The start date of the customer's enrollment in the program (will be used as the program subscription's **Start Date**). Applicable only to enrollment processes.
- **Stop Date**: The stop date of the customer's enrollment in the program (will be used as the program subscription's **End Date**). Applicable only to unenrollment processes.

Qualification Questions: **Enrollment Only**. This section lists the qualification questions associated with the selected program. These are taken from the **Qualifications** configured on the program (see **About Programs**).

Informational Questions: **Enrollment Only**. This section lists the informational questions associated with the selected program. These are taken from the **Informational** questions configured on the program (see **About Programs**).

Copyright © 2026, Oracle and/or its affiliates.

Controllable Devices: **Unenrollment Only**. A list of controllable devices installed at the device location

Interaction Details: This section is used to capture details around customer interaction. See description above.

When the use clicks Next on this panel, the system compares the answers provided in the Qualification sectionanswers the against "correct" answers configured on the program and checks the customer's eligibility for the program based in the program's **Program Initiative** (see **Understanding Program Initiatives**).

This panel is applicable only to enrollment processes.

Related Program Enrollment

The **Related Programs** panel is used to enroll the customer in a related program, and includes the following:

Related Programs: section allows you to select other programs that the customer may be currently enrolled in.

Qualification Questions: This section lists the qualification questions associated with the selected related program, if applicable. These are taken from the **Qualifications** configured on the program (see **About Programs**).

Informational Questions: This section lists the informational questions associated with the selected related program, if applicable. These are taken from the **Informational** questions configured on the program (see **About Programs**).

Interaction Details: This section is used to capture details around customer interaction. See description above.

This panel is applicable only to enrollment processes.

Device Setup

The **Device Setup** panel is used to manage device setup activities (based on the service task's service task type), and can include the following:

Asset Information: This section defines details regarding the asset related to the program enrollment or unenrollment, including:

- **Installation Method**: **Enrollment Only**. The installation method for the controllable devices associated with the enrollment process. Options include:
  - **Re-enroll**: The customer is re-enrolling a utility-owned device
  - **Ship Device**: The utility will ship the device to the customer.
  - **Utility Installation**: The utility will install the device at the customer's device location.
- **Devices to Re-Enroll**: A list of devices to be re-enrolled. At least one of the devices lists must be selected (applicable only if the **Installation Method** is "Re-enroll").
- **Removal Method**: **Unenrollment Only**. The removal method for the controllable devices associated with the unenrollment process. Options include:
  - **Keep**: The customer or utility will keep the device after its removal.
  - **Ship Device**: The customer will ship the device to the utility.
  - **Utility Removal**: The utility will remove the device at the customer's device location.
- **Asset Ownership**: Who owns the asset. Options include:

Copyright © 2026, Oracle and/or its affiliates.

- **Customer Owned**: The customer owns the asset/device.
- **Utility Owned**: The utility owns the asset/device.

Work Order Activities: This section defines details regarding work order activities related to the program enrollment or unenrollment (if applicable). This includes the ability to select an appointment for work activities and field activities. Do the following for each activity type:

- Select a **Crew Shift Type** for the work order activity
- Select a **Crew** for the work order activity (based on the selected crew shift type)
- Enter a date in the **Appointment Date** field to view a list of available appointments for that date . The appointment lists displays available appointments for the activity type, including start and end date and time, and the crew assigned.
- Select the appointment you wish to assign to the activity type. The list of available appointments is based on the **Appointment Date** and the **Appointment Slot Days** parameter defined in the Enrollment/Unenrollment Action Method.

Interaction Details: This section is used to capture details around customer interaction. See description above.

Summary

The **Summary** panel displays a summary of information related to the enrollment or unenrollment process, including:

- **Enrollment Task**: The Service Task used by the process and its current status
- **Customer Information**: Contact and identification for the customer, including the customer's Metered Service record
- **Program Information**: The program the customer will be enrolled in (or unenrolled from) and the start date or stop date (respectively)
- **Location Information**: The device location, including the customer's Metered Service Point
- **Interaction Details**: Customer interaction details from previous panels

Click **Finish** on the **Summary** panel to mark the process flow as Closed. Click **Cancel Process** to cancel the process.

Creating Program Service Calls Using Process Flow

Service call processing can be initiated, monitored, and maintained using the Process Flow portal and functionality. This allows users to step through a series of panels on which they enter information about the customer, their location, and other information. SeeAbout Service Calls for more information.

Service call process flows are initiated from the **Current Enrollment** zone of the **Customer Program Summary** portal.

When navigating between panels on the **Process Flow** portal, the **Next** and **Previous** buttons allow you to move to the next or previous panels in the process respectively. To save your progress and settings before moving on to the next panel, click **Save**. Click **Finish** on the **Summary** panel to mark the process flow as Closed. To cancel the process, click **Cancel Process**. This transitions the service task to the "Canceled" state. See Action Method in the *Administrative User Guide* for more information the process flow functionality,

Copyright © 2026, Oracle and/or its affiliates.

The following sections describe the content on the different panels used in service call process flows.

Customer

The **Customer** panel is used to specify data about the customer associated with the service call. This includes the following:

Customer Information: This section displays information about the customer, including:

- **Metered Service**: The customer's metered service record
- **Contact**: Name and contact information for the customer who initiated the service call

Location Information: This section displays information about the customer's location, including:

- **Metered Service Point**: The customer's metered service point

Interaction Details: This section is used to capture details around customer interaction. The information in this section is used to create Customer Communications. See About Contact Communications for more information.

- **Channel**: The channel used for the communication, such as email or phone
- **Class**: The class of the communication, such as enrollment, unenrollment, trouble report, and so on
- **Category**: The category of the communication, such as written or verbal
- **Notes**: Notes associated with the communication

Program Service Call

The **Program Service Call** panel is used to specify details related to the service call. This includes the following:

Service Call Details: This section is used to select the service task type and other information for the service call, including

- **Related Service Call**: A related program service call. This is used if the current service call is somehow related to a previous service call. This is used often when selecting either "Escalate" or "Resolved" actions (below).
- **Service Call Type**: The service task type to use for the service call.
- **Questionnaire**: A questionnaire applicable to the service call. Available questionnaires are defined on the service task type.

Questions: This section displays questions from the selected questionnaire (if applicable). This section is only displayed if a questionnaire is selected in the **Service Call Details** section.

Action Details: This section is used to specify details regarding the service call, including:

- **Category**: The category of the service call. Options include "Emergency" and "Non-Emergency".
- **Action**: The action to be taken for the service call. Options include:
  - **Create a Work Order**: Creates a work order based on a selected **Template Work Order**. Available template work orders are defined on the service task type.
    - **Template Work Order**: The template work order for work order activities created for this service call.
    - **Crew Shift Type**: The crew shift type for the work order.

Copyright © 2026, Oracle and/or its affiliates.

        - **Crew**: The crew for the work order.
        - **Work Order Activities**: A list of work order activity types based on the selected **Template Work Order**.
        - **Appointment Date**: The date on which field work based on the work activity type is scheduled to be performed. Select an appointment slot from the list of available appointment slots for the selected **Appointment Date**. The list of available appointment slots is based on the selected Template Work Order, Crew Shift Type, and Crew.
    - **Escalate**: Escalates an existing service call. Used with a **Related Service Call**. This creates a To Do Entry based on the the **Escalation To Do Type** defined on the service task type. The To Do Entry can be assigned to one of the To Do Roles defined for the **Escalation To Do Type** from the **Assigned to** drop-down list.
        - **Call Back Required**: Indicates (Yes or No) if the customer must be called back.
        - **Keep Service Call Open**: Indicates (Yes or No) is the service call should be kept open due to the escalation. When set to "Yes", the service task is kept in the "Escalated" state, and can be then selected as a "Related Program Service Call" in a future service call process flow.
        - **Assigned to**: The To Do Role to which To Do entries related to the escalated service call can be assigned.
    - **Resolved**: Resolves an existing service call. Used with a **Related Program Service Call**. Service calls with this action are closed when finished.

Interaction Details: This section is used to capture details around customer interaction. See description above.

Summary

The **Summary** panel displays a summary of information related to the service call, including:

- **Program Service Call**: The Service Task used by the process and its current status.
- **Customer Information**: Contact and identification for the customer, including the customer's Metered Service record .
- **Location Information**: Location information for the service call, including the customer's Metered Service Point.
- **Interaction Details**: Details related to customer interaction. See description above.

Click **Finish** on the **Summary** panel to initiate the action selected on the **Program Service Call** panel. Click **Cancel Process** to cancel the process.

Manually Completing Work Activities Using Process Flow

Work activities in Customer Program Management can be manually completed using the Process Flow portal and functionality. This allows users to step through a series of panels on which they enter completion information about the work activity, including asset installation and removal and service histories (if any) for the asset and activity. This process creates a set of completion events which are used to update the system as needed to reflect the completion of the work activity. See Understanding Completion Events for more information about completion events. When used with work activities associated with enrollment and unenrollment, the enrollment or unenrollment service tasks associated with work activities completed using this process are transitioned to an appropriate status, facilitating progression and completion of enrollment and unenrollment processes.

Copyright © 2026, Oracle and/or its affiliates.

Work activity manual completion process flows are initiated from work activities. Clicking **Initiate Completion** on work activities open the process flow portal. Note that this option is available only if the work activity's activity type has either an 'Update Asset' or a 'Create Any Service History' Completion Event Type (or both) defined, and the activity is in the "In Progress", "In Progress External", "Active", "Sent", or "Finished" state.

When navigating between panels on the Process Flow portal, the **Next** and **Previous** buttons allow you to move to the next or previous panels in the process respectively. To save your progress and settings before moving on to the next panel, click **Save**. Click **Finish** on the **Summary** panel to mark the process flow as Closed. To cancel the process, click **Cancel Process**. This transitions the service task to the "Canceled" state. See Action Method in the *Administrative User Guide* for more information the process flow functionality,

The following sections describe the content on the different panels used in work activity manual completion process flows.

Field Work Details

The **Field Work Details** panel is used to specify completion information related to the asset and location such as installation and removal of assets, as well as recording details for customer-owned assets. This includes the following:

Install Controllable Device: Completion details for an installation of or more controllable devices at the work activity's location.

- To add a controllable device for installation, enter an **Effective Date** for each device's installation and each device's **Serial Number** (and optionally **Manufacturer** and **Model** if the same serial number is used with different Manufacturers and Models). The **Comments** column indicates if the device is present in the system.

Remove Controllable Device: Completion details for an removal of or more controllable devices at the work activity's location.

- To add a controllable device for removal, enter an **Effective Date** for each device's removal and each device's **Serial Number** (and optionally **Manufacturer** and **Model** if the same serial number is used with different Manufacturers and Models). The **Comments** column indicates if the device is present in the system.

Customer Owned Asset: Completion details for recording relevant customer owned assets, such as HVAC components, solar panels, inverters, batteries, or other assets that were discovered by a crew at the work activity's location.

- To add a customer owned asset, enter an **Effective Date**, select the **Manufacturer**, select the model **Model**, and enter the **Serial Number** for each customer-owned asset. The **Comments** column indicates if the asset exists in the system.

A completion event will be created to update the system to reflect installation or removal of the controllable devices specified, and to create new controllable devices based on customer owned assets.

Click **Next** to move to the **Activity Service History** panel.

Activity Service History

The**Activity Service History** panel is used for entering completion information related to service history related to the work activity (if applicable). This includes the following:

Activity Service History: Details for service history information related to the work activity, including the **Service History Type** and corresponding details (such as Effective Date/Time, questions, and so on).

Copyright © 2026, Oracle and/or its affiliates.

- To enter activity service history completion information, select the **Service History Type**, enter the **Effective Date/Time**, and then populate other data fields as appropriate. Required service history types (those that must be completed) are marked with an asterisk (*).

> **Note**
>
> This section is visible only when the work activity has a **Planned Service History** of category "Inspection" and subclass "Questionnaire" and whose activity type has a 'Create Any Service History' **Completion Event Type** defined. Otherwise, this panel displays a "There is no Activity Service History to Complete" message.

For each service history updated on this panel, an appropriate completion event will be created to update the service history in the system.

Click **Next** to move to the **Asset Service History** panel. Note that you must provide details for all required service histories for the activity before moving to the next panel.

Asset Service History

The**Asset Service History**panel is used for entering completion information related to service history. This includes the following:

Asset Service History: Details for service history information related to the asset, including the **Service History Type** and corresponding details (such as Effective Date/Time, questions, and so on).

- To enter asset service history completion information, select the **Asset**, select the **Service History Type**, enter the **Effective Date/Time**, and then populate other data fields as appropriate.

> **Note**
>
> This section is visible only when the assets from the work activity **Asset List** and/or assets defined on the **Field Work Details** panel (in the **Install Controllable Device** and **Remove Controllable Device** sections) have associated service histories. Otherwise, this panel displays a "There is no Asset Service History to Complete" message.

For each service history updated on this panel, an appropriate completion event will be created to update the service history in the system.

Click **Next** to move to the **Summary** panel. Note that you must provide details for all required service histories for the activity before moving to the next panel.

Summary

The **Summary** panel displays completion events related to the work activity, including those defined on the **Field Work Details**, **Activity Service History**, and **Asset Service History** panels.

When initially opened, the Summary panel is empty. Click **Finish** on the **Summary** panel to create completion events based on details defined on the previous panels.

Completion events are created as follows:

Copyright © 2026, Oracle and/or its affiliates.

- A single completion even is created for the assets/devices defined on the **Field Work Details** panel.
- A completion event is created for each service history defined on the **Activity Service History** and **Asset Service History** panels.

These completion events can be viewed on the work activity that created the process flow.

Click **Close Out** on the work activity to execute the completion events and update the system as needed to reflect the completion of the work activity.

Using the Customer Program Summary Portal

The **Customer Program Summary** portal is used to view customer information, such as currently enrolled programs, programs the customer is eligible to enroll in, and program activity, such as specific events in which the customer participated.

To access the **Customer Program Summary** portal, select **Main**, select **Customer Program Management**, and select **Customer Program Summary**. Once you've selected this menu option, you're immediately brought to the initial tab of the portal.

The **Customer Program Summary** portal displays information about the customer on which you are currently working. This is customer currently “in context” or the customer most recently viewed on any of the following portals: **Controllable Asset**, **Device Location**, **Metered Service**, **Metered Service Point**, **Program Subscription**, or **Program Contact**, The **Customer Program Management Full Context Zone** dashboard zone also displays information about the customer currently in context.

The **Customer Program Summary** portal includes the following zones:

- **Current Enrollment**: This zone displays current program enrollment for the customer (for a single location). Each program enrollment is displayed in a separate **Information** box. Click **Unenroll** to unenroll the customer from a program. Click **Service Call** to initiate a service call for a program. Click **Opt Out** to navigate to a list of pending program events for the customer where you can select one or more events from which the customer wishes to opt out. Click **Schedule Unavailability** to navigate to a portal where you can define an unavailability period for a controllable device..
- **Customer Insights**: This zone displays any currently in progress service tasks and work activites related to enrollment or unenrollment. Options on this zone include:
  - **Go To**: Used to navigate to the Process Flow for the enrollment/unenrollment or to the work activity.
  - **Cancel**: Opens the **Cancel "In Progress" Service Task** window that allows the user to cancel the enrollment/unenrollment service task. This also creates a contact communication record as part of this.
  - **Reschedule**: Opens the **Reschedule "In Progress" Service Task** window. Allows the user to specify a new date for the start date of the program subscription. This also creates a contact communication record as part of this.
- **Eligible Programs**: This zone displays programs for which the customer is eligible based on currently installed devices (at a single metered service point). Click **Enroll** to enroll the customer in a program.
- **Customer Program Activity**: This zone displays the customer’s activity in the programs in which they are enrolled. This includes a graphic view of the percentage of successful vs. non-successful event participation (referred to “non-responsive device” or “NRD”), as well

Copyright © 2026, Oracle and/or its affiliates.

as a list of the events in which the customer participated, and other activity such as device installation.

Using the Program Dashboard

This portal is used to display the **Program Dashboard**.

To access this portal select **Main**, select **Dashboards**, and select **Program Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The **Program Subscription Trends** zone displays enrollments, unenrollments by period (week or month) as well as active subscription for that period. The x-axis displays the end date of the monthly or weekly period. The left y-axis shows the number of enrollments and unenrollments. The right y-axis shows the number of active program subscriptions. You can filter the data displayed using the following controls:

- **Time Scale**: Allows specifying the time scale of the graph is weekly or monthly.
- **Period View**: Allows specifying the range of data displayed, such as “Last X Weeks” or “Last Month” and so on.
- **Program Category**: Allows filtering the data based on a selected program category (such as “Device Control”)
- **Program**: Allows filtering the data based on a selected program.
- **Division**: Allows filtering the data based on a selected division.

Click **Configure** to navigate to the Program Dashboard Configuration master configuration. See Configuring the Program Dashboard in the *Administrative User Guide* for information regarding configuration of the dashboard.

Click **Capture** (next to **Snapshot Date/Time** in the lower right corner of the zone) to navigate to the **Statistics Control** portal where you can view a list of snapshots for the statistics control, and create a new snapshot if needed.

Using Unified Search with Customer Program Management

The Unified Search feature allows the use of the toolbar search field to search for business entities as well as to search for menu items. See **Search** for more information about the general use of the unified search field.

In Customer Program Management, the Unified Search field allows searching by a program subscription's contact (Name), address (Address), customer information system account ID (External Account ID), contact information such as a phone number or email address (Contact Information), or a controllable device's serial number (Serial Number).

You can use the following filters to indicate your specific search criteria:

Filter	Filter Rule
Name	Minimum of 5 characters excluding whitespaces
Address	One or more numeric characters followed by space and an alphanumeric character.
External Account ID	Up to 10 characters followed

Copyright © 2026, Oracle and/or its affiliates.

Filter	Filter Rule
Contact Info	Phone Number: Plus character and 1-3 numeric characters (country code, optional) followed by a space, period, dash or open parentheses, 3 numeric characters (area code), followed by a space, period, dash or close parentheses, followed by 3 numeric characters (exchange code), followed by a space, period or dash, followed by 4 numeric characters (subscriber number).] Email Address: Non-whitespace characters followed by an @ sign, non-whitespace characters, a period, non-whitespace characters
Serial Number	Minimum of 5 characters excluding whitespaces.

For example, to search for a program subscription located at 1401 Flower Street, you could enter “1401 Flower”.

To search explicitly using a specific filter, precede the filter value with one of the following predefined hints:

Filter	Hint	Example
Name	n:	n:Smith, J
Address	ad:	ad:1401 Flower
External Account ID	ac:	ac:1234567890
Contact Info	c:	c:555-2287 c:mbrazil@oracle.com
Serial Number	sn:	sn:678912

Examples:

- To search for a metered service point located at 1401 Flower Street, you could enter "ad:1401 Flower" (hint) or “1401 Flower” (no hint)
- To search for a contact named John Smith, you could enter "n:Smith, J" (hint) or “Smith,J” (no hint)
- To search for an external (CIS) account with an ID of ACCT01 you could enter "ac:ACCT01" (hint) or "ACCT01" (no hint)
- To search for a contact whose phone number is 555-2287 , you could enter "c:555-2287" (hint) or “555–2287” (no hint)
- To search for a controllable device whose serial number is 6789, you could enter "sn:6789" (hint) or “6789” (no hint)

Clicking the **Advanced Search** link opens the **Metered Service Query** portal and the **Metered Service Search** zone.

For information about configuring the Unified Search for your implementation, refer to **Understanding Unified Search** in the *Administrative User Guide*.

Search Results

Each result in the search result list may include any of the following:

- Name (highlighted when using the Name filter) or “No Person Information”
- Address (highlighted when using the Address filter) or “No Premise Information”
- External Account ID (highlighted when using the Account ID filter) or “No Account Information”

Copyright © 2026, Oracle and/or its affiliates.

- Contact Info (highlighted when using the Contact Information filter) or “No Contact Information”
- Controllable Device Info (highlighted when using the Serial Number filter)
- Filter Name and Search Criteria (always displayed)

To select an item from the search results, click the result. The **Customer Program Summary** portal opens displaying information for the selected result.

Using Service Order Management for Customer Program-Related Field Activities

This section provides information about using Service Order Management functionality for customer program-related field activities.

At a high-level, this process works like this:

1. A Program Enrollment or Program Unenrollment service task creates a service order field activity, based on configuration of the corresponding service task type.
2. The service order field activity creates an outbound communication to send the required information to the field work system.
3. When the field work is completed, the field work system sends a response and the system creates an inbound communication.
4. The inbound communications create completion events as defined on the field task type configured on the service task type.
5. After receiving the inbound communication, the service order field activity executes completion events, which update device location information as appropriate.

Refer to Service Order Overview and Understanding Service Order Management for more information about the service order management functionality.

> **Note**
>
> Customer Program Management does NOT use Service Order Activities (also referred to Orchestration Activities such as Enable Service, Disable Service and so on). Instead service order field activities are created by customer program service tasks.

See **Defining Customer Program Service Order Management Options** for information about configuring the system to support service order management.

Customer Program Service Order Field Activities

Service order field activities are specific types of tasks performed by field personnel. When used with Customer Program Management, examples include installing/removing controllable devices or performing a home energy assessment. Service order field activities are created when a service task includes required field work (based on the configuration of the service task type).

Refer to About Service Order Field Activities and Maintaining Activities for more information.

In addition to other data, customer program field activities are defined by the following:

- **Program Subscription:** The program subscription for which the activity was created.

Copyright © 2026, Oracle and/or its affiliates.

System Data

Customer program service order field activities are based on the following system data:

Data Type	Description
Maintenance Object	Activity (D1-ACTIVITY)
Business Object(s)	Field Activity (D1-FieldActivity)

Customer Program Outbound Communications

When communicating with field work systems, service order field activities create "two-way" outbound communications to send the required information to the field work system. Outbound communications create outbound messages which are routed to the field work system.

Refer to About Communication Out and Maintaining Communications for more information.

In addition to other data, customer program outbound communications are defined by the following:

- **Notes**: Any notes obtained by the customer service representative during the program enrollment, unenrollment or program service call
- **Transaction Id**: The Service Task ID for the program enrollment, unenrollment or program service call
- **Eligible Related Programs**: A list related programs for which the customer is eligible
  - **Program ID**: The ID of the related program
  - **Information:** A description of the related program
- **Program Location**: The device location associated with the field activity
- **Device Details**: A list of devices installed at the service point
  - **Equipment ID**: An external ID for the device
  - **Serial Number**: The device's serial number
  - **NIC Identifier**: The device's network identification number
  - **Manufacturer**: The device's manufacturer
  - **Model**: The device's model number
  - **Installation Date/Time**: The date and time at which the device was installed
  - **Device Type**: The device's device type
  - **Service History Information**: Valid service history types including the failure and down time information

    Comments: Additional information about the equipment's specific location

System Data

Customer program outbound communications are based on the following system data:

Data Type	Description
Maintenance Object	Communication Out (D1-COMMOUT)
Business Object(s)	Field Activity Outbound Communication (D1-FieldActivityOBComm)

Copyright © 2026, Oracle and/or its affiliates.

Customer Program Inbound Communications

When the field work is completed, the field work system sends a response to the outbound communication and the system creates an inbound communication.

Refer to About Communication In and Maintaining Communications for more information.

In addition to other data, customer program inbound communications are defined by the following:

- **Program Location**: The device location associated with the field activity
- **Device Details**: A list of devices installed at the service point
  - **Equipment ID**: An external ID for the device
  - **Serial Number**: The device's serial number
  - **NIC Identifier**: The device's network identification number
  - **Manufacturer**: The device's manufacturer
  - **Model**: The device's model number
  - **Installation Date/Time**: The date and time at which the device was installed
  - **Device Type**: The device's device type
  - **Action**: The action performed based on the field activity, such as "Attach" or "Detach".
  - **Comments**: Additional information about the equipment's specific location

System Data

Customer program inbound communications are based on the following system data:

Data Type	Description
Maintenance Object	Communication In (D1-COMMIN)
Business Object(s)	Field Activity Inbound Communication (D1-FieldActivityIBComm)

Customer Program Completion Events

Completion events are used to implement the outcome of an activity. They are created by the inbound communication and are executed by the service order field activity.

Refer to About Completion Events for more information.

System Data

Customer program completion events are based on the following system data:

Data Type	Description
Maintenance Object	Completion Event (D1-CEVT)
Business Object(s)	Update Controllable Device Location - FW Completion (X1-UpdateCDL)

Other Related Functionality

Copyright © 2026, Oracle and/or its affiliates.

This section includes topics covering other functionality used by Customer Program Management.

- Approvals
- About Specifications
- Maintaining Specifications

Approvals

- About Approvals
- Maintaining Approval Requests
- About Change Requests
- Maintaining Change Requests

About Approvals

Approvals simplify the routing of work orders, purchase requests, timesheets, and other documents that require approval.

Approval requests use approval profiles to define the requirements for the approval:

- Who receives the approval
- Method for notifying the approver
- Which work calendar to use

When the approval request is submitted, the system sends a notification to all approvers defined on the approval profile according to the **To Do Roles** and the **Sequence** defined on the profile. Notifications can be sent using email, To Do List, or both, depending on the system notification settings.

Approvers can use the **Approval Dashboard** to manage the approvals that require their attention.

This section describes how various processes work together to manage approvals throughout the system.

Refer to Maintaining Approval Requests in the **Business User Guide** for information on maintaining approval requests.

Refer to Understanding Approval Processing in the **Administrative User Guide** for more information.

- Approval Profiles
- Approval Routing
- Outstanding Approvals at a Glance
- Approval Dashboard

Approval Profiles

Approval profiles are used to define the approval hierarchy within your organization and controls which approvers are required to approve the record.

Approval Profiles based on **To Do Roles** define:

Copyright © 2026, Oracle and/or its affiliates.

- the approvers level of authority to approve a document
- the types of documents the role can approve
- the working calendar to use when determining approval time lines and when to send To Dos
- dollar amount maximums that the role is authorized to approve

An approval profile can have one or many **To Do Roles** and it is possible for an approver to be assigned to more than one To Do Role in the same approval profile.

Depending on the notification setting designated on the approval profile, the approver can receive an email, an alert, or both in their **To Do List**.

> **Note**
>
> Individual approvers can choose to modify their notification settings for approvals. Navigate to the **User** portal and access the **Characteristic** tab to set the **Approval Profile E-Mail Override** setting to disable email notifications for approval requests. When this is set, the approver does not receive any emails and must rely on the approval dashboard or To Dos.

Approval Routing

The approval request, once submitted, is routed for approval based on the roles of the approvers and their approval decisions. For example:

- **One Standard Role**: If the approval profile consists of one To Do Role with mandatory or standard approval type, the users with that role receive the approval notification and any one of them can approve the request as long as the To Do Role has authority for the needed dollar amount.
- **Multiple Roles/All Standard**: If the approval profile consists of multiple To Do Roles and all are standard, any To Do Role can approve the request as long as they have the authority for the needed dollar limit.
- **Mix of Mandatory and Standard Roles**: If the approval profile consists of multiple To Do Roles with a mix of mandatory and standard, the users with mandatory must approve the request even if the standard approvers have authority based on the dollar limit or other factors. In other words, the mandatory approval role must take an action by agreeing or rejecting the request.
- **Request Time-out**: If approvers do not respond to the request within the time allotted by the time-out threshold, the system sends another notification to the approvers. If action is still not taken, the request moves to the next approval role, as applicable and the next role receives notification.
- **One Approver with Multiple Roles**: If one approver has multiple To Do Role s on one approval profile for the request, the system prompts that person to indicate which role they are approving. For example, approver Jon is in both the supervisor and manager To Do Roles. When an approval request comes in and Jon approves the request, a selection box opens to allow Jon to choose in which **To Do Role** the request is being approved . If Jon selects multiple approval requests to approve at once in the **Approval Dashboard** Jon may receive a warning to specify an **Acting To Do Role** before selecting an action. This means you must pick a role from the **Acting To Do Role** drop down and click **Search** to let the system know which role you are using to approve the requests.

Copyright © 2026, Oracle and/or its affiliates.

- **Insufficient Approval Amount**: If an approval request comes in for a record with a higher dollar amount than anyone on the approval path is authorized to approve, the requestor is notified that no one on the approval profile has sufficient authority. The requestor must choose a different approval profile and resubmit.
- **One Standard Role**: If an approver rejects the approval request, the request terminates and additional approvers on the approval path are not notified. The requestor is notified that the request has been rejected. Any approver in the route can reject and terminate the approval process regardless of their approval authority or limit amount. However, this does not apply for “notification only” approvers.
- **Not Authorized to Approve**: Some roles might be set as “not authorized to approve” but they still appear on the approval path. These approvers receive notification of the request and can approve or reject, but their decision is simply recorded for the next approver on the approval path to be aware of their decision. This setting is only applicable for an approval profile that is for no amount.
  If this type of approver rejects the request, the same processing applies as described for “one standard role”.

Outstanding Approvals at a Glance

The **Approval Dashboard** provides approvers with a quick glance, by document type, of the status of approvals that are currently due or are now past due. Approvers can then take appropriate action to reject or approve the requesting document.

Also, the system includes an **Approval Request Action** dashboard zone in the side dashboard which highlights any new approval requests that require the logged in user’s attention. The user can approve or reject the request directly from this dashboard, if desired.

Approval Dashboard

The approval dashboard provides a summary of approval requests that have been sent to the broad list of approvers with a breakdown of the approval profiles. This overview provides approvers with a simple summary of the records requiring their attention.

The summary list includes the counts of approval requests that have been sent out to any approvers that are not **Notification Only** and shows the current approvers and the mandatory future approvers. The list is broken down by document type and approval profile.

It is possible to filter the list by current or future approvers using the **Queue** field. The filter is used to distinguish current and future approval requests which may require the attention of the approver. For example, if an approver is about to go on vacation, they may want to take action on an approval request early. In this case the list can be filtered by “future” approval requests where this approver is not necessarily next in the approval route. This is a way for the approver to obtain a “preview” of requests that are coming from the route.

Broadcasting an individual line opens a zone where the approver can check off records and choose a button to make their approval decision (approve or reject). The approver can also click the link to open the approval request itself, and to open the individual record.

Note that the dashboard contents are specific to the logged in user, so one person may see a different list than another depending upon their approval authority.

Refer to About Approvals for more information.

Copyright © 2026, Oracle and/or its affiliates.

Work Order Approval Summarization and Work Request Approval Summarization

Work Order Approval Summarization and Work Request Approval Summarization use generative Artificial Intelligence (GenAI) models to compile and provide a summary of the approval. The specific information provided for each is as follows:

- **Work Request Approval Summarization**: A summary of the approval request and work request to help users understand the work to be performed. In addition, related possible-duplicate information and, when available, approval history information is displayed with the summary.
- **Work Order Approval Summarization** A summary of the approval request, work order, and activity information to help users understand the work to be performed. In addition, access to individual activity records requiring approval and, when available, approval history information is displayed with the summary.

Click the icon in **Summarize** column of the **Approval Detail List** zone to view the **Approval Summary**, which provides a summary of the requesting document and approval request, as well as some key observations.

See Enabling the Work Order and Work Request Approval Summarization Features for more information about enabling this feature for work orders and work requests.

> **Note**
>
> - Work Order Approval Summarization and Work Request Approval Summarization are available under Limited and Controlled Availability in approved regions.
> - Work Order Approval Summarization and Work Request Approval Summarization rely on generative AI models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Work Order Approval Summarization or Work Request Approval Summarization.
> - Work Order Approval Summarization and Work Request Approval summaries are prepared by generative AI. Carefully review and verify them against trusted sources, as they may include out-of-date, inaccurate, or incomplete information.

Maintaining Approval Requests

This portal is used to view and maintain approval requests.

Refer to About Approvals for more information.

You can access the portal from **Menu**, select **Home Page**, then select the **Approvals** tab. Broadcast the **Document Type** and then select the **Approval Request ID** hyperlink from within the **Approval Detail List** zone.

The **Approval Request** zone on the portal's **Main** tab page provides information about the approval request, including approvers, approval amounts, approval profile, etc.

About Change Requests

Change requests simplify the process associated with managing changes respective to assets, asset locations, vendors, vendor locations, stock items, stock item details, and any other records as appropriate.

Copyright © 2026, Oracle and/or its affiliates.

The **Change Request** portal controls the full cycle of the change request, from request creation, review, approval, application, and closure. The portal includes an **Original** zone, which holds all the original data elements, and a **Proposed** zone, which holds all the changed data. The updated data elements are saved as temporary until the change is approved.

When the change request is submitted for approval, the system sends a notification to all approvers defined on the approval profile according to the **To Do Roles** and the **Sequence** defined on the profile. Notifications can be sent using email, To Do List, or both, depending on the system notification settings.

Approvers can use the **Approval Dashboard** to manage the change request approvals that require their attention.

Refer to Maintaining Change Requests in the **Business User Guide** for information on maintaining change requests.

Refer to Understanding Change Request Types in the **Administrative User Guide** for more information.

- Change Request Approval Profiles
- Change Request Approval Routing

Change Request Approval Profiles

Change request approval profiles are used to define the approvers who are required to approve a change request record.

Approval Profiles based on **To Do Roles** define:

- the approvers level of authority to approve a record
- the types of documents the role can approve
- the working calendar to use when determining approval time lines and when to send To Dos

An approval profile can have one or many **To Do Roles** and it is possible for an approver to be assigned to more than one **To Do Role** in the same approval profile. Valid types of approval profiles include the following:

- Non-Amount Based
- No Approval Required

Depending on the notification setting designated on the approval profile, the approver can receive an email, an alert, or both in their **To Do List**.

> **Note**
>
> Individual approvers can choose to modify their notification settings for approvals. Navigate to the **User** portal and access the **Characteristic** tab to set the **Approval Profile E-Mail Override** setting to disable email notifications for approval requests. When this is set, the approver does not receive any emails and must rely on the **Approval Dashboard** or **To Dos**.

Change Request Approval Routing

The change request approval request, once submitted, is routed for approval based on the roles of the approvers and their approval decisions. For example:

Copyright © 2026, Oracle and/or its affiliates.

- If the approval profile consists of one **To Do Role** with mandatory or standard approval type, the users with that role receive the approval notification and any one of them can approve or reject the request.
- If the approval profile consists of multiple **To Do Roles** and all are standard, any valid **To Do Role** can approve or reject the request.

The approver is able to review the proposed changes and approve or reject the change request. Rejected requests can be resubmitted for approval once the issue is resolved. Once approved, a user can click the **Apply Changes** button to apply the change. This action completes the change request and updates the appropriate record.

Maintaining Change Requests

The **Change Request** portal is used to display and maintain a change request record.

Refer to About Change Requests for more information.

You can access the portal from **Menu**, select **Resource Management**, then select **Change Request**.

- If **Add** is selected:
  - Only a “Generic” change request can be created.

> **Note**
>
> Creation of any other type of change request must be on the change request associated record. Once on the associated record, user permissions drive who is able to create the change request. Users having full permissions can select **Edit** to make their changes or select the **Create Change Request** button directly from the **Dashboard**. Users having limited permissions can only create a change request by selecting the **Create Change Request** button directly from the **Dashboard**. Once the change request is created, users having full permissions can view the change request by going directly to the associated record and clicking the change request hyperlink from the zone on the **Dashboard**.

- If **Search** is selected:
  - User is brought to a query portal with options for searching for a specific change request. Once a change request has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Change Request**: This zone displays information about the change request, including the description of the change and any warning information.
- **Original**: This zone displays the original details of the request.
- **Proposed**: This zone initially displays original data. Selection of an **Edit** hyperlink enables the user to update the data to proposed values. This zone highlights the updated **Field** names and also the **List Headers** on the associated record if the content of the list has changed.
- **Attachments**: This zone enables users to manage all types of attachments for the change request record.

The following zones may appear as part of the portal's **Log** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Change Request Log**: This zone displays log details for the change request.
- **Approval Request History**: This zone displays the approval histories for the request.

About Specifications

Specifications describe design details and functional performance criteria for a set of assets. This can include manufacturer, model, and other information specified for the assets.

Specifications may also contain written procedures, careful handling instructions for hazardous or delicate materials, or any other set of attributes needed to provide information about a set of assets.

The base package specifications provide a general starting point for your specification set-up including the following:

- General Maintenance Specification
- Operational Maintenance Specification
- Operational Device Specification
- Fleet Specification
- General Specification
- IT Specification
- Infrastructure Specification

In **Oracle Utilities Operational Device Management**

- Analog Meter Specification
- Communication Component Specification
- Communication Relay Specification
- Firmware Specification
- Smart Meter Specification

Implementers may want to include additional specification details. An example of how this can be accomplished using the fleet or vehicle specification would be to extend the data area *W1-VehicleDetail*. Specification detail values may also be extended by adding values to the following look-up fields:

- **MODEL_YEAR_FLG** - model year
- **VEHICLE_COLOR_FLG** - vehicle color
- **DRIVE_TYPE_FLG** - drive type
- **FUEL_TYPE_FLG** - fuel type

Some of the main uses for specifications are as follows:

- **Configuration Specifications** enable definition of different attributes on each device type.
- **Engineering Specifications** enable creation of Manufacturer, Bill of Material (BOM), and Model details for specific types of assets.
- **Preventive Maintenance Specifications** provide for the building and sharing of maintenance strategies between an asset or group of assets.

Specifications can be linked to stock items. This linking associates the stock items to the tracked assets that are also associated with the specification which allows the stock item to be

Copyright © 2026, Oracle and/or its affiliates.

used to track the asset. Only specifications with asset types configured with an ‘inventory controlled storeroom’ node type can be added.

Please note the following with regard to specifications:

- Specifications often include an attachment which supports the specification definition or provides the actual specification from the manufacturer.
- A specification may also include default *in service* and *out of service* maintenance schedules for assets or components linked to the specification. When the specification is associated with the asset or component, the related maintenance schedule becomes in effect for the asset or component.
- Firmware

Firmware

Firmware is software that is embedded in a piece of hardware and enables the hardware to work as the manufacturer intended. For example, commands to enable relay devices to automatically open and close.

Refer to About ODM Firmware for more information.

Maintaining Specifications

This portal is used to display and maintain specification records.

Refer to About Assets for more information.

You can access the portal from **Menu**, select **Asset Management**, then select **Specification**. You are brought to a query portal with options for searching for a specific specification. Once the specification has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Specification**: This zone provides details about the specification.
- **Specification Attributes**: This zone displays a list of attributes for this specification. The asset type defines valid attributes for specifications of this type. Press the **Edit** icon to add or change the attribute's value. Press the **Clear** icon to clear its value. Click the **Edit** link in the zone header to edit multiple attributes at one time.
- **Warranty Term**: This zone lists the warranty terms associated with the specification.
- **Specification Assets**: This zone provides information about the asset associated with the specification.
- **Attachments**: This zone enables users to manage all types of attachments for the specification record.

Oracle Utilities Rate Cloud Service

- Overview
- Setting Up Rates with Oracle Utilities Rate Cloud Service

Copyright © 2026, Oracle and/or its affiliates.

- Using Rate Check with Oracle Utilities Rate Cloud Service

Overview

Oracle Utilities Rate Cloud Service provides access to the rate calculation functionality of Oracle Utilities Customer Cloud Service. This includes the ability to configure and test rate calculations via the user interface, as well as the ability to trigger the rate calculation process from an external system.

This section describes the functionality provided with Oracle Utilities Rate Cloud Service.

Note: Oracle Utilities Rate Cloud Service supports rate calculation for traditional scalar or volume-based billing quantities only. It does not include capabilities for generating billing determinants from interval data input.

Setting Up Rates with Oracle Utilities Rate Cloud Service

Oracle Utilities Rate Cloud Service provides access to pages, portals, and zones used to create and maintain rates and rate configuration data, including:

- Bill Factors
- Bill Factor Values
- Calculation Groups
- Calculation Line Category Types
- Calculation Rules
- Rate Schedules

Refer to the Rates chapter of the *Business User Guide* for detailed information about setting up the above types of configuration data. Note: Some fields and options used with the above types of configuration data are not applicable when using Oracle Utilities Rate Cloud Service when compared to using Oracle Utilities Customer Cloud Service.

Using Rate Check with Oracle Utilities Rate Cloud Service

In addition to rate configuration functionality, Oracle Utilities Rate Cloud Service includes access to the Rate Check feature, which allows users to test their rates.

When using Rate Check, service quantities and characteristic types/values must be provided by the user and entered via the user interface. Oracle Utilities Rate Cloud Service does not support the use of the Rate Check feature based on service agreements, bill segments, or service points.

Refer to the Rate Check section in the **Rates** chapter of the *Business User Guide* for more information about using the Rate Check feature.

Oracle Utilities Billing Cloud Service

- Overview

Copyright © 2026, Oracle and/or its affiliates.

- Billing Cloud Service Functional Areas

Overview

Oracle Utilities Billing Cloud Service provides access to the billing functionality of Oracle Utilities Customer to Meter, which is the foundational component of Oracle Utilities Customer Cloud Service. This includes the ability to manage customer information, load validated measurement data, calculate billing determinants, configure and test rate calculations, calculate billing charges, aggregate billing related data.

Oracle Utilities Billing Cloud Service is intended for customers who

- May not require a full customer information system implementation
- Customers whose legacy customer information system does not support specific billing capabilities that they may now require such as net-metered billing, billing for electric vehicle charging events, advanced billing for specific customer segments (commercial or industrial), and so on.

This section describes the functionality provided with Oracle Utilities Billing Cloud Service.

Billing Cloud Service Functional Areas

This page outlines the functional areas available with Oracle Utilities Billing Cloud Service. The menus provided with Oracle Utilities Billing Cloud Service display only those menu items that correspond to capabilities that are available with the cloud service.

> **Note**
>
> Not all fields and field values available in Oracle Utilities Customer to Meter associated with these menu items are used with Oracle Utilities Billing Cloud Service.

The functional areas listed below are organized based on the main **Menu**.

- 360 Search
- Communication
- Control Central
- Customer Information
- Dashboards
- Data Synchronization
- Device
- Device Installation
- Financial
- Financial Query
- Initial Measurement Data
- Initial Measurement Snapshot
- Market Transaction Management

Copyright © 2026, Oracle and/or its affiliates.

- Process Flow
- Rates
- Service Task
- To Do
- Total and Trends
- Umbrella Agreement Management
- Usage
- My Preferences
- Tools
- Integration

Refer to specific chapter and sections in the *Business User Guide* for more information about working with these above listed functional areas.

Settlement

- Understanding Market Settlement
- Viewing and Managing Settlement Processes
- Settlement Calculations
- Settlement Item Market Participants
- Market Allocations

Understanding Market Settlement

Market settlement is the process of reconciling the difference between a supplier's energy purchases and its customer's usage. Suppliers purchase energy via bilateral contracts and day-ahead market transactions. Any differences between these purchases and supplier customer usage must be determined by the market and charged or credited at real time prices. The supplier's customer usage is not directly determined by the market, but by numerous other market participants that track customer supplier enrollments, aggregate customer usage to the supplier, and submit supplier's usage schedule to the market. This process is called load settlement, and is the primary purpose of Oracle Utilities Market Settlements Management.

The market participant responsible for load settlement can differ per market. For example, load settlement for a large portion of the Texas market is performed by an Independent System Operator (ISO) known as the Electric Reliability Council of Texas (ERCOT), whereas in the PJM market (encompassing Pennsylvania, New Jersey, and Maryland) load settlement is the responsibility of the Electric Distribution Company (EDC). While there are significant differences between markets in how load settlement is preformed, the high-level process steps are basically the same.

Load settlement starts with the bottom up aggregation of actual load or estimated load for individual customer accounts or service points. Distribution and transmission losses are added in to calculate an unreconciled load amount. The market determines the total generation

Copyright © 2026, Oracle and/or its affiliates.

supplied to a zone and the difference with the unreconciled load is referred to as Unaccounted for Energy (UFE). The UFE is then allocated to suppliers on a load ratio share basis.

The load settlement process is typically performed multiple times. The “preliminary” load settlement is done soon after the Operational Day and is often based on estimated load. The “final” load settlement is done around 60 days after the Operational Day, so that actual data has been received and customer bills issued.

- About Market Settlement Processing

About Market Settlement Processing

Market settlement is the process of deriving billing determinants to be used in supplier billing based on the results of a data aggregation processes.

The load settlement process also involves calculating imbalances. An imbalance is the difference between two load shapes representing different estimates of the same period, typically a scheduled load shape and an actual load shape. Imbalances are typically used when money has changed hands based on an estimate that has been improved or finalized by the settlement process. Because the second estimate has more accurate data, the allocations performed by the first process have to be modified.

Imbalance shapes can be further divided by applying bands and TOU components. Bands are error tolerances around zero that allow for differential pricing of the imbalance curve by how well or poorly the scheduled estimate matched the actual usage. Time of Use periods are often defined, allowing a higher penalty for faulty forecasting during higher-priced, on-peak periods. Whether an imbalance calculation is performed and how it is performed varies greatly from market to market.

The following sections provide details regarding some of the specific processing used in settlement processing.

- Importing Data
- Forecasting and Estimating Usage
- Aggregating Data
- Unaccounted for Energy (UFE) Calculations
- Allocating Load via Market Products, Awards, and Contracts

Importing Data

In addition to customer-level data such as meter readings or billed service quantities (which are used aggregation), settlement processing also involves other data that must be available to the application. For example:

- UFE calculations use both aggregated data and overall system, or zonal, load data
- Estimations and forecasting often use weather data
- Settlement billing requires pricing data

Data of these types can be imported into the system from external sources via a number of ways:

- For external sources that send data on a regular basis, data can be imported in a manner similar to how measurement data is received head end systems. See Processing Initial

Copyright © 2026, Oracle and/or its affiliates.

Measurement Data in this guide and Data Import - Message Driven Bean Configuration in the *Oracle Utilities Meter Solution Administrative User Guide* for more information.

- For external systems that send data upon request, data can be requested via measuring component data requests. Requests of this type can be either scheduled or performed manually. See Requesting Measurement Data in this guide for more information.
- For external systems that make data available for use but don't support sending the data directly, data can be imported manually in a comma separated values (CSV) format. See Uploading Measurement Data in this guide for more information.

Forecasting and Estimating Usage

Some settlement processes, such as initial settlement, often have to be performed before actual usage has been received or calculated for service points participating in the settlement market. In these cases, usage must be estimated, and in some cases, forecasted before aggregation and other settlement calculations can happen. A forecast is the term used when estimating usage data for future dates based on historical or profile data. Backcasting is the term used for estimating data for dates in the past. These can also be based on historical and/or profile data.

Two specific methods used for forecasting and backcasting include:

- **Profile Estimation**: This method uses profile data that represents typical usage for the service point to estimate usage. In some cases, profile data can also be adjusted based on scalar usage for the same service point.
- **Proxy Day Estimation**: This process compares data between measuring components to identify an historical day that most closely matches the date for which data is needed and uses data from that historical day to estimate usage

Aggregating Data

Aggregation involves summing real or assigned (profiled or deemed) interval data to a user-defined aggregation level for further processing. Aggregation may occur at a variety of levels for a variety of reasons, including reporting requirements and applying losses and other factors at an aggregated level up to the market participation level. Aggregation is typically used to build ground-up estimates of load for energy market participants for forecasting, scheduling, and settlement.

Aggregation is performed using the Dynamic Aggregation feature of the Oracle Utilities Meter Solution.

Aggregation Levels

Because aggregation involves summing up interval data to a specified level for processing, it's important to understand what is meant by the term "aggregation level" Put simply, an aggregation level is any point along the distribution chain to which entities below that point are summed. For instance, if you were going to sum up all the accounts that are related to a specific bus, that bus would be an aggregation level. Similarly, if you were going to aggregate all the energy used by all the accounts for a given Energy Service Provider (ESP), that ESP would be an aggregation level. In general, aggregation is usually performed at a level below you on the distribution chain from generators to end users.

The entities and/or market participants that serve as aggregation levels are specific to the particular area and market in which your utility is operating, but can include the following:

Copyright © 2026, Oracle and/or its affiliates.

Energy Service Providers (ESP)

Energy Service Providers (ESPs) sell power to customers. They may generate it and/or buy it from other sources. ESPs are also referred to as Retail Energy Providers (REPs) or simply suppliers.

Distribution Companies (DisCo)

A distribution company is responsible for the delivery of power to all customers in its service territory (Jurisdiction), regardless of who the customers have selected as their suppliers. The DisCo owns and operates the distribution lines that bring electricity into homes and businesses, and is responsible for metering, billing, and service calls. However, a DisCo may or may not be affiliated with a customer's power supplier. Distribution companies are also referred to as Operating Companies, Utility Distribution Companies (UDCs), or Local Distribution Companies (LDCs).

Schedule/Service Coordinators (SC)

Schedule Coordinators (or Service Coordinators) are responsible for creating schedules of energy usage that represent the estimated usage for a future time period (most often one day) for a given segment of the market. Schedules are most often based on results generated during the load profiling process, but can also be based on historical usage information and other data.

Independent System Operators (ISO)

An Independent System Operator (ISO) is a non-profit corporation that manages the regional power grid to coordinate the generation and transmission of energy. Its goal is to guarantee a safe and reliable power supply, and to ensure that all suppliers get nondiscriminatory access to the transmission system (while the ISO controls the power grid, commercial entities still own it). In the U.S.market, ISOs are regulated by the Federal Energy Regulatory Commission (FERC).

The various aggregation levels used in settlement processing are defined by Aggregation Groups, Measuring Component Sets, and Aggregation Measuring Components.

Aggregation Dimensions

In addition to aggregating data by aggregation levels, it is useful to be able to aggregate by other factors, such as customer class, voltage class, or loss class. These and other factors are referred as dimensions and are referenced on measuring component sets. Some common aggregation dimensions are described below. Note that these are provided as examples of common aggregation dimensions only.

Specific markets may require aggregation dimensions other than those listed here.

Customer Class

A customer class represents a 'type' of customer or account. Three common customer classes include residential, commercial, and industrial, but others are also prevalent. In addition, there may be multiple types of residential, commercial, or industrial classes, representing distinct types of accounts within each of those broad categories. For instance, residential customers might be divided into apartments, condominiums, and houses.

Voltage Class

A voltage class represents a group of customers or accounts based on the type of voltage used by the end users in that group. Voltage classes differ by market and region, but some common voltage classes include Transmission, Primary, and Secondary. Aggregating by voltage class is often performed for reporting purposes, to track usage of a given voltage type within the system.

Copyright © 2026, Oracle and/or its affiliates.

Loss Class

A loss class represents a group of customers or accounts based on the type of loss (transmission loss, distribution loss, etc.) associated with those accounts. Loss is often related to voltage levels, and so correlations between loss classes and voltage classes are common. Aggregation by loss class is used when calculating losses in the overall settlement process, and allows different loss calculation algorithms to be used with different loss classes, as appropriate.

Rate Class

A rate class represents a group of customers or accounts based on pricing. Rate classes can be based on a number of characteristics, including usage, territory, or even voltage type.

These and other dimensions are typically defined as attributes or characteristics of usage subscriptions, service points, devices, or measuring components.

- Applying Losses

Applying Losses

Markets typically operate using generation-level determinants, meaning that the individual metered entity data must be adjusted for system losses in order to compare the market participant load to the system load, market participant schedules, etc. Loss calculations can vary from simple to complex, ranging from an annual percentage by rate class or voltage level, to seasonal or time of use loss factors, to hourly dynamic loss factor calculations based on system conditions. Loss factors can be applied at the metered level or any aggregation level, as needed. In addition, losses can be applied to generators in markets where generation losses are calculated and used.

Applying losses to aggregated data is performed via value derivation algorithms. These are algorithms which apply a configured formula to aggregated measurement values. For instance, to apply a loss factor, an algorithm can retrieve effective-dated factor values for a loss factor and multiply each aggregated measurement value by the appropriate factor value. Value derivation algorithms can retrieve values from all types of factors, including multi-variable factors defined for settlement units. See Aggregation Calculation — Other Calculations for more information about how losses can be applied during data aggregation.

Unaccounted for Energy (UFE) Calculations

Unaccounted for Energy (UFE) is calculated by comparing the known system load to the sum of all market participant estimates from the estimation, aggregation, and loss adjustment processes. Because the market participant estimates will never perfectly match the system load, these estimates are fine-tuned by allocating a portion of the total UFE to each estimate. Allocation methods differ, but in general each market participant is allocated a portion of the system UFE for each hour based on the ratio of that particular estimate's load in that hour to the total of all estimated loads in that hour. This process forces the total of all estimates to match the system load. Some markets use different UFE allocation methods, including differential allocation of UFE to interval metered loads apart from profiled loads, and adjusting the profile methodologies to minimize UFE where possible.

UFE calculations are performed as part of the settlement calculation process, and can be performed using the Measuring Component Set Calculation Rule settlement calculation rule defined in a Bulk Settlement Calculation Group executed via the Measuring Component Set Calculation (D1-MCSCA) batch process.

Copyright © 2026, Oracle and/or its affiliates.

For smaller markets, UFE calculations can be performed using a chained sequence of Array Math and Vector and Service Quantity Math settlement calculation rules. At a basic level, this sequence looks like this:

1. Sum the market participant load from aggregation results (using Array Math)
2. Subtract the market participant load from the system load to derive the total UFE for the calculation period (using Vector and Service Quantity Math)
3. Allocate the total UFE to the various market participants as appropriate (using Array Math)

> **Note**
>
> The results of UFE calculations are also used in other settlement calculations. These three steps represent part of a larger set of calculations.

Allocating Load via Market Products, Awards, and Contracts

Distribution companies operating in deregulated markets are often designated as the Provider of Last Resort (POLR) for customers that either have not selected a supplier or have been dropped by a supplier. If allowed by legal and market rules, the distribution company can elect to outsource the responsibility of supplying POLR load to qualified wholesalers.

The POLR consumers will still, either directly or indirectly, receive bills for distribution services from and pay the distribution company, but the wholesaler takes responsibility for procuring and serving the POLR load, for which they receive payments from the distribution company.

A simple example of the POLR allocation process might involve the following steps:

1. Subtract all other suppliers from the zonal load to get the remaining POLR load.
2. Allow for the distribution company to separate out any owned generation.
3. Divide POLR load amongst awarded wholesale contracts.

Details of these transactions would be sent to the market operator by contract ID.

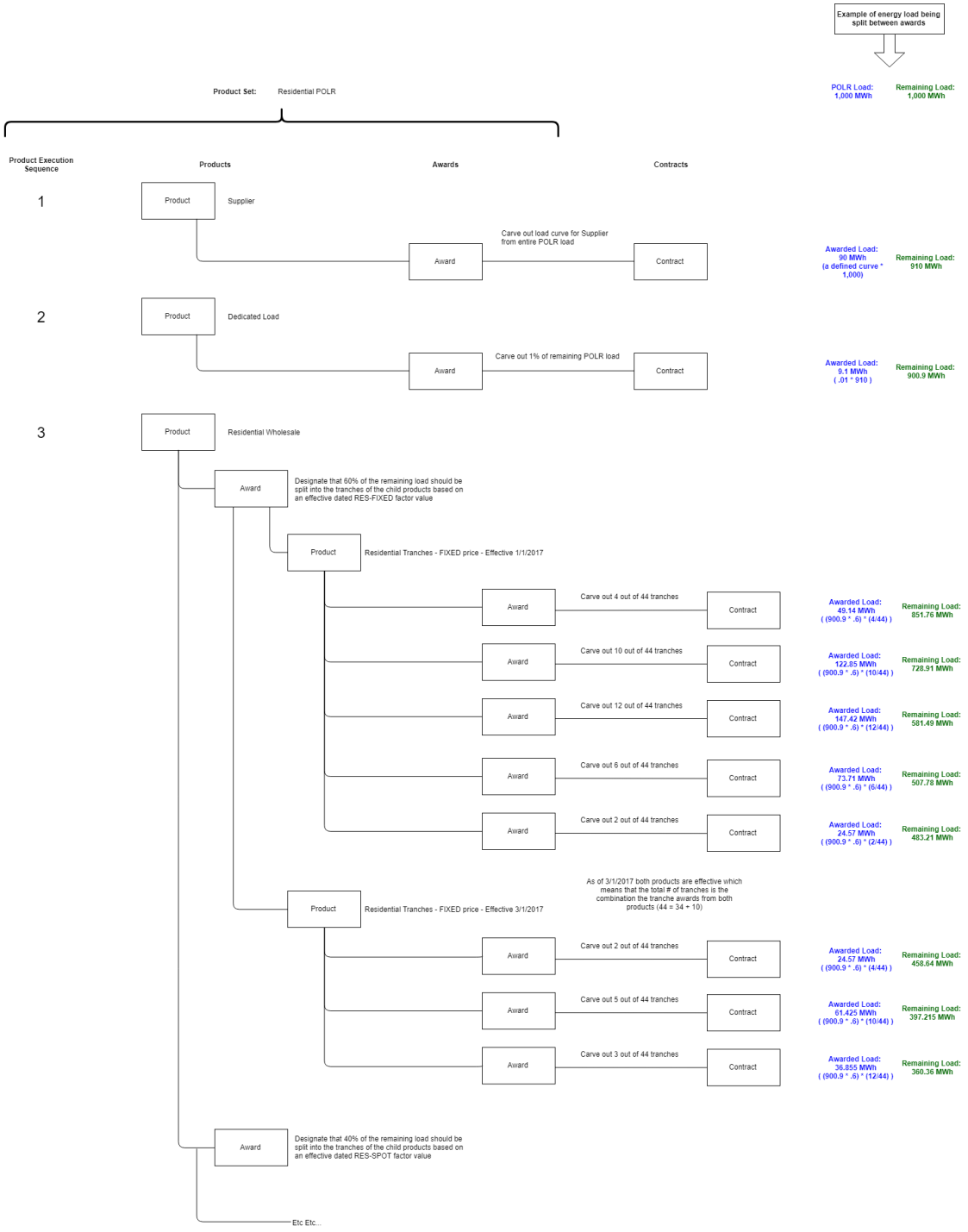
These awards of POLR load can be defined in the application by Market Products (which belong to a Market Product Set), Market Awards, and Market Contracts.

To elaborate on the example above, a market product set might the following set of market products:

- A product and award to carve out load from the POLR load for one or more suppliers. This would be a “Fixed Profile” award.
- A product and award to carve out a percentage of the remaining POLR load for owned generation. This would be a “Percentage” award.
- A product and awards that distribute the remaining POLR into various bundles of tranches (chunks of load) to wholesalers. These would be “Tranche” awards.

The following diagram illustrates the way in which this distribution would work.

Copyright © 2026, Oracle and/or its affiliates.



The Market Award Allocation settlement calculation rule is used to perform the allocation of load the designated market products and awards. This rule is part of a larger settlement calculation process, using settlement quantities (the POLR load) calculated by other rules (such as Array Math and Vector and Service Quantity Math). The Market Award Allocation rule allocates those quantities to the products within a specified Market Product Set, and saves the results for export to the appropriate market participants.

Viewing and Managing Settlement Processes

Copyright © 2026, Oracle and/or its affiliates.

You can use the Process Flows feature of the Oracle Utilities Application Framework to create, view, and manage settlement processes. See Process Flows in the *Oracle Utilities Application Framework Administrative User Guide* for more information.

Settlement Calculations

- About Settlement Calculations
- Settlement Subscriptions
- Settlement Transactions
- Settlement Items
- Settlement Item Quantities
- Settlement Billed Usage

About Settlement Calculations

Oracle Utilities Market Settlements Management can calculate and publish settlement results calculated from aggregated data to market participants on an ongoing basis. Settlement calculations derive a settlement transaction's service quantities using the aggregated measurement associated with one or more market participants.

Before settlement results can be calculated, you must first create a Settlement Subscription. The settlement subscription defines the external system to which the data should be sent as well as the settlement calculation groups that will be used for calculation. The settlement calculation rules defined for the settlement calculation groups are responsible for performing the specific calculations, validations, or estimations for settlement results.

Settlement service quantities are derived from aggregated measurement data the calculation period.

The results of all calculations performed by settlement subscriptions are stored as a Settlement Transaction.

The settlement calculation process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.

Settlement Calculation Groups

Settlement calculation groups are collections of settlement calculation rules that are used to calculate service quantities for market participants. During the settlement transaction process, the system executes the settlement calculation rules defined in the settlement calculation group referenced on the settlement subscription. The rules within a settlement calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

For information on configuration, reference the related section for Configuring Settlement Calculation Groups in the Administrative User Guide.

Settlement Calculation Rules

The specific calculation processing performed on a settlement transaction is defined in individual settlement calculation rules, each performing a specific set of targeted logic. The base product contains many settlement calculation rules you can use in your implementation, but you can also create your own custom settlement calculation rules. Many settlement

Copyright © 2026, Oracle and/or its affiliates.

calculation rules are identical to usage calculation rules used by Oracle Utilities Meter Data Management

For a list of validation settlement calculation rules, reference the About Usage Validations section. For a list of estimation usage calculation rules, reference the About Usage Estimations section. For a list of all other usage calculation rules that handle calculation of bill determinants, reference the Inventory of Other Usage Calculation Rules section.

For information on configuration, reference the related section for Configuring Settlement Calculation Rules in the Administrative User Guide.

Settlement Transaction Exceptions

Each settlement calculation rule defines an exception type and severity that specify how exceptions are tracked by the system. When a settlement transaction fails some part of a settlement calculation rule, an exception of the type specified for the failed settlement calculation rule is created. A single settlement transaction can have multiple exceptions, one (or more) for each rule that failed. This allows users to see all of the problems detected during the settlement calculation process.

There are three levels of severity for settlement transaction exceptions:

- **Information**: Used to highlight minor issues, but not sufficient to cause the settlement transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the settlement transaction from being exported. Multiple "issue exceptions" can be created during settlement transaction processing. If at least one issue exists after all rules have been applied, the settlement transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the settlement calculation process to stop and the settlement transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a settlement transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full settlement period should always be set to Terminate as an Exception Severity.

Note that exceptions are not deleted when a settlement transaction is adjusted or corrected. After any issues are corrected or the settlement transaction is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

In addition to exceptions, settlement processing can also trigger the creation of To Do Entries related to issues. If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the settlement transaction is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the settlement transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

For information on configuration, reference the related section for Configuring Usage Transaction Exception Types in the Administration Guide.

Settlement Calculation Rule Eligibility Criteria

Settlement calculation rule eligibility criteria are user-definable conditions that could cause a given settlement calculation rule to be applied or skipped. This can involve the evaluation of some attribute of the settlement subscription or something else entirely. A settlement calculation rule can have multiple eligibility criteria for determining if the rule should be applied

Copyright © 2026, Oracle and/or its affiliates.

or skipped, based on a user-defined sequence. Eligibility criteria for settlement calculation rules works in the same manner as eligibility criteria for usage calculation rules used by Oracle Utilities Meter Data Management.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administrative User Guide.

Settlement Calculations and Anchor Time

Settlement calculation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing settlement transactions. When processing settlement calculations based on dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

Note: When processing calculations for measuring components whose type specifies an **Anchor Time**, if the Start Time and End Time of the calculation period do NOT match the **Anchor Time** on the measuring component type, the Start and Stop Time are shifted to align with the closest time to the **Anchor Time** that aligns with the interval size of the measurement data being processed. The Start Time is shifted backwards (to an earlier time) and the End Time is shifted forwards (to a later time). In the case of hourly intervals with Start Times and End Times on the hour, no shift occurs. In the case of daily intervals, the shift is to the nearest Anchor Time. For example, in the Anchor Time is 07:00AM but the Start Time and End Time are both 12:00, the Start Time would be shifted to the previous 07:00AM, and the End Time would be shifted the next 07:00AM.

Settlement Subscriptions

- About Settlement Subscriptions
- Maintaining Settlement Subscriptions

About Settlement Subscriptions

Settlement subscriptions track the ongoing need to calculate a set of data, often aggregated load, for the purpose of settling a market and send the results to a recipient.

Settlement subscriptions must reference the following:

- **Recipient**: The identity of the subscribing system. In other words, you must set up a recipient for any system that subscribes to settlement calculations.
- **Settlement Calculation Groups**: The calculation groups used to perform settlement calculations for settlement transactions based on this settlement subscription. Settlement calculation groups can be configured in a specific sequence of execution. In addition, each group can be configured to require approval before the calculation execution proceeds to the next calculation group.

Settlement subscriptions can also reference the following;

- **Factor Overrides**: One or more factors can be used as override values when performing calculations
- **Measuring Components**: One or more measuring components directly linked to the settlement subscription. These are most often aggregation measuring components, but can also include profile measuring components.

Copyright © 2026, Oracle and/or its affiliates.

- **Measuring Component Sets**: One or more measuring component sets directly linked to the settlement subscription.

Maintaining Settlement Subscriptions

You use the portal **Settlement Subscription** portal to view and maintain settlement subscriptions.

Refer to About Settlement Subscriptions for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Subscription**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Subscription**: This zone displays basic information about the settlement subscription, including factor overrides, settlement calculation groups, measuring components and measuring component sets.
- **Settlement Transaction**: This zone displays a list of settlement transactions for the settlement subscription.

Settlement Transactions

- About Settlement Transactions
- Settlement Transaction Communication
- Maintaining Settlement Transactions

About Settlement Transactions

Settlement transactions are records of calculations for a settlement subscription. All of the service quantities calculated are stored as part of the settlement transaction.

Settlement Calculation Group Determination

The first step in settlement calculation is to determine the settlement calculation group holding the rules set used in calculating or validating the calculations. There are two ways of configuring the settlement calculation group to use in the processing. It may be specified on the settlement subscription during its creation with an effective and expiration date. If there are no settlement calculation groups in effect for the settlement transaction, the **Fallback Settlement Calculation Group** specified in the settlement subscription type will be used.

Calculation Period

Settlement transaction requests must specify the date range for the settlement transaction. This date range is referred to as the calculation period for the settlement transaction.

Date Breaks

Date breaks are used to break up a settlement period into sub-periods based on the dates on which changes took place for market participants. One approach to this situation would be for the system to crate settlement transactions (where each has the desired date range). Another

Copyright © 2026, Oracle and/or its affiliates.

approach is for the system to create single settlement transaction with date breaks that define the date ranges before and after the change.

If a settlement transaction has date break(s), the settlement calculation engine segregates the usage into multiple calculation periods based on the date breaks. If there are no date breaks in the settlement transaction, a single calculation period is created for the entire calculation period.

Service Quantities

Every settlement period created for a settlement transaction contains one or more service quantities. Service quantities are calculated by the settlement calculation group's rules specified for the settlement subscription from which the settlement transaction is created. For example, a settlement transaction might have quantities calculated by a single rule that applies a TOU map to aggregated kWh .

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

Quality Assessment

The Get Interval Data, Get Scalar Details, and Get Time of Use Mapped Usage calculation rules include a "Perform Measurement Quality Assessment" option that indicates (yes or no) if the measurement conditions of the measurements used when calculating usage should be evaluated and included in the resulting settlement transaction and outbound message. When enabled (set to "Yes"), the calculation rule algorithm evaluates the condition codes of the measurements used in the calculation to determine if they fall within a "Regular" range of conditions (this range is defined by the "Bottom Regular Condition Range" and "Top Regular Condition Range" algorithm parameters). The result of this assessment is displayed in the "Data Quality Assessment" column in the Calcuatio0n Periods section which indicates if the service quantity is based on "regular" or "non- regular" data. (If the "Perform Measurement Quality Assessment" option is disabled, the "Data Quality Assessment" column displays "No Assessment Done). In addition, the "Source Measurement Quality List" column contains XML that contains a list of measurements aggregated by condition/quality.

Validations

As settlement transactions are created and move through their various states, several validations are performed to ensure proper processing. These include:

- Validating the business object used to create the settlement transaction
- Validating the start and end dates for the settlement transaction
- Validating the settlement transaction’s service quantities
- Validating the existing settlement transactions for the same calculation period
- Validating whether or not the settlement transaction can be deleted

Creating Settlement Transactions Using Batch Processing

The Create Scheduled Settlement Transaction (D1-CSST) batch process can be used to generate settlement transactions based on a scheduled timetable.

The schedule for creating settlement transactions is defined based a Processing Timetable Type defined on the settlement subscription type. The process uses the details in the Processing Timetable Type (and its **Execution Frequency Class**) to determine the settlement transaction start and end dates. **Execution Frequency Class** options include:

Copyright © 2026, Oracle and/or its affiliates.

- **Daily**: The process runs based on the **Scheduling Information** parameters defined in the Processing Timetable Type .
- **Defined Dates**: The process runs for the most recent period defined in the **Processing Dates** section of the Processing Timetable Type that falls before the Batch Business Date. (but after the **Start Date** on the settlement subscription)

The business object to use for creating settlement transaction(s) is based on what is configured on the settlement subscription type. If there is no processing timetable type specified on a settlement subscription type, it will be skipped.

By default the process creates settlement transactions for all settlement subscription types. The optional "Restrict by Settlement Subscription Type" (restrictBySettlementSubType) parameter can be used to restrict processing to a specific Settlement Subscription Type.

Refer to Processing Timetable Types for more information about processing timetable types.

Settlement Transaction Communication

Upon completion of calculations (or approval of calculations if the approval is required), settlement transaction results can be exported and sent to market participants This is done via an outbound communication created by the Create Settlement Transaction Outbound Communication algorithm. This communication is responsible for deriving and gathering the appropriate data to send to the market participant and creating and sending an outbound message.

There are two many ways in which settlement transactions are sent:

- **Contract**: Sending by contract involves sending the results of a settlement transaction to a recipient (such as an ISO) using an identifier from the market contract.
- **Identifier**: Sending by identifier involves sending the results of a settlement transaction to a recipient using an identifier from the market participant.

Communication Types

The type of outbound communication created is based on the **Outbound Communication Type** defined on the settlement transaction's settlement subscription type. Communication types define the following about settlement transaction outbound communications:

- **Communication Flow**: The direction of communications based on this type (outbound or inbound).
- **Suppress Zeros**: Designates whether or not data should be sent for a service quantity of 0.
- **Message Generation Method**: Specifics if messages be sent as a single message to the market or broken down to one message per Identifier/Contract.
- **External Communication Type**: Defines whether the communication type is "Settlement by Contract" or "Settlement by Identifier".
- **Export Data Identification**: Defines combinations of UOM/TOU/SQI that represent the final outcome of the settlement transaction to be passed to the market
- **Reconciliation**: Defines the appropriate identifier or contract to include in the data sent to the market participant:
  - **Contract Reconciliation**: defines the appropriate identifier for the market contract associated with the service quantity. Used when sending data by contract.

Copyright © 2026, Oracle and/or its affiliates.

- **Identifier Reconciliation**: Defines the characteristic type for the market participant used to retrieve the appropriate identifier. Used when sending data by identifier.

- **Exception Handling**: Defines how exceptions are handled, including To Do Type and Role, Retry Frequency, and the maximum number of times the system will retry to process the communication

Retrieving Service Quantities

Once created, the outbound communication retrieves a list of service quantities for the settlement transaction and inserts the details into the Settlement Transaction Details list. The specifics of how this is done is based on whether the settlement transaction is sent by contract or by identifier.

Send Settlement Transaction By Contract

Sending data by contract uses outbound communications based on the Send Settlement Transaction by Contract business object. The Retrieve Settlement Transaction SQs Contract algorithm retrieves the list of service quantities that match the UOM/TOU/SQI combination defined on the communication type, and for each service quantity found that has a market contract it will retrieve the identifier for that contract and populate the appropriate details into the Settlement Transaction Details list. The algorithm processes the entire service quantity result set even if it cannot find an identifier for a given contract. This way a user can see all identifier resolution errors at once (instead of needing to fix them one by one to see the next error).

Send Settlement Transaction By Identifier

Sending data by identifier uses outbound communications based on the Send Settlement Transaction by Identifier business object. The Retrieve Settlement Transaction SQs for Identifier algorithm extracts the market participant from the attributes associated to the service quantity's measuring component. The measuring components that will be associated to the service quantity in this instance will be aggregation measuring components that hold a list of dimensions that incudes a market participant. The algorithm then retrieves the list of service quantities that match the UOM/TOU/SQI combination defined on the communication type and for each service quantity found that has a measuring component, it retrieves the measuring component's market participant and then retrieve the identifier for that market participant and populate the appropriate details into the Settlement Transaction Details list. The algorithm processes the entire service quantity result set even if it cannot find a market participant or an identifier for a given market participant. This way a user can see all identifier resolution errors at once (instead of needing to fix them one by one to see the next error).

- Sending Settlement Transactions to Market Participants

Sending Settlement Transactions to Market Participants

After the outbound communication has retrieved the appropriate service quantities, the outbound message creates an outbound message to the market participant (defined as the **Recipient** on the settlement transaction's settlement subscription or settlement subscription type).

The Send Settlement Transaction Details algorithm creates one more outbound messages based on the outbound message type defined for the market participant's Settlement Transaction Notification processing method (see below for more details).

- If the **Message Generation Method** on the communication type is set to "One Comprehensive Message" the outbound communication creates a single outbound message that contains all the service quantities retrieved.

Copyright © 2026, Oracle and/or its affiliates.

- If the **Message Generation Method** on the communication type is set to “Multiple Messages Based on Data” the outbound communication creates a separate outbound message for each of the service quantities retrieved.

Configuring Market Participants for Settlement Transaction Communication

To support sending settlement transactions to market participants, the following processing method must be defined for the market participants to which settlement transactions are sent:

Processing Role	Processing Method
Settlement Transaction Notification	How to Send Settlement Subscription Related Information Used to send a settlement transaction outbound message to a market participant. **Default Outbound Message Type**: The default outbound message type to create for outbound message sent to market participants. The base package does not include an outbound message business object for this. This must be created based on the requirements of each implementation and market participant. If appropriate, override outbound message types can be defined for specific settlement subscription types.

In addition, the following objects must be configured to send data requests:

- External System
- Outbound Message Type
- Message Sender

`

Maintaining Settlement Transactions

You use the portal **Settlement Transaction** portal to view and maintain a settlement transactions.

Refer to About Settlement Transactions for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Transaction**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Transaction**: This zone displays basic information about the settlement transaction, including processing information and settlement calculation groups.
- **Outbound Communication**: This zone displays a list of outbound communications sent for the settlement transaction.
- **Calculated Data Overview**: This zone displays an overview of calculated data related to the settlement transaction.
- **Calculated Data Details**: This zone provides ability to query the detailed service quantity interval data for the settlement transaction.

A number of important actions are available for a settlement transaction depending on its status:

Copyright © 2026, Oracle and/or its affiliates.

Action	Allowed Statuses	Details
Approve	Approval In Progress Issue Detected (only when the exceptions are marked to require approval)	Allows a user to bypass any exception or failure message for the settlement transaction and attempt to send it to the requesting system.
Calculate	Approval In Progress Pending	Allows a user to trigger settlement calculations.
Discard	Approval In Progress Export Error Issue Detected	Allows a user to discard the settlement transaction.
Export	Export Error	Allows a user to transition the settlement transaction to the Export status and send the settlement transaction to the recipients.
Export Manually	Export Error	Allows a user to manually export the settlement transaction send it to the recipients.
Override Exception	Issue Detected	Allows a user to override an exception and transition the settlement transaction to the Override Exception state.
Re-Calculate	Issue Detected	Reruns the entire process of calculation along with any validations (all settlement calculation rules) for the settlement transaction.
Supercede	Completed	Allows a user to indicate that the settlement transaction was corrected after being sent. Sets the status of the settlement transaction to Superceded.

Batch Processing

The Create Scheduled Settlement Transaction (D1-CSST) batch process can be used to generate settlement transactions based on a scheduled timetable. Refer to Creating Settlement Transactions Using Batch Processing for more information about this batch process.

Settlement Items

- About Settlement Items
- Maintaining Settlement Items

About Settlement Items

Copyright © 2026, Oracle and/or its affiliates.

Settlement items represent end customer accounts. These are market participants for which settlement calculations are performed, and generally correspond to the market participants which subscribe to settlement calculation results via settlement subscriptions.

Settlement items are defined by the following:

- **Settlement Item Type**: The Settlement Item Type for the settlement item, which defines the business object to use to create the settlement item, the business object to use to capture billed usage for a settlement item (via Settlement Account Billed Usage), and valid service point types.
- **Division**: The division to which the settlement item belongs
- **Access Group**: The access group to which the settlement item belongs (used to limit access to records)
- **Status**: The status of the settlement item
- **Start Date/End Date**: The date range during which the settlement item is active.
- **Main Contact**: The main contact for the settlement item.
- **Time Zone**: The time zone in which the settlement item is active.
- **Measuring Components**: One or more measuring components directly linked to the settlement item. These are most often aggregation measuring components, but can also include profile measuring components.
- **Rate Class**: One or more effective-dated rate classes for the settlement item.

Maintaining Settlement Items

You use the portal **Settlement Item** portal to view and maintain a settlement items.

Refer to About Settlement Items for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item**: This zone displays basic information about the settlement item, including related service points and measuring components.
- **Settlement Item Quantities**: This zone displays a list of settlement item quantities. See About Settlement Item Quantities for more information. To add a new quantity, for the settlement item, click **Add** in the zone title bar.
- **Settlement Item Market Participants**: This zone displays a list of settlement item market participants for the settlement item, via a Settlement Item Market Participant. To add a new market participant for the settlement item, click **Add** in the zone title bar.
- **Characteristics**: This zone displays a list of characteristics related to the settlement item.
- **Identifiers**: This zone displays a list of identifiers used with the settlement item.
- **Interval Measurements**: This zone displays a list of interval measurement data snapshots associated with the settlement item. To add a new interval measurement data snapshot for the settlement item, click **Add** in the zone title bar.
- **Scalar Measurements**: This zone displays a list of scalar measurement data snapshots associated with the settlement item. To add a new scalar measurement data snapshot for the settlement item, click **Add** in the zone title bar.

Copyright © 2026, Oracle and/or its affiliates.

Settlement Item Quantities

- About Settlement Item Quantities
- Maintaining Settlement Item Quantities

About Settlement Item Quantities

Settlement item quantities are quantities that are stored at the settlement item level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given settlement item to the market wide coincidence peaks.

Settlement item quantities are defined by the following:

- **Settlement Item**: The settlement item for which the quantities were calculated
- **Settlement Item Quantity Type**: The Settlement Item Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Settlement Item Quantities**: Calculated quantities based on **Settlement Item Quantity Identifiers** configured on the Settlement Item Quantity Type (see Understanding Settlement Item Quantity Types for more information)

> **Note**
>
> There cannot be more than one Settlement Item Quantity for the same Settlement Item and Settlement Item Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

Maintaining Settlement Item Quantities

You use the **Settlement Item Quantity** portal to view and maintain settlement item quantities.

Refer to About Settlement Item Quantities for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Quantity**: This zone displays the basic attributes of the settlement item quantity

Settlement Billed Usage

- About Settlement Billed Usage

Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Settlement Billed Usage

About Settlement Billed Usage

Settlement billed usage represents billed usage for a settlement item. Settlement billed usage is calculated via settlement calculations performed for the settlement subscription associated with the settlement item. Settlement billed usage typically corresponds to settlement transactions.

Settlement billed usage records are defined by the following:

- **Settlement Item**: The settlement item for which the billed usage was calculated
- **Start Date/Time**: The date and time of the start of the billing period for the billed usage
- **End Date/Time**: The date and time of the end of the billing period for the billed usage
- **Billed Usage Periods**: one or more billed usage periods for the current settlement billed usage. A billed usage period is a period of time for which usage is calculated, including details calculated for each billed usage period such as UOM/TOU/SQI and quantities

Maintaining Settlement Billed Usage

You use the portal **Settlement Billed Usage** portal to view and maintain settlement billed usage.

Refer to About Settlement Billed Usage for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Billed Usage**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Billed Usage**: This zone displays basic information about the settlement billed usage, including billed usage periods.

Settlement Item Market Participants

- About Settlement Item Market Participants
- Maintaining Settlement Item Market Participants

About Settlement Item Market Participants

Settlement item market participants connects a Market, Market Participant and Settlement Item together over a specified date range. Settlement item market participants can be created either via the **Settlement Item Market Participant** portal, or by adding a market participant to a settlement item via the **Settlement Item** portal.

Settlement item market participants are defined by the following:

- **Settlement Item**: The settlement item for which this settlement item market participant was created.

Copyright © 2026, Oracle and/or its affiliates.

- **Market Relationship Type**: The market relationship (Distributor, Supplier, etc.) of the market participant and its role within a market.
- **Start Date/Time**: The start date/time for when the record is applicable.
- **End Date/Time**: The end date/time for when the record is applicable.
- **External ID**: An ID for the market participant used by external systems.
- **Market Participant**: The market participant associated with the **Settlement Item**, including an optional ID used by the market participant to uniquely identify the settlement item.

Maintaining Settlement Item Market Participants

You use the **Settlement Item Market Participant** portal to display and maintain settlement item market participants.

Refer to About Settlement Item Market Participants for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item Market Participant**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Market Participant**: This zone displays the basic attributes of the settlement item market participant.
- **Sub Settlement Items**: This zone lists settlement items for the settlement item market participant.

Market Allocations

- Market Contracts
- Market Awards
- Market Products

Market Contracts

- About Market Contracts
- Maintaining Market Contracts

About Market Contracts

Market contract are agreements with the market operator (such as an Independent System Operator, or ISO) related to how settlement calculation results are sent to the market. Market contracts are related to a specific market, have a seller and buyer, as well as a date range for when they are effective.

Market contracts are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Market Contract Type**: The Market Contract Type for the contract. This defines the market in which the contract is in effect and the business object used to create the contract.
- **Contract** : A unique code for the contract, often provided by an external system
- **Start Date** / **End Date**: The date range during which the contract is in effect
- **Seller**: The market participant that serves as the Seller for the contract.
- **Buyer**: The market participant that serves as the Buyer for the contract.

Maintaining Market Contracts

You use the **Market Contract** portal to view and maintain market contracts.

Refer to About Market Contracts for additional functional information.

You can access the portal from the **Main > Market > Market Contract**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Contract**: This zone displays basic information about the market contract.
- **Related Market Awards**: This zone displays a list of market awards related to the market contract.

Market Awards

- About Market Awards
- Maintaining Market Awards
- Maintaining Market Allocations

About Market Awards

Market awards are used to allocate load to various market participants.

Market awards are defined by the following:

- **Market Product**: The Market Product associated with the award
- **Award Class**: The type of award. The base package supports three award classes: Tranche Awards, Percentage Awards, and Fixed Profile Awards.
  - **Tranche Awards**: Tranches are used to handle a whole number division of load. For example, an agreement may be reached that the Provider of Last Resort (POLR) load for a zone should be broken into 64 tranches, or exactly 64 chunks of load that can be purchased. In this example, the load could be broken up in the following manner: Contract #1234 buys 12 tranches, Contract #1235 buys 1 tranche, and Contract #1236 buys 51 tranches (each of these three would be a market award) A factor can be used to define different tranche quantities over time for the same market award.
  - **Percentage Awards**: Percentage Awards allocate a percentage of the remaining load. A factor can be used to define a different percentage over time for the same market award.

Copyright © 2026, Oracle and/or its affiliates.

  - **Fixed Profile Awards**: Fixed Profile Awards leverage a profile measuring component to designate an amount that must be allocated from the overall load to this market award. A factor can be used to define different profiles over time for the same market award.
- **Start Date / End Date**: The date range for the market award
- **Award Details**: Details of the award, including the source (Static Value or Factor) and quantity (either a fixed value or factor value)
- **Characteristics**: Characteristics for the award, if appropriate

Maintaining Market Awards

You use the **Market Award** portal to view and maintain market awards based on Market Products with a **Product Class** of “Contract Fulfillment”.

Refer to About Market Awards for additional functional information.

You can access the portal from the **Main > Market > Market Award**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Award**: This zone displays basic information about the market award, including **Award Details**, related market **Contracts**, and **Characteristics**.

Maintaining Market Allocations

You use the **Market Allocation** portal to view and maintain market awards based on Market Products with a **Product Class** of “Parent Allocation”.

Refer to About Market Awards for additional functional information.

You can access the portal by selecting **Main**, then **Market**, then **Market Award**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the **Market Allocation** portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Allocation**: This zone displays basic information about the market award, including **Award Details**, and **Characteristics**.

Market Products

- About Market Products
- Maintaining Market Products

About Market Products

Market products are used to allocate market awards to market participants. Market products are referenced by the Market Award Allocation settlement calculation rule via a parent Market Product Set. Market products of class “Parent Allocation” can have child products. This allows definition of hierarchies of market products and related awards,

Copyright © 2026, Oracle and/or its affiliates.

Market products are defined by the following:

- **Market Product Set**: The Market Product Set to which the Market Product is associated.
- **Market Product Type**: The Market Product Type for the market product. This defines the business object used to create the market product and its market awards. For formula-based products, the market product type also defines the formula used to calculate award allocation.
- **Product Code**: A unique code for the product, often provided by an external system
- **Product Class**: The type of product. The base package supports two product classes: Parent Allocation or Contract Fulfillment:
  - **Parent Allocation**: Indicates that the market product is a parent level product that divides out a certain allocation of the total amount to be then divided further amongst child products.
  - **Contract Fulfillment**: Indicates that the market product ties to an "end result" award that can then be tied to a Market Contract
- **Execution Sequence**: Defines the sequence in which the market product will be executed within its parent Market Product Set
- **Awards**: One or more Market Awards associated with the market product. These define how load and procurement are allocated when the market product is processed by the Market Award Allocation settlement calculation rule.

Maintaining Market Products

You use the **Market Product** portal to view and maintain market products.

Refer to About Market Products for additional functional information.

You can access the portal from the **Main > Market > Market Product**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product**: This zone displays basic information about the market product.
- **Awards**: This zone displays a list of market awards related to the market product.

Data Access Restrictions

- Using Data Access Restrictions

Using Data Access Restrictions

The focus for multi-jurisdiction support in Oracle Utilities Meter Data Management is on restricting access to the user interface. This occurs through defining different Access Groups by operating company which then marks the Service Point with the access capabilities. The Service Point has a field named **Access Group** which allows you to define specific access to the Service Point.

Copyright © 2026, Oracle and/or its affiliates.

If Access Group is set on the Service Point, then the user interface search screens within Oracle Utilities Meter Data Management will only allow users that are assigned to the proper Access Role to be able to access that Service Point.

For additional information see Defining Access Groups.

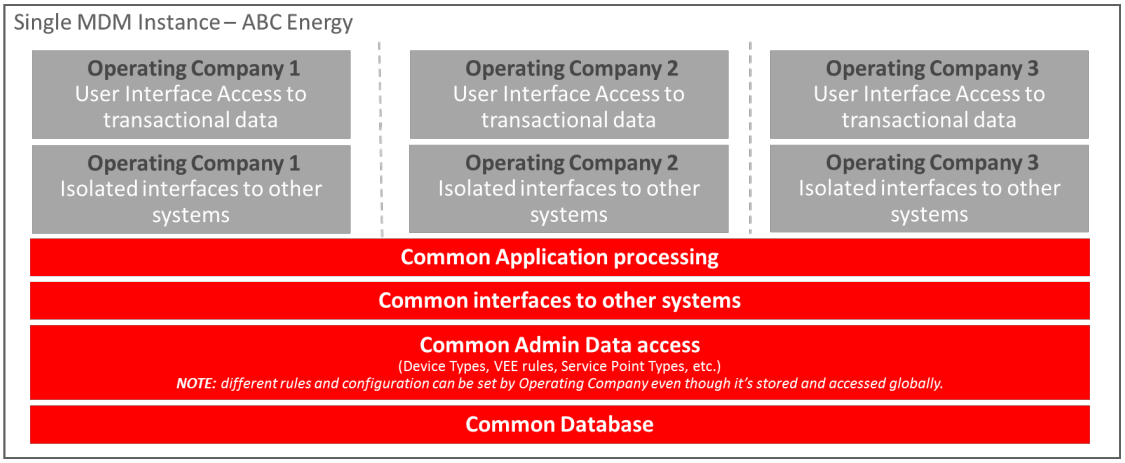
Defaulting Access Groups on Service Points

If there is a desire to set a default Access Group by jurisdiction, then this can be accomplished on configuration for Markets. When defining a Market there is an option to add a **Default Access Group**. This value will be used to mark any newly created Service Point with the same Access Group value defined on the related Market.

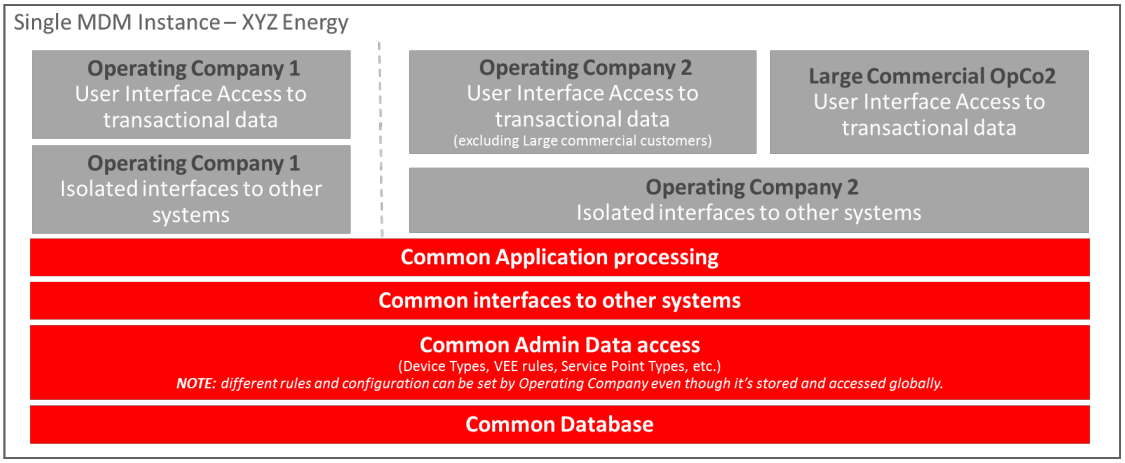
Example Scenarios

Below are some examples of how Data Access Restrictions can be implemented:

Example 1: Split Access by Market - an example utility "ABC Energy" that has 3 operating companies.



Example 2: Isolating Important Customers - an example utility "XYZ Energy" that has 2 operating companies but would also like to isolate access to Large commercial customers for one of the companies.



Copyright © 2026, Oracle and/or its affiliates.

To Do Processing

There are numerous events that trigger the creation of email-like messages describing work that requires human attention. For example, if a bill segment has an error, a message is created to highlight such. We refer to each message as a **To Do Entry**. In this section, we describe how to view, manage and work To Do entries.

> **NOT_SUPPORTED**
>
> We recommend familiarizing yourself with the topics described in The Big Picture of To Do Lists to fully appreciate how To Do entries are managed by the system.

- Understanding To Do Processing
- Managing To Do Entries

Understanding To Do Processing

The topics in this section provide background information on To Do processing functionality.

- To Do Role
- To Do Priority
- Work Distribution
- Working On A To Do
- To Do Characteristics
- Related To Do Entries
- Complete A To Do
- Log Entry Events

To Do Role

Each To Do entry is assigned a specific To Do Role. The role defines the users who may work on the entry. The permissible roles that may be assigned to a To Do entry are defined on the entry's To Do type. One or more users within a role may be designated as a supervisor, who is allowed to assign To Do entries to other people. Supervisors are determined through application security access.

Refer to To Do Entries Reference A Role to learn more about role assignment.

After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type. Manual changes of a To Do entry's role are recorded on the To Do entry log.

To Do Priority

A To Do entry is associated with a priority level representing its processing order compared to other entries. Refer to The Priority Of A To Do Entry for more information on how an entry's priority is initially assigned and how it may be automatically updated by the system.

Copyright © 2026, Oracle and/or its affiliates.

A user may manually override a To Do entry's priority at any time. Once overridden, automatic recalculation of the priority is no longer performed so as not to override the value explicitly set by the user. Manual overrides of a To Do entry's priority are recorded on the To Do entry log.

> **Note**
>
> Only users that are currently **Assigned** to a To Do entry or ones that are valid for the role of an **Open** To Do entry may manually change the priority.

Work Distribution

This topic goes overs the various possible actions a user can take to adjust what role or what user is associated with a To Do entry while it is in a non-complete state.

Adjusting the Role

A To Do entry must always reference a role. When a To Do entry are created in the **Open** state and is associated with a role. The following points highlight ways to adjust the role associated with a To Do entry.

- A user viewing a record on the To Do Entry page may use the Forward or Send Back action to change the role.
- A supervisor using the To Do Search page may assign a user not linked to the current role. In this case, the system will update the role to one valid for the assignee.

Assigning a To Do to a User

There are several ways to **Assign** a To Do entry to a user:

- The To Do Management portal provides many filters and viewing options to review To Do entries for a give To Do Type. In addition, it displays a rich amount of detail for the resulting To Do entries, helping users to make decisions. Users may use this page to assign work to themselves and supervisors may use this to assign work to others.
- A user viewing an entry on the To Do Entry page may use the Forward to a User or the Send Back action to assign the entry to a different user.
- **On demand assignment**. The system can be configured to dynamically assign the highest priority work to a user on demand from the Assign me a To Do button on the Current To Do Zone. One of the main benefits of the assignment on demand approach is that work is allocated real-time based on most current information such as skills, priority, user availability, etc. This also means that work remains unassigned until a user is actually going to work on it making it available to any qualified user.

  > **Note**
  >
  > Refer to On Demand Assignment Configuration for information on configuring the system to support this feature.

- Several legacy pages include assignment capability
  - A supervisor can use the Supervisor Functions pages to review, assign and reassign To Dos to users.
  - A user may use the To Do list pages to review and assign To Dos to themselves.

Copyright © 2026, Oracle and/or its affiliates.

- All of the above assignment functions may also be performed using the To Do Search page.

Reopening / Unassigning a To Do

It is possible to transition a To Do entry from **Assigned** back to **Open**.

- A user viewing a record assigned to them on the To Do Entry page may use the Forward to a role action to move the entry back to **Open**.
- The **Unassign** button on the To Do Management portal or the **Reopen** button on the To Do Search page causes the To Do entry to go back to **Open**.

Working On A To Do

The To Do entry you are actively working on is your **In Progress** To Do entry. This To Do entry is visible in the Current To Do zone. Refer to that zone's help topic for all the features enabled on that zone.

> **Important**
>
> The information here assumes that you have configured the system to use the ability to have a substatus of Assigned, which includes the **In Progress** value. If you are not taking advantage of the substatus, then see below for information about how the functionality described here works differently.

If you don't currently have an **In Progress** To Do entry, there are several ways to choose a To Do and indicate that you want to start working on it. Doing so proactively populates the To Do into the Current To Do zone and navigates you to the related object (or if the To Do type is configured to launch a script, the script is launched).

If your organization's practice is to assign work to users up front, then a typical approach is that you would start by reviewing a list of entries assigned to you using one of the base pages and indicate that you want to work on an entry in the list. Or your organization may allow you to review To Do entries that are open and in a role that you are in. In some pages, there is a way for you to enable a list of To Do Entries that you can work through without needing to return to the query page to get the next entry.

The following points highlight the available base pages and their supported functionality.

- To Do Management allows you to use filters to view a list of entries assigned to you or that are open. Click the Work List icon in the Action column to populate the Work List zone in the sidebar with the results of the search. Click the **Work** button to choose the To Do entry you wish to start with. This drills you into the related object and populates the Current To Do zone with this To Do. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you). Use the entries in the Work List zone to subsequently work on other To Do entries.
- If you are viewing a record on To Do Entry that you are allowed to work on, click the **Work** button to indicate that you want to start working on the record. This drills you into the related object and populates the Current To Do zone with this To Do. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you).
- To Do List allows you to start working on a To Do by clicking the entry's message description link. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you). Any other To Do entries that were in that results list on that page become an internal work list and the **Previous** button and **Next** button are enabled in the Current To Do zone to work through that list. Note that clicking

Copyright © 2026, Oracle and/or its affiliates.

the next or previous button does not automatically assign the entry to you. Use the **Start Progress** button on the Current To Do zone to change the status. If you don't start progress, it means the To Do in this zone is not your current In Progress to do entry. It remains in this zone until you perform another action to populate a different To Do in this zone. If you open another browser window, the Current To Do zone will not have this same To Do entry populated.

- To Do Search allows you to drill into the details of a To Do by clicking the entry's message description link, but it does not automatically assign it to you or update it to **In Progress**. You can click **Start Progress** on the Current To Do zone once it's populated. Any other To Do entries that were in that results list on that page become an internal work list and the **Previous** button and **Next** button are enabled in the Current To Do zone to work through that list. As with the first To Do selected, the subsequent To Do entries will not be assigned to you or progress started automatically.

> **Note**
>
> The To Do Search works differently than other pages in that you can drill in on a To Do entry and push the details into the Current To Do zone even if the To Do entry is assigned to someone else or if you are not allowed to work on that To Do. If you have an **In Progress** To Do entry and use the To Do Search to drill in on a different To Do Entry that doesn't get assigned to you, that To Do is now the one in the Current To Do zone and remains there until you do something else to refresh the data in that zone. If you open another browser window, the Current To Do zone will not have this same To Do entry populated.

Refer to Current To Do for more information about actions available to update To Do entries that you are working on.

Additionally, you may have the option to click on the **Assign me a To Do** button on the Current To Do zone to request your next task. The returned To Do entry automatically becomes the your **In Progress** To Do. This does not enable any work list functionality.

> **Note**
>
> If you do already have an **In Progress** To Do and you perform one of the above actions that results in a different To Do to become **In Progress**, the To Do entry that was already **In Progress** is updated to **On Hold**.

> **Important**
>
> If your implementation is not taking advantage of the substatus as described in The Lifecycle Of A To Do Entry, then the system doesn't have a persistent way of marking which To Do entry is the one you are working on. You may still perform any of the steps described above to make the To Do entry your 'current' To Do. If any of the above steps indicate that the To Do will be assigned to you, that action still occurs, but the To Do entry is not moved to 'in progress'. The To Do entry is only considered the 'current To Do' for your current session. If a different session is opened, you must perform the explicit action again to make the To Do the 'current To Do' for that session.

Copyright © 2026, Oracle and/or its affiliates.

To Do Characteristics

Characteristics are used to associated related information to a To Do entry. Refer to Linking Additional Information to a To Do Entry for more information.

Several pages in the system allow you to view a To Do entry's characteristic values and / or search for To Do entries by their characteristics.

- The To Do Management portal and the To Do Details tab on the To Do Dashboard portal include up to five filters to search for To Do entries by a characteristic value. In addition, the search results display all the current characteristics as name / value pairs.
- The To Do Entry includes a zone to display any existing characteristics. In addition, if a user is allowed to modify a To Do Entry, the edit dialogue allows a user to add or change characteristics. Note that for manual To Do types, the valid characteristic types are driven by the To Do type. For automated To Do types, if valid To Do characteristic types are configured on the To Do type, only those are available. Otherwise, any valid To Do characteristic type may be linked.
- The To Do Search includes a single characteristic type / value filter.

Related To Do Entries

A user working on a To Do may want to know of other To Dos that may be related to the same problem. If they can be worked by the same user they may get resolved faster. It seems almost impossible to programmatically determine whether two To Dos really stem from the same problem. The user should therefore be presented with a list of potentially related To Dos from which they can select the truly relevant ones.

A To Do entry is considered potentially related to a given To Do if they reference a common entity via a characteristic. For example, if a given To Do references a certain account then any To Do referencing that same account is considered related to it.

> **Note**
>
> The common entity is any characteristic value linked to the To Do entry where the characteristic type is a **Foreign Key** and the main table of the foreign key is classified as a **Master** or **Transaction** table.

There are various places where the user is informed as to whether or not the current To Do Entry is related to other To Do Entries.

- To Do entry maintenance displays in insight indicating the number of related To Do Entries for the current To Do. In addition, a Related To Do Entries zone shows information about each related To Do.
- The Current To Do zone includes related To Do information and actions.
- The To Do Management query includes a count of related To Do entries.
- The To Do Search page supports a **Related To Do filter** that lists all related To Do entries for a given entry.

Copyright © 2026, Oracle and/or its affiliates.

Complete A To Do

After you finish working on an entry, you can mark it as **Complete**. At this point, you may want to review potentially related To Do entries and complete those that have been addressed by resolving the problem associated with the current entry.

When you complete a To Do, a log entry is created that indicates when the To Do was completed and by whom.

There are multiple ways in which you can complete a To Do.

- The To Do Management page provides many filters in addition to To Do Type. One or more entries may be completed in this page.
- Use the To Do Entry page to complete a single To Do Entry. The user may also choose to add comments when completing a To Do on this page.
- You may use the Current To Do zone to complete the current entry you are working on and/or all related entries assigned to you.
- The To Do List page allows you to complete one or more selected entries of a specific To Do Type.
- The To Do Search page allows you to search for To Do entries using various search criteria and complete selected ones.

> **Note**
>
> It is possible for the system to automatically mark an entry as **Complete**. Refer to The Lifecycle Of A To Do Entry for more information.

Log Entry Events

Any user may add a **User Details** log entry to any To Do at any time (even after it's completed). Refer to To Do Entry page for more information.

The To Do Management portal also allows users to add a **User Details** log entry to one or more selected entries.

In addition, there are several log entries created automatically by the system. The log includes a status column, which captures the status of the To Do at the time of log. If the To Do entry is **Assigned** and its substatus **In Progress** or **On Hold**, the log captures the substatus value, otherwise it captures the status value.

The following table contains the list of all possible log events.

Event	Log Type	Log Status	Content of Details	Content of Assigned To (on To Do entry and To Do Log)	Log User ID	Contents of Sent By (on To Do entry)
A To Do is created (by the system or by a user)	**Created**	**Open**			ID of user who created the To Do	

Copyright © 2026, Oracle and/or its affiliates.

Event	Log Type	Log Status	Content of Details	Content of Assigned To (on To Do entry and To Do Log)	Log User ID	Contents of Sent By (on To Do entry)
A To Do is created (by the system or by a user) in Assigned status	**Created**	**Assigned**		User ID of user who was assigned this To Do	ID of user who created the To Do	Creating user's ID
A To Do is completed (by the system or by a user)	**Completed**	**Complete**			ID of user who completed the To Do	
A user takes an open To Do	**Assigned**	**Assigned**	Selected by user	Assigned To User ID	Assigned To User ID	ID of user who assigned the To Do
A supervisor assigns a To Do to a user (also happens if they reassign)	**Assigned**	**Assigned**	Assigned to user	Assigned To User ID	ID of user who assigned the To Do	ID of user who assigned the To Do
A user forwards an entry to a user	**Forwarded**	**Assigned**	*Log details entered by user*	Forwarded To User ID	ID of user who forwarded the To Do	ID of user who forwarded the To Do
A user forwards an entry to a role	**Forwarded**	**Open**	Forwarded to role *Role Description (Role ID)* + *Log details entered by user*		ID of user who forwarded the To Do	ID of user who forwarded the To Do
A user sends back a To Do	**Sent back**	**Assigned**	*Log details entered by user*	Sent Back To User ID	ID of user who sent back the To Do	ID of user who sent back the To Do
The status is updated (aside from the cases that have an explicit log type)	**Status Updated**	Current status of To Do		User currently Assigned, if applicable	ID of the user the made the status update	
A user adds a log entry	**User Details**	Current status of To Do	*Log details entered by user*		ID of user who added the entry	
A user manually overrides priority	**Override Priority**	Current status of To Do	Priority changed from *old value* to *new value*	Assigned To User ID	ID of user who made the change	

Copyright © 2026, Oracle and/or its affiliates.

Event	Log Type	Log Status	Content of Details	Content of Assigned To (on To Do entry and To Do Log)	Log User ID	Contents of Sent By (on To Do entry)
Priority is updated by a calculate priority algorithm	**Calculated Priority**	Current status of To Do	Priority changed from *old value* to *new value*		ID of user that added or updated the To Do or the user associated with the Recalculate Priority batch job.	

Managing To Do Entries

This section describes the various pages available to view and work on To Do entries.

- To Do Management
- To Do Dashboard
- To Do Entry
- Billing To Do Dashboard
- Legacy UI Features

To Do Management

The To Do Management portal provides a multi-query zone that includes a search for To Do entries using To Do Type and other To Do information. Users may search for entries using multiple criteria and perform common actions, such as assignment and completion, on selected entries.

It is intended to be used by both supervisors and non-supervisor To Do users. Supervisors are identified by their security access to the portal.

Use **Menu**, and then **To Do**, and then **To Do Management** to navigate to the portal.

To Do Information Search

This zone allows you to search for non-closed To Do entries using various attributes of the entries, and to perform actions on selected ones.

> **Note**
>
> **Results are restricted by role.** For supervisors, this search returns To Do entries that meet the search criteria and are in a role of which the supervisor is a member. For non supervisors, the search returns open To Do entries that meet the search criteria and are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The following sections describe the features of this zone.

Copyright © 2026, Oracle and/or its affiliates.

Search Filters

The following table describes each of the different filters supported by this zone.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection. If the To Do Role filter is populated, the list is further restricted to valid To Do Types for the To Do Role.
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	On the **To Do Management** portal, this option is represented by a pair of filter fields. Use these filters to restrict the search to To Do entries whose age falls within a given range in days. For example, entering values of 2 - 5 will restrict the search to entries that are greater than or equal to two days old and less than or equal to five days old. On the **To Do Dashboard** portal, this option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
To Do Role	Use this filter to restrict the search to To Do entries of a specific role. Only To Do Roles where the user is a member will be available for selection. If the To Do Type filter is populated, the list is further restricted to valid To Do Roles for the To Do Type.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Message	Use this filter to restrict the search to To Do entries with a specific message. This filter appears after the To Do Type filter has been populated and shows a list of the message texts from all non closed To Do entries of the selected To Do Type.
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. If the To Do Type filter is populated and the To Do Type is 'manual', the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown. If an automated To Do Type doesn't define valid characteristics or if no To Do Type filter is populated, all valid characteristic types for the To Do Entry entity are shown. Note that if you have populated characteristic filters and then you change the To Do Type filter, the system retains the characteristic filter as long as the characteristics are applicable to the newly selected To Do Type.

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> For performance reasons, because the filter criteria may be broad, the search zone is configured to only get the first 5000 records that match the criteria and will then sort that number to then display the first 400 sorted by priority and age. The To Do Dashboard portal may be used to get exact counts of records with certain criteria. Use additional criteria to narrow down the search as needed.

Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined. The following columns are displayed initially:

- The Action column includes the button Work. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

  > **Important**
  >
  > **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

  > **Note**
  >
  > **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.

  > **Note**
  >
  > Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.

Copyright © 2026, Oracle and/or its affiliates.

- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string. By default this list is collapsed to one line to save vertical space. If additional characteristics exist, you can use the expand arrow to display them.

> **Note**
>
> A feature configuration option is supplied to opt out of the compact view. Under the **To Do Options** feature configuration type, define the **Collapse Characteristics Column** option and set the value to "N".

- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.
- **To Do Role** displays the description of the role for the To Do entry.

Actions

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

Copyright © 2026, Oracle and/or its affiliates.

To Do Dashboard

This portal provides a high level summary of open and assigned To Do entries by various criteria. It is intended to be used by both supervisors and non-supervisor To Do users. An implementation may introduce additional zones to this portal as needed.

> **Note**
>
> **Results are restricted by role.** For supervisors, the charts consider To Do entries that are in a role of which the supervisor is a member. For non supervisors, the charts consider open To Do entries that are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

Use **Menu**, and then **To Do**, and then **To Do Dashboard** to navigate to the portal.

The following zones may appear as part of the portal's **Main** tab page

- **Filters.** This zone displays and manages filters restricting the information being displayed.
- **To Do Entries By Role.** This zone summarizes To Do entries by their currently assigned role.
- **To Do Entries By Status.** This zone summarizes To Do entries by their current status.
- **To Do Entries By Priority.** This zone summarizes To Do entries by their priority.
- **To Do Entries By Age.** This zone summarizes To Do entries by age based on age ranges defined for the current user.
- **To Do Entries By Type.** This zone summarizes To Do entries by their to do type.

Selecting a specific slice of data on any of these zones sets it as a filter and broadcasts its value to the other zones on the portal allowing them to filter their data accordingly.

The following zone appears as part of the portal's **Details** tab page

- **To Do Details** This zone lists To Do entries that match various criteria and allows the user to work and manage the list.

To Do Details

This zone allows you to search for non-closed To Do entries using various attributes of the entries, and to perform actions on selected ones.

> **Note**
>
> **Results are restricted by role.** For supervisors, this search returns To Do entries that meet the search criteria and are in a role of which the supervisor is a member. For non supervisors, the search returns open To Do entries that meet the search criteria and are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The following sections describe the features of this zone.

Search Filters

Copyright © 2026, Oracle and/or its affiliates.

The following table describes each of the different filters supported by this zone.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection. If the To Do Role filter is populated, the list is further restricted to valid To Do Types for the To Do Role.
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	On the **To Do Management** portal, this option is represented by a pair of filter fields. Use these filters to restrict the search to To Do entries whose age falls within a given range in days. For example, entering values of 2 - 5 will restrict the search to entries that are greater than or equal to two days old and less than or equal to five days old. On the **To Do Dashboard** portal, this option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
To Do Role	Use this filter to restrict the search to To Do entries of a specific role. Only To Do Roles where the user is a member will be available for selection. If the To Do Type filter is populated, the list is further restricted to valid To Do Roles for the To Do Type.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Message	Use this filter to restrict the search to To Do entries with a specific message. This filter appears after the To Do Type filter has been populated and shows a list of the message texts from all non closed To Do entries of the selected To Do Type.
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. If the To Do Type filter is populated and the To Do Type is 'manual', the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown. If an automated To Do Type doesn't define valid characteristics or if no To Do Type filter is populated, all valid characteristic types for the To Do Entry entity are shown. Note that if you have populated characteristic filters and then you change the To Do Type filter, the system retains the characteristic filter as long as the characteristics are applicable to the newly selected To Do Type.

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> For performance reasons, because the filter criteria may be broad, the search zone is configured to only get the first 5000 records that match the criteria and will then sort that number to then display the first 400 sorted by priority and age. The To Do Dashboard portal may be used to get exact counts of records with certain criteria. Use additional criteria to narrow down the search as needed.

Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined. The following columns are displayed initially:

- The Action column includes the button Work. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

  > **Important**
  >
  > **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

  > **Note**
  >
  > **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.

  > **Note**
  >
  > Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.

Copyright © 2026, Oracle and/or its affiliates.

- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string. By default this list is collapsed to one line to save vertical space. If additional characteristics exist, you can use the expand arrow to display them.

> **Note**
>
> A feature configuration option is supplied to opt out of the compact view. Under the **To Do Options** feature configuration type, define the **Collapse Characteristics Column** option and set the value to "N".

- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.
- **To Do Role** displays the description of the role for the To Do entry.

Actions

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

Copyright © 2026, Oracle and/or its affiliates.

To Do Entry

Refer to Understanding To Do Processing for detailed explanations about many To Do entry features.

To view a To Do entry, use **Menu**, and then **To Do**, and then **To Do Entry**, and then **Search** . You are brought to a query portal with options for searching for a To Do Entry. Note that some query options limit the results. Refer to the inline help for more information.

> **Note**
>
> **Restricted by role.** Any user may change Comments for any To Do Entry. In addition, any user may add user Log details. However, for all other changes and actions, you may only change To Do entries that are assigned to you or ones that are **Open** and assigned to a role to which you belong. Refer to To Do Entries Reference A Role for more information about roles.

Once a To Do Entry has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The inline help provides information about several fields. The following points highlight information about some of the fields that may be visible.

If the To Do entry has non-complete related entries an insight appears indicating the number of related entries. In addition, details about the related entries are shown in the Related To Do Entries zone.

If the To Do entry was **Created By** a background process, information about that background process appears with a link allowing the user to drill into the batch run, if desired. Refer to To Dos Created for Object-Specific Error Conditions for more information.

If the To Do type is configured to To Do Entries May Be Routed Out Of The System, the **Extract Info** appears. If the extract process is pending, the information indicates that. Otherwise, if it has run, information about the background process appears with a link allowing the user to drill into the batch run, if desired.

The **Characteristics** collection shows existing characteristics. This collection is maintained using the Edit action for users that are allowed to update this To Do entry. Refer to To Do Entry Characteristics for more information.

Actions

Most of the actions are self-explanatory and are related to simply updating the status value. This page has three special actions.

- Using the **Work** button causes several things to occur. You drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. If the To Do entry is Open, it gets assigned to you and the assigned substatus is set to **In Progress**, if your implementation has implemented this functionality. Refer to What Happens When You Work A To Do Entry for more information.
- Use the **Forward** to assign a To Do to another user or role. When a To Do is forwarded, a log entry indicates who forwarded the To Do, to whom it was forwarded, and the specified reason. The dialogue prompts for whether to send to a User or a Role. The search for User

Copyright © 2026, Oracle and/or its affiliates.

includes information about the number of To Do entries already assigned to this user. When choosing a User, confirm the correct Role. The Details provided are captured in the log entry.

- **Send Back** is only applicable if the **Sent By** user is populated. When a To Do is sent back, a log entry is created that indicates when the To Do was sent back and by whom. If the To Do is **Open**, this dialogue allows you to send it back to a different role. If the To Do is assigned to you, the dialogue allows you to re-assign the To Do to the person who assigned it to you. The Details provided are captured in the log entry.

Related To Do Entries

If the To Do has related To Do entries, this zone appears with information about the related entries. Use this zone to perform actions on the current To Do and its related entries.

- Adding a Manual To Do Entry

Adding a Manual To Do Entry

If your system is configured with manual To Do types, you can add a To Do entry online.

Open the **Menu**, and then **To Do**, and then **To Do Entry**, and then **Add.**

You are prompted for a **To Do Type**. Only To Do types that are set up for manual usage appear in the list.

The default **To Do Type Priority** is displayed adjacent. Use **Override Priority** to override the default priority assigned by the system.

Type a **Subject** for your To Do entry. Depending on how the To Do type is set up, the subject may appear as the message in the To Do entry and To Do lists.

Enter a **Comment** to provide a longer description of and more details about the To Do entry.

Send To indicates whether the new To Do will be assigned to a **Role** (the default) or **User**.

The **Role** indicates the role to which this To Do entry will be assigned. Only users of the specified role will be able to view and work on the To Do entry. The role is defaulted from the To Do type. If you want to assign the To Do to a different role, change the specified role. You can only assign the To Do entry to one of the roles specified on the To Do type.

User appears if **Send To** is set to **User**. Use this option if you want to assign the To Do to a specific user. The user must be a member of the selected role.

If the To Do type defines valid characteristic types, the **Characteristics** collection is visible. If any of the characteristic types are flagged as Default on the To Do type, the characteristic type with its default value (if applicable) are displayed.

Billing To Do Dashboard

This portal provides a high-level summary of To Do Entry counts by various criteria. It is intended for supervisor To Do users. An implementation may introduce additional zones to this portal as needed.

The To Do Entry counts are retrieved from statistics snapshots. Refer to the following statistics snapshot business objects in the application, for more information:

- C1-NCToDoPerBillCycleSnapshot - Non-Complete To Dos Per Bill Cycle Snapshot
- C1-ToDoCreatedSnapshot - To Dos Created in the Last n Days Snapshot
- C1-NCToDoPerMessageSnapshot - Non-Complete To Dos Per Message Snapshot

Copyright © 2026, Oracle and/or its affiliates.

For information about capturing statistics, refer to Capturing Statistics

The zones display the counts of To Do Entries of To Do Types with the 'Billing' To Do Type Category (F1-TDCAT) predefined characteristic. The base product provides the C1BILL characteristic value, for this purpose.

Use **Main Menu**, and then **To Do**, and then **Billing To Do Dashboard** to navigate to the portal.

> **Note**
>
> The Billing To Do Dashboard Configuration master configuration can be optionally configured with settings that control how the statistics are shown on this portal - i.e., maximum number of open bill cycles to retrieve, display mode (messages vs. message groups), message groups (list), maximum number of messages to retrieve, maximum number of messages groups to retrieve, To Do trend days and period average. This master configuration is optional. If not configured, the statistics snapshots will run with the assumed defaults. Refer to the master configuration's embedded help, for more information.

The following zones may appear on load this portal:

- **Non-Complete To Dos Per Bill Cycle.** This zone contains a table/grid that summarizes To Do entry counts by **To Do Role** and bill cycle **Window End Date**. It displays the count for each open bill cycle and a single count for To Dos associated with closed bill cycles (i.e., under the **Past Due** column). Use the filters, to further restrict the counts shown.
  - The statistics are restricted by the To Do characteristics: Account (C1-ACCT) and Bill Cycle (C1-BILCY).
  - The maximum number of open bill cycles to retrieve is configurable in the Billing To Do Dashboard master configuration. If not configured, the assumed default is ten (10) open bill cycles.
  - Refer to the Calculate Non-Complete To Dos Per Bill Cycle (C1CALCTDBC) algorithm, for more information.
  - Selecting a count from this zone will navigate to the **Details** tab page, where the To Do Entries for that count are listed, using the **Bill Cycle** query.
- **To Dos Created in the Last N Days.** This zone contains a line chart that summarizes To Do entry counts by Create Date. It displays the daily count of To Dos from the business date (day 1) to a number of days prior, where the number of days is configurable. A Period Average is also configurable. Use the filters, to further restrict the counts shown.
  - The number of days for the daily count is configurable in the To Do Trend Days setting of the Billing To Do Dashboard master configuration. If not configured, the assumed default is thirty (30) days.
  - The Period Average is configured in the Billing To Do Dashboard master configuration. If not configured, the assumed value is 0.
  - Refer to the Calculate To Dos Created in the Last n Days (C1CALCTDCRE) algorithm, for more information.
  - Selecting a count from this zone will navigate to the **Details** tab page, where the To Do Entries for that count are listed, using the **Create Date** query.
- **Non-Complete To Dos Per Message.** This zone contains a row chart that summarizes To Do entry counts by either message category and message number, or by message group (if applicable). Use the filters, to further restrict the counts shown.

Copyright © 2026, Oracle and/or its affiliates.

- The Display Mode in the Billing To Do Dashboard master configuration controls how the statistics are shown on this zone. If the master configuration does not exist, the assumed display mode is by message category and message number.
- The **Message Groups** display mode is useful when the volume of billing-related To Do message numbers is large. One or more message groups can be defined in the master configuration. Each message group defines one or more related message numbers.
- The maximum number of messages to retrieve is configurable. If not configured, the assumed limit is fifty (50) message numbers.
- The maximum number of message groups to retrieve is configurable. If not configured, the assumed limit is twenty-five (25) message groups.
- Refer to the Calculate Non-Complete To Dos Per Message (C1CALCTDMSG) algorithm, for more information.
- Selecting a count from this zone will navigate to the **Details** tab page, where the To Do Entries for that count are listed, using the **Message** query.

Billing To Do Details

The **Details** multi-query zone allows you to view the To Do entry list for a selected statistics count (from the Main tab), and to perform actions on selected records. Each query supports the unique information that each statistics snapshot contains:

- **Search by Bill Cycle.** This query is used when navigating from the **Non-Complete To Dos Per Bill Cycle** zone. It includes filters for: Bill Cycle and Window End Date range.
- **Search by Create Date.** This query is used when navigating from the **To Dos Created in the Last N Days** zone. It includes filters for Create Date range.
- **Search by Message.** This query is used when navigating from the **Non-Complete To Dos Per Message** zone. It includes filters for: Message Category, Message Number and Message Group. The display mode controls the applicable filter's visibility.

The following sections describe the features of this tab page portal.

Search By Bill Cycle - Filters

The following table describes each of the query filters.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection.
To Do Role	This is defaulted to the selected To Do Role (from the Non-Complete To Dos Per Bill Cycle zone). Only To Do Roles where the user is a member of will be available for selection.
Bill Cycle	Use these filters to restrict the search to To Do entries of a specific Bill Cycle.
Window End Date From, Window End Date To	Use this filter to restrict the search to To Do entries of a specific Window End Date range.
CIS Division	Use this filter to restrict the search to To Do entries of a specific CIS Division.
Customer Class	Use this filter to restrict the search to To Do entries of a specific Customer Class.

Copyright © 2026, Oracle and/or its affiliates.

Search Filter	Description
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	This option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. For manual To Do types, the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown; otherwise, all valid characteristic types for the To Do Entry entity are shown.

Search By Create Date - Filters

The following table describes each of the query filters.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection.
To Do Role	This is defaulted to the selected To Do Role (from the Non-Complete To Dos Per Bill Cycle zone). Only To Do Roles where the user is a member of will be available for selection.
Creation Date From, Creation Date To	Use these filters to restrict the search to To Do entries of a specific Create Date range.
Bill Cycle	Use this filter to restrict the search to To Do entries of a specific Bill Cycle.
CIS Division	Use this filter to restrict the search to To Do entries of a specific CIS Division.
Customer Class	Use this filter to restrict the search to To Do entries of a specific Customer Class.
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	This option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.

Copyright © 2026, Oracle and/or its affiliates.

Search Filter	Description
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. For manual To Do types, the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown; otherwise, all valid characteristic types for the To Do Entry entity are shown.

Search By Message - Filters

The following table describes each of the query filters.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection.
To Do Role	Use this filter to restrict the search to To Do entries of a specific To Do Role. Only To Do Roles where the user is a member of will be available for selection.
Message Category, Message Number	Use this filter to restrict the search to To Do entries of a Message Category and Message Number. These filters are available when the **Display Mode** is **Messages** (either specified in Billing To Do Dashboard Configuration master configuration or the assumed default in effect).
Message Group	Use this filter to restrict the search to To Do entries of a specific Message Group. This filter is available when the **Display Mode** is **Message Groups** that is configured in the Billing To Do Dashboard Configuration master configuration.
CIS Division	Use this filter to restrict the search to To Do entries of a specific CIS Division.
Customer Class	Use this filter to restrict the search to To Do entries of a specific Customer Class.
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	This option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. For manual To Do types, the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do Type, only those are shown; otherwise, all valid characteristic types for the To Do Entry entity are shown.

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> For performance reasons, the search zone is configured to restrict the number of rows retrieved by the query. The 'Total Records' number will indicate whether there are more To Do entries that meet the criteria than can be displayed.

Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined as a parameter to the zone. The following columns are displayed initially:

- The **Action** column includes the **Work** button. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

  > **Important**
  >
  > **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

  > **Note**
  >
  > **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.

  > **Note**
  >
  > Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.

Copyright © 2026, Oracle and/or its affiliates.

- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string.
- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.

Actions

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

Legacy UI Features

The pages described in this section are legacy pages that are retained for backward compatibility. The product plans to deprecate the pages described here in an upcoming release.

- To Do Supervisor Functions
- To Do List Functions
- To Do Search

Copyright © 2026, Oracle and/or its affiliates.

To Do Supervisor Functions

This section describes several To Do transactions that aid supervisors in the monitoring and assignment of To Do entries.

- Supervisor To Do Summary
- Supervisor User Summary
- Supervisor To Do Assignment

Supervisor To Do Summary

This Supervisor To Do Summary page summarizes To Do entries by To Do Type. Only entries that are **Open** (i.e., not assigned to a specific user) or **Assigned** are summarized.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Dashboard portal for the page that the product recommends using. Besides providing counts by To Do Type it provides other breakdowns and allows for interaction between graphs.

> **Note**
>
> Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to To Do Entries Reference A Role for more information.

Use **Menu**, and then **To Do**, and then **Supervisor To Do Summary** to open this page.

Description of Page

This page summarizes all To Do entries that are **Open** or **Assigned**.

> **Note**
>
> This page does not further break down the assigned To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Assigned** entries. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Open** and **Assigned** entries for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Open** entries for the given To Do Type.

Copyright © 2026, Oracle and/or its affiliates.

- **Assigned** contains the number of To Do entries whose status is **Assigned**. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Assigned** entries for the given To Do Type.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is user-configurable.

Supervisor User Summary

The Supervisor User Summary page summarizes To Do entries by the user who is assigned to work on the entries. Only entries that are **Assigned** are summarized.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that allows you to view To Do entries currently assigned to users.

> **Note**
>
> Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to To Do Entries Reference A Role for more information.

Use **Menu**, and then **To Do**, and then **Supervisor User Summary** to open this page.

Description of Page

This page contains a grid that displays all assigned To Do entries by User ID and To Do type. It indicates how many To Do entries of the displayed Type are assigned to the user (Assigned this Type) and how many To Do entries in total are assigned to the user (Assigned Total).

> **Note**
>
> This page does not further break down the To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.

Clicking the hyperlinks transfer you to the Supervisor To Do Assignment transaction. On this transaction, you can assign / reassign To Do entries to a specific user. The difference is that clicking the To Do Type hyperlink causes only **Open** entries to be displayed on the Assignment transaction. Clicking the count in the Assigned this Type column causes **Assigned** entries assigned to the displayed user to be shown.

Supervisor To Do Assignment

This page lists To Do entries of a given type and allows a supervisor to assign / reassign them to a designated user.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for the page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and provides more actions for a supervisor to perform for To Do entries.

Use **Menu**, and then **To Do**, and then **Supervisor To Do Assignment** to open this page.

Description of Page

This page contains **Open** or **Assigned** To Do entries of a given type. These entries can be assigned to a specified user one-by-one or several at a time. The following points describe this functionality:

- You can select a different **To Do Type** by typing in a value or pressing the search button in the upper right corner. An **Info** description of the To Do type appears on the left.
- You can change the list of entries displayed according to their status. Select the **Filter by** status to be All (blank), **Open**, or **Assigned**.
- For **Assigned** entries, you may also **Filter by User** to limit the search to a specific user assigned to the To Do entries.
- If you want to re-sort the entries by a different value, click the search button adjacent to **Extra Information**. Each row's respective sort value is displayed in the **Extra Information** column.

> **Note**
>
> **Extra Information.** Each To Do list's default **Extra Information** is defined on the list's To Do Type. Also note that each To Do list's available **Extra Information** differs.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to assign specific entries.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Click the Create Date/Time hyperlink to navigate to the To Do page where additional information is visible and where additional actions are possible.
- Click the number of **Related** entries hyperlink to navigate to the To Do Search query to list potentially related entries to the current entry.
- **Worked By** shows the user currently assigned to the To Do entry.
- **To Do Entry Status** is the current status of the To Do entry. If the assigned substatus is populated, that value is shown, otherwise the status value is shown. Refer to The Lifecycle Of A To Do Entry for more information.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Note that clicking this action does not push the To Do Entry into the Current To Do zone and does

Copyright © 2026, Oracle and/or its affiliates.

not assign the To Do Entry to you. Refer to Working On A To Do for information about using that functionality.

- You can assign entries to a user:
  - Enter a user ID or search for a user using the **Assign To User** field. You can only select a user that is associated with a role that is associated with this **To Do Type**. Note that the current number of assignments is displayed adjacent.
  - Select one or more entries to assign to this user and click the **Assign** button. The entries' status becomes **Assigned**.

To Do List Functions

This section describes To Do pages that aid users in the assignment and working of entries in To Do lists.

- To Do Summary
- To Do List

To Do Summary

This page summarizes To Do entries that meet the following qualifications:

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Dashboard portal for the page that the product recommends using. Besides providing counts by To Do Type it provides other breakdowns and allows for interaction between graphs.

- The entry is **Open** and you belong to the entry's To Do role. Refer to To Do Entries Reference A Role for more information about roles.
- The entry is **Assigned** and you are the user to which it is assigned.

Use **Menu**, and then **To Do**, and then **To Do Summary** to open this page.

Keyboard Shortcut	Alternate Shortcut
`Alt+X`	`Alt+Shift+X`

> **Note**
>
> Refer to Shortcut Key Summary for information about the alternate shortcut.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of To Do Lists for an overview of To Do functionality.

Description of Page

This page is dedicated to a grid that summarizes the To Do entries on which you can work. The following information is displayed:

Copyright © 2026, Oracle and/or its affiliates.

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Assigned to You** entries. The hyperlink navigates to the To Do List transaction, setting it to display all **Open** and **Assigned** entries assigned to you for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the To Do List transaction, setting it to display all **Open** entries for the given To Do Type.
- **Assigned to You** contains the number of To Do entries that are assigned to you whose status is **Assigned**. The hyperlink navigates to the To Do List transaction, setting it to display all **Assigned** entries for the given To Do Type.

> **Note**
>
> This page does not further break down the assigned To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is user-configurable.

To Do List

This page lists To Do entries of a given type that are either **Open** (not assigned to a specific user) or are assigned to you.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and allows the user to adjust the priority of To Do entries.

> **Note**
>
> **Restricted by role.** Only those entries assigned to a role to which you belong may be seen on this page. Refer to To Do Entries Reference A Role for more information.

> **NOT_SUPPORTED**
>
> We recommend familiarizing yourself with the topics described in The Big Picture of To Do Lists to fully appreciate how To Do entries are managed by the system.

Use **Menu**, and then **To Do**, and then **To Do List** to open this page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

This page contains a grid that summarizes the To Do entries of a given type on which you can work.

The **Info** section contains a concatenation of the **To Do Type**'s description and the user's name.

The area beneath **Info** provides you with options that control which To Do entries appear and the order in which they are displayed. The following points describe the various options:

- Use **Sort By** to define how you want the To Do entries sorted. The following options are available:
  - **Assigned Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
  - **Assigned Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
  - **Create Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
  - **Create Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
  - **Extra Information**. This option sorts the To Do entries by the value of the **Extra Info**.
- Use **Extra Info** to choose data related to the To Do Entry that should be used to sort the To Do entries. This information is displayed in each entry's **Extra Information** column.

> **Note**
>
> **Extra Info.** Each To Do types's available **Extra Info** is defined on the To Do Type Sort Keys. Also note that each To Do type's available **Extra Info** differs based on the configuration of the To Do Type.

- Use **Filter by** to restrict the To Do entries based on their status. The following options are available:
  - **All**. This option shows all entries that are **Open** or **Assigned**.
  - **Assigned**. This option shows all entries that are assigned to you.
  - **Open**. This option shows all entries that are **Open**.

The grid that follows contains the To Do entries that match your criteria. The following information appears:

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** button to remove the selections in the grid.

Copyright © 2026, Oracle and/or its affiliates.

- Use the checkbox if you want to mark specific entries as **Completed** or **Assigned** using the **Complete** or **Assign** buttons at the bottom.
- A Comment icon appears if there are comments on the To Do item and a Log Entry icon appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do has either comments or a special log entry.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- Click the Create Date/Time hyperlink or either of the special icons to navigate to the To Do entry page.
- Click the number of **Related** entries hyperlink to navigate to the To Do Search query to list potentially related entries to the current entry.
- **Assigned Date/Time** contains the date and time the To Do entry was assigned to a user. This field is only populated on entries that are **Assigned**. If a user assigned the To Do entry, the user ID and name of the user who assigned the To Do entry are displayed beneath the Date/Time.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Important**
>
> **Automatic Assignment** When you drill into an open entry using the message link, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

- **To Do Entry Status** is the current status of the To Do entry. If the assigned substatus is populated, that value is shown, otherwise the status value is shown. Refer to The Lifecycle Of A To Do Entry for more information.
- **Extra Information** contains the data for the entry that corresponds to the Extra Info chosen in the sort options above.
- **To Do ID** contains the unique identifier of the To Do entry.
- If you want to mark one or more entries as **Assigned**, check the desired entries and then click the **Assign** button.
- If you want to mark one or more entries as **Completed**, check the desired entries and then click the **Complete** button. Completed entries do not appear on the To Do list (but they are retained on the database for audit purposes).

To Do Search

This page allows you to search for To Do entries using various search criteria, and perform various actions on selected ones. This page includes completed To Do entries in its search results, when applicable with respect to the filter criteria.

Use **Menu**, and then **To Do**, and then **To Do Search** to open this page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and allows the user to adjust the priority of To Do entries.

> **Note**
>
> **No restriction by role.** This page returns all To Do entries that meet the search criteria, regardless of the To Do entry's role. Using the search, a user may drill into the details of a To Do entry even if the To Do is not assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The top half of the page is where you enter the criteria used to search for To Do entries.

> **Caution**
>
> Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times.

The following table describes each of the different search filters.

Search Filter	Description
Related To Do	If you're searching for potentially related entries for a given To Do entry, choose **Specific To Do** for the **Related To Do Filter**. Enter the **To Do ID**. A filter value of **Not Applicable** defaults.
To Do Type	If you're searching for a To Do entry and you know the value of a To Do Type linked to the To Do entry, choose **Specific To Do Type** for the **To Do Type Filter**. Enter the **To Do Type**. A filter value of **All** defaults.
Char Type	If you're searching for a To Do entry and you know the value of a characteristic linked to the To Do entry, choose **Specific Char Type** for the **Char Type Filter**. Enter the **Characteristic Type** and **Characteristic Value**. A filter value of **Not Applicable** defaults. Only characteristic types that include a characteristic entity of **To Do Entry** and that have been marked to allow searching by characteristic value are available.
User	Use this search method if you know the **Created By User**, **Assigned To User** or **Completed By User** for the To Do entry. Enter the appropriate **User Id**. A filter value of **Not Applicable** defaults.
Date Range	Use this field if you're searching for a To Do entry and know the **Date Created** or **Date Completed**. A filter value of **Date Created** defaults. The **Start Date/Time** defaults to the current date, 12:00AM and the **End Date/Time** defaults to the current date, 11:59PM. If you enter a **Date Range Filter** of **All**, the system ignores this search method.

Copyright © 2026, Oracle and/or its affiliates.

Search Filter	Description
Status	Use this search method to restrict the search to To Do entries with a specific status value. You may search for To Do entries in the status of **Open**, **Assigned** or **Completed**. You may also perform a combination search for To Do entries that have a status value of either **Open** or **Assigned**. A filter value of **All** defaults.

> **Note**
>
> **The system allows any combination of values.** The system does not try to validate your search combinations, even if they do not make sense. For example, if you enter a **User Filter** value of **Completed By User** and a **Status Filter** value of **Open**, you will not get an error. In this case, the system will find no records that match your criteria (because To Do entries that are Open do not have a Completed By User).

The system shows the total number of To Do entries that satisfy your search results immediately below the grid.

The first group of To Do entries is displayed in the grid at the bottom of the page.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to perform actions (see below) on specific entries. Only **Open** or **Assigned** entries may be selected.
- A Comment icon appears if there are comments on the To Do item and a Log Entry icon appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do item has either comments or a special log entry. Use either icon to navigate to the To Do entry page.
- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Use the hyperlink to navigate to the To Do entry page.
- Click the number of **Related** entries hyperlink to redisplay this page listing potentially related entries to the current entry.
- **Last Assigned Date/Time** contains the date and time the To Do entry was most recently assigned to a user, if applicable, along with the Assigned To user.
- **Message** contains a description of the To Do entry. If the To Do entry is in the status of **Open** or **Assigned**, the message description allows you to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Current To Do**. Drilling into the page by clicking the Message hyperlink does not automatically assign **Open** To Do entries to the user (as the To Do List page does when clicking on the Message). But it does treat this as the user's current to do, enabling the functionality in the Current To Do zone. Refer to Working On A To Do for more information.

- **Status** contains the current status of the To Do entry.
- **Completion Date/Time** contains the date and time the To Do entry was completed and the user who completed the entry, if applicable.
- **To Do Type** contains a description of the To Do type for the To Do entry.

Actions

- Use the **Assign** button to assign or reassign the selected entries to a user. Enter a user ID or search for a user using the **Assign To User** field. By default this field is populated with the current user's ID. Note that the user's current number of assignments is displayed adjacent. Upon successful processing, the selected entries are assigned to the specified user and the status of **Open** entries becomes **Assigned**. Additionally, a corresponding log entry is added to each To Do.

  > **Note**
  >
  > You may only assign To Do entries to yourself or to someone else if you are a valid user in the To Do entry's current role.

  > **Note**
  >
  > Only users with **Supervisor Assignment** access rights may reassign entries that are not currently assigned to them.

- Use the **Reopen** button to unassign the selected entries from their users. Upon successful processing, the selected entries become **Open** and a corresponding log entry is added to each To Do.

  > **Note**
  >
  > Only users with **Supervisor Assignment** access rights may unassign entries that are not currently assigned to them and the user must be a valid user in the To Do entry's current role.

- Use the **Complete** button to complete the selected entries.

Reports

- About Reports
- Running Oracle Utilities Meter Data Management Reports

Copyright © 2026, Oracle and/or its affiliates.

About Reports

Reports include operational reports related to device events, validation exceptions, and other operational data. Reports can also be based on devices, measuring components, Service Points, service providers, and other data.

Oracle Utilities Meter Data Management reports are implemented using Oracle Business Intelligence Publisher (BI Publisher), and are administered via Oracle Business Intelligence Publisher and the Report functionality of the Oracle Utilities Application Framework.

Refer to the Oracle Business Intelligence Publisher documentation for more information about installing Oracle Business Intelligence Publisher and creating and defining reports. For additional details on how reports are defined and administered see Reporting and Monitoring Tools in the Oracle Utilities Application Framework *Administrative User Guide*.

Running Oracle Utilities Meter Data Management Reports

This portal is used to submit a new instance of a Report.

Refer to About Reports for additional functional information.

You can access the portal from the **Main > Tools > Report Submission**.

First, the appropriate Report code must be selected. The Reports provided out-of-the-box are:

- **Device Events Report**: this report generates a pie graph depicting the percentage of Device Events for each Device Event Type for the specific search criteria.
- **Device VEE Exceptions Report**: this report generates a pie graph depicting the percentage of VEE Exception for Exception Type and Market for the specific search criteria.

Next, enter the parameter values for the report as appropriate and submit the report.

Copyright © 2026, Oracle and/or its affiliates.

Part III

Application Framework Administrative User Guide

Welcome to the Oracle Utilities Application Framework Administrative User Guide. The topics in this guide describe how to administer the Oracle Utilities Application Framework.

- General Options
- Security & User Options
- User Interface Tools
- Database Options
- Configuration Tools
- To Do Lists
- Background Processes
- Attachments
- Monitoring and Analytics
- External Messages
- Integrations
- Market Transaction Management
- Content Migration Assistant (CMA)
- Facts
- Conversion
- Data Export

General Options

This section describes control tables that are used throughout your product.

- Defining Installation Options
- Support For Different Languages
- Defining Countries
- Defining Currency Codes
- Defining Time Zones
- Defining Geographic Types
- Defining Work Calendar
- Defining Display Profiles
- Defining Phone Types

Copyright © 2026, Oracle and/or its affiliates.

- Defining Feature Configurations
- Defining Master Configurations
- What's New Portal

Defining Installation Options

The topics in this section describe the various installation options that control various aspects of the system.

- Installation Options - Main
- Installation Options - Messages
- Installation Options - Algorithms
- Installation Options - Accessible Modules
- Installation Options - Installed Products

Installation Options - Main

Select **Admin**, and then **General**, and then **Installation Options - Framework** to define system wide installation options.

Description of Page

The **Environment ID** is a unique universal identifier of this instance of the system. When the system is installed, the environment id is populated with a six digit random number. While it is highly unlikely that multiple installs of the system at a given implementation would have the same environment ID, it is the obligation of the implementers to ensure that the environment ID is unique across all installed product environments.

System Owner will be **Customer Modification**.

The **Admin Menu Order** controls how the various control tables are grouped on Admin.

- If you choose **Functional**, each control table appears under a menu item that corresponds with its functional area. Note, the menu that is used when this option is chosen is the one identified with a menu type of **Admin**.
- If you choose **Alphabetical**, each control table appears under a menu item that corresponds with its first letter, using a Roman alphabet. For example, the Language control table will appear under the L menu item entry.

> **Note**
>
> The **Alphabetical** option only supports the Roman alphabet. For languages that do not use the Roman alphabet, the recommendation is to configure the system for the **Functional** setting.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> In order to improve response times, installation options are cached the first time they are used after a web server is started. If you change the Admin Menu Order and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. Refer to Caching Overview for information on how to clear the system login cache (this is the cache in which installation options are stored).

The **Language** should be set to the primary language used by the installation. Note that if multiple languages are supported, each user may define their preferred language.

The **Currency Code** is the default currency code for transactions in the product.

If your product supports effective dated characteristics on any of its objects, define the date to be used as the **Characteristic Default Date** on objects without an implicit start date. The date you enter in this field will default when new characteristics are added to these objects (and the default date can be overridden by the user).

Active Owner displays the owner of newly added system data (system data is data like algorithm types, zones, To Do types, etc.). This will be **Customer Modification** unless you are working within a development region.

Country and **Time Zone** represent the default country and time zone that should be used throughout the application.

> **⚠ Caution**
>
> In most implementations, the time zone defined here matches the database time zone. However, if there is some reason that the database time zone does not match the installation time zone, an implementation may configure a setting in the properties file to automatically convert data from the database time zone to the time zone defined here when displaying dates. Note that when this property setting is defined, changes to the installation time zone will require the server and the thread pool workers to be restarted in order for the changes to take effect.

Turn on **Seasonal Time Shift** if your company requires seasonal time shift information to be defined. Note that this is currently only applicable to Oracle Customer Care and Billing > Interval Billing functionality.

Installation Options - Messages

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Messages** tab to review or enter messages that will appear throughout the application when a given event occurs.

The **Message** collection contains messages that are used in various parts of the system. For each message, define the **Installation Message Type** and **Installation Message Text**. The following table describes the **Message Types** provided by the framework product and how they are used in the system. Your specific product may have introduced addition message types.

Copyright © 2026, Oracle and/or its affiliates.

Message Type	How The Message Is Used
Company Title for Reports	This message appears as a title line on the sample reports provided with the system. Generally it is your company name. It is only used if you have installed reporting functionality and are using the sample reports (or have designed your reports to use this message).
Domain Name	The text entered for this message type is meant to identify the environment. If populated, the text is displayed next to the product name in The Application Toolbar. It may be used in other cases where it is useful to indicate the environment for a user. For example, the email sent when a background process finishes includes this information, if populated.

Installation Options - Algorithms

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Algorithms** tab to review or enter the algorithms that should be evoked when a given event occurs.

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

> ⚠ **Caution**
>
> These algorithms are typically significant processes. The absence of an algorithm might prevent the system from operating correctly.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Validate Email Attachment	Optional	Algorithms of this type are used to validate the attachments for size and total count while sending attachments using the Email service. Refer to Sending Email for more information
Address Geocoding	Optional	Algorithms of this type use Oracle Locator to retrieve latitude and longitude coordinates using address information.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Attachment Malware Scan	Optional	Algorithms of this type are invoked when adding attachment content. The algorithm receives the attachment ID and is expected to use a malware scanning tool to verify the contents of the attachment. Only one algorithm is allowed. Refer to Enabling Malware Scan for more information.
Global Context	Optional	Algorithms of this type are called whenever the value of one of the global context fields is changed. Algorithms of this type are responsible for populating other global context values based on the new value of the field that was changed. Refer to Global Context Overview for more information.
Guaranteed Delivery	Optional	Algorithms of this type may be called by processes that receive incoming messages that should 'guarantee delivery'. Refer to Guaranteed Delivery for more information. The business service `F1-GuaranteedDelivery` may be used to invoke this plug-in spot.
Ldap Import	Optional	Algorithms of this type are called for operations on users, groups, and group memberships after they have been processed.
Ldap Import Preprocess	Optional	Algorithms of this type are called to preprocess data retrieved from LDAP.
Next To Do Assignment	Optional	This type of algorithm is used to find the next To Do entry a user should work on. It is called from the Current To Do zone when the user asks for the next assignment. Refer to On Demand Assignment Configuration for more information about setting up this information.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Reporting Tool	Optional	If your installation has integrated with a third party reporting tool, you may wish to allow your users to submit reports on-line using report submission or to review report history online. This algorithm is used by the two on-line reporting pages to properly invoke the reporting tool from within the system.
SMS Receive	Optional	This type of algorithm is used to provide SMS receive service. Only one algorithm of this type should be plugged in.
SMS Send	Optional	This type of algorithm is used to provide SMS send service. If your installation uses the base algorithm that uses BPEL, you will need to create a feature configuration with the SMS Send Configuration feature type to define your Oracle BPEL server and service call details. If your installation has integrated with a third-party SMS service, you may want to override this algorithm type with your own implementation. Only one algorithm of this type should be plugged in.
To Do Information	Optional	We use the term To Do information to describe the basic information that appears throughout the system to describe a To Do entry. Plug an algorithm into this spot to override the system default "To Do information".
To Do Pre-creation	Optional	These types of algorithms are called when a To Do entry is being added. Refer to Refining a To Do Entry's Details at Creation Time for more information.

Installation Options - Accessible Modules

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Accessible Modules** tab to view the list of accessible modules.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

This page displays the full list of the application's function modules. A **Turned Off** indication appears adjacent to a module that is not accessible based on your system's module configuration setup.

> **NOT_SUPPORTED**
>
> Refer to Module Configuration for more information on function modules and how to turn off modules that are not applicable to your organization.

Installation Options - Installed Products

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Installed Products** tab to view a read only summary of the products that are installed in the application version that you are logged into.

Description of Page

The **Product Name** indicates the name of the "products" that are installed. The collection should include **Framework**, an entry for your specific product and an entry for **Customer Release**.

Release ID shows the current release of the application that is installed. This field is used by the system to ensure that the software that executes on your application server is consistent with the release level of the database.

The **Release ID Suffix**, **Build Number** and **Patch Number** further describe the details of your specific product release.

> **Note**
>
> If you have developed implementation-specific transactions, you can populate the release ID, release ID suffix, build number and patch number for the **Customer Release** entry to define the latest version of implementation-specific logic that has been applied to this environment. The framework provides a script (`F1CustRelM`) to allow these fields to be updated manually. In addition, the `F1-InstallationOptions` inbound web service provides operations to view and update those fields via a REST call. Alternatively, your implementation team may populate these fields as part of their upgrade scripts.

The **Product Status** is used to indicate if a product installed by default in the version you are logged into is **Active** or **Inactive**.

The **Display** column indicates the product whose name and release information should be displayed in the title bar. Only one product sets this value to **Yes**.

Owner indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

Product Type indicates if the product is a Parallel Application. A parallel application is one that is independent of, and does not conflict with, other parallel applications. Multiple parallel applications can be installed in the same database and application server.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **About Information**. The information on this tab is used to populate the information displayed in the About window for your product.

Support For Different Languages

- User Language
- Customer Language
- Defining Languages

User Language

The language English (with the locale "en") is provided by the system and all system metadata are delivered with an English description or label. The system provides support for defining other languages and supports multiple languages in a single environment. System users can use the system in their preferred language, as long as a translation into that language has been provided. A user sees the system in the language defined on their user record. If enabled, users can use the Switch Language zone to switch to another supported language real time.

Supporting a New Language

Normally, setting up the system for another language (besides the delivered English language entry) is an implementation issue, not an administrative setup issue. However, there are several online administrative features that are used to set up a new language, and these are described here.

1. **Define a language code and indicate that it is enabled.** For details on this procedure, see Defining Languages.
2. **Install or confirm the existence of a language pack**. For an on-premise installation, if the product supplies a language pack with translations for the system metadata descriptions, follow the instructions for downloading the language pack and installing it into your environment. For a cloud installation, if the product supplies a language pack for a given language, no additional step is needed here. The language pack is already installed.
3. **Copy descriptions of all language-enabled tables from an existing translation (e.g., English).** The copied values act as placeholders while the strings are translated into the new language. It is necessary to do this as a first step in order to create records using the new language code created in the previous step. Language-based descriptions can be copied using a supplied batch process, `F1-LANG`. The batch copies all English labels in the system.

   The batch process will also update the new language rows with the translated metadata descriptions from the language pack, if installed.
4. **Translate additional content.** Translatable descriptions and labels for implementation data may be updated / entered in the application. First the user record must be updated to reference the new language. This may be done in one of the following ways:
   a. Switch to the new language using the Switch Language zone.
   b. If that zone is not available, navigate to the user page, assign the new language code to your User ID, sign out, and sign back in again.

Copyright © 2026, Oracle and/or its affiliates.

Any online functions that you access will use your new language code. You can change the language code for all users who plan to use/modify the new language.

> **Note**
>
> The language pack updates all language entries for base owned system data. If your implementation updates base owned labels and descriptions after applying a language pack, they will be overwritten the next time an updated language pack is applied. Note that most user facing labels and messages support defining an Override Label or Override Description. This information is not updated by the base product and should be utilized if your implementation desires a specific label or description.

Managing Multiple Languages

Once a non-English language has been set up, there are several points to note when adding or updating records that include translatable strings.

- When a new record is added through the user interface or through a REST service call or any other means that goes through the standard maintenance service, the system automatically generates entries for each supported language and copies the text of the entered description to all the generated languages.
- When the translatable string for a record is updated, only the user's language row is updated. The system does not ripple the change to other supported language entries (because it could be overriding a valid translation).
- When your system is upgraded to a new version, the new version typically includes new system data. The upgrade only includes the English language rows. Implementers should run the `F1-LANG` batch program to generate language rows for the your supported languages for this new data. If a new translation pack for your supported language is available at the time of upgrade, then running this process also applies the translated strings. (As mentioned above, for on-premise installations, you need to download and install the language pack before this step). If the language pack is not available at the time of upgrade, then the batch program should be run again when the language pack has been installed.
- New translatable objects may also be included in a patch / hot fix. In this situation, the `F1-LANG` batch program should be run after installing the patch in order to ensure there are language rows for your supported languages. Updated language packs are only provided for full release upgrades and not for hot fixes. So implementations may need to temporarily supply translations for the new strings until you update to the next release with the next updated translation pack.

Customer Language

Your specific product may also support capturing the language of a customer. Such that correspondence sent from the product may be produced in a language set on a customer record. Refer to your specific product documentation for more information about additional language support.

Defining Languages

Your product may support multiple languages. For example, the field labels, input text, and even outputs and reports can be configured to appear in a localized language. A language code for every potential language exists in the system to supply this information in various languages.

Copyright © 2026, Oracle and/or its affiliates.

Select **Admin**, and then **General**, and then **Language** to define a language.

This is a standard All-in-One portal.

When **Language Enable** is checked, the system adds a row for this language whenever a row is added in another language. For example, if you add a new currency code, the system will create language specific record for each language that has been enabled. You would only enable multiple languages if you have users who work in multiple languages. Languages that are configured as enabled, appear in the Switch Language zone.

> **Note**
>
> For on premise applications, the login page displays all the languages that are enabled, allowing the user to toggle the login instructions in that language. The list of enabled languages is captured on the server at startup time. If a new language is enabled, contact your server administrator to refresh the server in order to see the new language displayed in the login page.

The **Locale** is a string containing three portions:

- ISO language code (lower case, required)
- ISO country code (upper case, optional)
- Variant (optional).

Hyphens separate the various portions, and multiple variants are possible.

The locale along with the collator strength control how the contents of grids and search results are sorted by the Java virtual machine (JVM) on your web server: The specific JVM in use by your particular hardware/OS configuration constrains the available **Locales**. Validating the **Locale** against the JVM is outside the scope of this transaction. This means you are responsible for choosing valid **Locales**.

Additionally, external systems that communicate with the product and provide a language use a format that matches the locale. For example in the Accept-Language header for REST web service calls, a requested language is provided.

The following are examples of valid locales:

Locale	Comments
`en-US`	American English
`en-AU`	Australian English
`pt-BR`	Brazilian Portuguese
`fr-FR-EURO`	European French
`ja-JP`	Japanese

The **Collator Strength** parameter is used for the Java collation API . This parameter controls whether, for example, upper and lower-case characters are considered equivalent, or how accented characters are sorted. Valid values for collator strength are **PRIMARY**, **SECONDARY**, **TERTIARY**, and **IDENTICAL**. If you leave this field blank, Java will use its default value for the language. We'd like to stress that the impact of each value depends on the language.

Copyright © 2026, Oracle and/or its affiliates.

Please see https://docs.oracle.com/javase/7/docs/api/java/text/Collator.html for more information about the collator strength for your language.

Display Order indicates if this language is written **Left to Right** or **Right to Left**.

Note that all administrative control tables and system metadata that contain language-specific columns (e.g., a description) reference a language code. In addition, other tables may reference the language as a specific column. For example, on the User record you indicate the preferred language of the user.

Defining Countries

The topics in this section describe how to maintain countries.

- Country - Main
- Country - States

Country - Main

To add or review Country definitions choose **Admin**, and then **General**, and then **Country.**

The **Main** page is used to customize the fields and field descriptions that will be displayed everywhere addresses are used in the system. This ensures that the all addresses conform to the customary address format and conventions of the particular country you have defined.

Description of Page

Enter a unique **Country** and **Description** for the country.

The address fields that appear in the **Main** page are localization options that are used to customize address formats so that they conform to address requirements around the world. By indicating that an address field is **Optional**, you make that field available everywhere addresses for this country are used in the system. You can enter your own descriptions for the labels. These labels will appear wherever addresses are maintained in the system.

> **Note**
>
> Your specific product may also add the ability to mark an address field as **Required**. If that is available, then the product is also supplying appropriate validation in all places where a user defines an address.

> **Note**
>
> For any country where the **State** is enabled, the valid states for the country must be entered on the **Country - State** tab. When entering address constituents on a record that captures this detail, the value for State is verified against the data in the State table. For any country where there is a component of the address that represents a "state" but your implementation does not want to populate the valid states for that country, choose a different field such as County for this constituent (and define an appropriate label). When entering address constituents on a record that captures this detail, no validation is done for the County column.

Copyright © 2026, Oracle and/or its affiliates.

Country - States

To maintain the states located in a country, choose **Admin**, and then **Country**, and then **Search** and navigate to the **State** page.

Description of Page

For any country where you have enabled the State switch, use the **State** collection to define the valid states in the **Country**.

- Enter the standard postal abbreviation for the **State** or province.
- Enter a **Description** for this state or province.

Defining Currency Codes

The currency page allows you to define display options related to currency codes that are used by your system. Use **Admin**, and then **General**, and then **Currency** to define the currency codes in which financial information is denominated.

This is a standard All-in-One portal.

Most attributes are self explanatory.

> **Note**
>
> Please contact your specific product to verify whether it supports a currency with more than 2 **Decimals**.

The **External Reference** may be used to define a currency whose currency code is more than three characters. To support that type of currency, define a currency code that uniquely identifies the currency in the system and use the external reference to define the currency code recognized by external products. Note that your specific integration with an external system where financial information is included may need to be adjusted to use the external reference instead of the system's currency code. Refer to the specific documentation for your integrations to confirm.

Defining Time Zones

The following topics describe how to design and set up time zones.

- Designing Time Zones
- Setting Up Time Zones
- Setting Up Seasonal Time Shift

Designing Time Zones

The Time Zone entity is used to define all the time zones where your customers may operate. Each time zone should define an appropriate Time Zone Name. This is a reference to an external source that defines time zones, their relationship to Greenwich Mean Time, whether the time zone follows any shifting for summer / winter time (daylight savings time) and when this shift occurs.

Copyright © 2026, Oracle and/or its affiliates.

The following sections describe concepts and configuration topics related to managing time zones.

> **Note**
>
> Oracle Utilities Customer Care and Billing - Interval Billing applications customers should consult the topic *Time Issues* (search the Help index for "time issues") for specific information relating to that product's interval billing time related functionality.

The Base Time Zone

When designing your time zones, the first thing to determine is the base time zone. You may choose the time zone where the company's main office resides. Once this is done you can link the time zone code to the installation option as the base time zone. Refer to Installation Options - Main for more information.

> **Note**
>
> An attribute in the system properties file may be configured to indicate that the DB session time zone should be synchronized with the value defined on the Installation Options. Refer to the *Server Administration Guide* for more information.

If your company does business beyond your main office's time zone, define the other time zones where you may have customers or other systems with which you exchange data. At this point, your specific product may include configuration tables to capture default time zones, for example based on a postal code or geographic location.

Standard vs. Legal Time

The term **Legal Time** refers to the actual clock time as may be impacted by daylight saving (also called 'local time'). This time line has a gap hour on entry to the daylight saving period and a duplicate hour on exit from it.

The term **Standard Time** refers to a time line without any daylight saving shifts applied to it. This virtual time line has no gap hour on entry to the daylight saving period nor a duplicate hour on exit, i.e. it is continuous and non-ambiguous. As a general rule, it is recommended that all time sensitive data is stored in the standard time of the base time zone as defined on the installation options. This will prevent any confusion when analyzing data and will ensure that your algorithms do not have to perform any time zone or daylight saving shifting of data that may be stored in different time zones.

Depending on your specific product, entities may have their date/time information stored in either of the following options:

- Standard time of the base time zone.
- Standard time of another time zone related to the entity’s geographical location.
- Legal time of the base time zone.
- Legal time of another time zone related to the entity’s geographical location.

The metadata definition of a table specifies whether any of its date/time fields are stored in standard time or not. While typically all date/time fields of a table that supports standard time would be stored in the same way the system allows each field to specify either of these options.

Copyright © 2026, Oracle and/or its affiliates.

- **Physical Standard Time** indicates that the field is stored in base standard time and is pre-populated by the system. For example, creation time, status update time etc.
- **Logical Standard Time** indicate that the field is stored in base standard time.
- **Referenced Time Zone** indicate that the field is stored in standard time of a time zone related to the entity's geographical location.

Storage vs. Display

Regardless of the time zone option used for storing the data in standard time, all date/time information is entered and displayed on the user interface in the legal time of its respective time zone. If a field is defined to be stored in the standard time of the base or other time zone, the user interface is designed to accept the data in the legal time of the respective time zone and shift it to the standard time of the same time zone before saving the data and performs the reverse shift from standard time back to legal time before displaying the data to the user.

It is important to understand that time zone and daylight saving time conversions to and from standard time are performed behind the scenes as part of the user interface layer. Once data reaches the server it is already in its storage time option. The user should not be aware of these time conversions.

Time shifting between standard and legal times does not happen automatically. When dealing with fields stored in standard time, their data entry and display need to explicitly include time shifting features as part of their user interface and zone configuration.

Date / Time Schema Elements

When defining date / time fields in a business object schema, schema attributes can be used to define whether or not data should be stored in standard or legal time for the base time zone or if it should be stored in the standard time or legal time of another time zone (related to the data).

By default data stored in standard or legal time is displayed in the corresponding time zone's legal time. An additional schema attribute can be used to indicate if the display of the time should be shifted to the legal time of a different time zone. For example, if the data is stored in the base time zone but the data is related to a different time zone, the data will be shown in the time zone appropriate for the data (including the appropriate seasonal adjustment). Refer to Schema Nodes and Attributes- Standard Time Considerations for more information.

Exchanging Date / Time Information With External Systems

Date/time information should be exchanged between systems in standard XSD format as it includes a reference to the time zone of the specified time as an offset from Coordinated Universal Time (UTC). The system automatically converts date/time elements in an inbound message from XSD to internal format and vice versa for an outbound message. The latter is controlled by an explicit date/time format setting on the External System record for the outbound message type.

The offset is determined as followed and it is based on the time zone associated with the element as explicitly defined by the respective inbound or outbound schema:

- Elements defined as stored in the standard time of the base or other time zone would always have the same offset throughout the year as this time line is never shifted for daylight saving. The offset would be the standard offset of that time zone from UTC, i.e. not during daylight saving time.
- Elements stored in the legal time of the base time zone or other time zone would have the standard offset of the their storage time zone when the date/time value is outside the daylight saving period and the shifted offset when it is inside that period of time.

Copyright © 2026, Oracle and/or its affiliates.

User Time Zone

If your company does business in multiple time zones, the user record of each user may reference the time zone of their location. This time zone provides additional information about the location of the user and as such may be used by specific business rules that involve time zone settings.

> **Note**
>
> The time zone defined on the user record is not used to automatically display date/time information in the user's time zone. By default, the system displays date/time fields in the legal time of their respective time zone and not the user's time zone.

Setting Up Time Zones

Refer to Designing Time Zones for background information about defining time zones.

Open this page using **Admin**, and then **General**, and then **Time Zone**, and then **Search** .

This is a standard All-in-One portal.

Refer to the embedded help text for more information on the Time Zone attributes.

Setting Up Seasonal Time Shift

> **Note**
>
> The information in this topic applies only to Oracle Utilities Customer Care and Billing - Interval Billing applications.

Open **Admin**, and then **General**, and then **Seasonal Time Shift**, and then **Search** to define the seasonal time shift schedule.

Description of Page

Enter a unique **Seasonal Time Shift** code and **Description** for the seasonal time shift.

The Collection defines the **Effective Date/Time** (in standard time) that a time zone may shift in and out of standard time. If time is changed from standard time on the effective date/time, enter the **Shift in Minutes** that the time changes from standard time (usually `60`). If the time is changed back to standard time on the effective date/time, enter a **Shift in Minutes** of `0`.

Defining Geographic Types

If your company uses geographic coordinates for dispatching or geographic information system integration, you need to setup a geographic (coordinate) type for each type of geographic coordinate you capture on your premises and/or service points (geographic coordinates can be defined on both premises and service points).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Product specific.** There is no framework functionality that uses this information. Refer to your specific product documentation to verify how this table is used in your specific product. In addition, you can determine if this object is a foreign key on any tables specific to your product. Navigate to the Table page in the application, search for and select the Geographic Type table and use the Relationships tab to view this information.

To define geographic types, open **Admin**, and then **Geographic**, and then **Geographic Type.**

This is a standard All-in-One portal.

The **Validation Format Algorithm**, if specified, is used to validate that the geographic location entered on the premise and/or service point for the geographic type is in the format as defined in the algorithm. If you require validation, you must set up this algorithm in the system.

Defining Work Calendar

This portal is used to view and maintain information about a work calendar. Work calendars are used to ensure system-calculated dates fall on a workday.

You can access the portal from **Admin**, and then **General**, and then **Work Calendar**. You are brought to a query portal with options for searching for a specific calendar. Once a calendar has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones will appear as part of the portal's **Main** tab page

- **Work Calendar.** This zone provides information about the selected calendar.
- **Work Calendar Holidays.** This zone lists the holidays observed by your organization for this calendar. Use the zone actions to add, maintain or duplicate a work holiday entry.

Defining Display Profiles

When you set up your users, you reference a display profile. A user's display profile controls how dates, times, and numbers displayed. Choose **Admin**, and then **General**, and then **Display Profile**, and then **Search** to maintain display profiles.

Description of Page

Enter a unique **Display Profile** and **Description** to identify the profile.

Enter a **Date Format**. This affects how users view dates and how entered dates are parsed. The following table highlights standard supported date mnemonics and what is displayed at runtime.

Mnemonic	Comments
dd	Day of the month.
d	Day of the month, suppressing the leading 0.
MM	Month number.
M	Month number, suppressing the leading 0.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Comments
yyyy	The 4-digit year.
yy	The 2-digit year.
y	Allows entry in either 2 or 4-digit form and is displayed in 2-digit form.

Other characters are displayed as entered. Typically, these other characters should be separators, such as "-", ".", or "/". Separators are optional; a blank space cannot be use.

Examples:

Configuration Format	Sample Output
MM-dd-yyyy	04-09-2001
d/M/yyyy	9/4/2001
yy.MM.dd	01.04.09
MM-dd-y	04-09-01 - In this case you could also enter the date as 04-09-2001

> **Note**
>
> For centuries, the default pivot for 2-digit years is 80. Entry of a 2-digit year greater than or equal to 80 results in the year being interpreted as 19xx. Entry of a 2-digit year less than 80 results in the year being interpreted as 20xx.

In addition, the following date localization functionality is supported. Note that in every case, the date is stored in the database using the Gregorian format. The settings below result in a conversion of the date for the user interface.

- **Hijri Dates**

  Entering **iiii** for the year is interpreted as a year entered and displayed in Hijri format. For example, the Gregorian date 2014–05–30 may be entered / displayed as 1435/07/30 for a user whose display profile date format is iiii/MM/dd. Note that this functionality relies on date mapping to be defined in the Hijri to Gregorian Date Mapping master configuration. entry. Refer to Additional Hijri Date Configuration for more information.

- **Taiwanese Dates**

  Entering **tttt** for the year is interpreted as a year entered and displayed in Taiwanese format where year 1911 is considered year 0000. For example, if the Gregorian date is 01-01-2005, it is displayed as 01-01-0094 for a user whose display profile date format is dd-mm-tttt.

- **Japanese Dates**

  There are two options available for configuring Japanese Era date support. The setting **Gyy** for the year is interpreted as a year entered and displayed using an English character for the era followed by the era number. The letter 'T' is used for dates that fall within the *Taisho* era. The letter 'S' is used for dates that fall within the *Showa* era. The letter 'H' is used for dates that fall within the *Heisei* era. The letter 'R' is used for dates that fall within the *Reiwa* era. For example, for a user whose display profile date format is Gyy/mm/dd the Gregorian date 2008/01/01 is shown as H20/01/01; the Gregorian date 1986/03/15 is

Copyright © 2026, Oracle and/or its affiliates.

shown as S61/03/15. The setting **GGGGyy** is interpreted as a year entered and displayed using Japanese characters for the era followed by the era number.

Japanese date limitations are as follows:

- The years 1912 through the current date are supported.
- Any functionality that displays Month and Year does not support Japanese Era dates. These dates are shown in Gregorian format.
- Graphs that display dates do not support the **GGGGyy** format.

Enter a **Time Format**. The following table highlights standard supported date mnemonics.

Mnemonic	Comments
hh	The hour 1-12.
h	The hour 1-12, suppressing the leading 0.
HH	The hour 0-23.
H	The hour 0-23, suppressing the leading 0.
KK	The hour 0-11.
K	The hour 0-11, suppressing the leading 0.
kk	The hour 1-24.
k	The hour 1-24, suppressing the leading 0.
mm	Minutes.
m	Minutes, suppressing the leading 0.
ss	Seconds.
s	Seconds, suppressing the leading 0.
a	Indicates to include **am** or **pm**. This is only needed for 12 hour formats, not 24 hour formats. (**hh**, h, KK, K). If an **am** or **pm** is not entered, it defaults to **am**.

Examples:

Configuration Format	Sample Output
hh:mma	09:34PM (can be entered as 09:34p)
hh:mm:ss	21:34:00
h:m:s	9:34:0

There are several options for displaying Numbers.

Decimal Symbol defines the separator between the integer and decimal parts of a number. Valid values are "." (a period) or "," (a comma).

Group Symbol defines the means to separate groups of bigger numbers. Valid values are as follows:

- A comma (","). Large numbers group by threes separated by a comma, for example 1,000,000.
- A period ("."). Large numbers group by threes separated by a period, for example 1.000.000.
- **None**. Large numbers do not have any separator, for example 1000000.

Copyright © 2026, Oracle and/or its affiliates.

- **South Asian**. This option uses a comma for its separator but will group large numbers as follows: the first comma is used for the thousands separation and numbers over 9,999 are grouped with 2 units, for example 10,00,000.
- **Space**. Large numbers group by threes separated by a space, for example 1 000 000.

Negative Format defines how negative values are displayed. Valid values are `-9.9`, `(9.9)`, or `9.9-`.

Currency values can have a different **Negative Format** from other numbers. Valid values are `-S9.9`, `(S9.9)`, or `S9.9-`, where the "S" represents the currency symbol.

- Additional Hijri Date Configuration

Additional Hijri Date Configuration

For implementations that wish to support displaying dates according to the Hijri calendar, besides appropriate configuration in the Display Profile, the mapping between the Hijri dates and the Gregorian dates must be entered. This mapping is defined in the **Hijri to Gregorian Date Mapping** master configuration record.

The mapping record contains a collection of entries for each year in the Islamic calendar.

For each year, clicking the Expand Zone icon shows the mapping collection with the first date of each month of the Hijri calendar. The corresponding date in the Gregorian calendar should be entered for each row.

Defining Phone Types

Phone types define the format for entering and displaying phone numbers.

To add or review phone types, choose **Admin**, and then **General**, and then **Phone Type.**

This is a standard All-in-One portal.

The **Phone Number Format Algorithm** for each phone type controls the format for entry and display of phone numbers.

Use **Phone Type Flag** is used to identify phone types that may have specific functionality associated with it. For example if your product had support for sending a facsimile transmittal, identify the phone type that captures a **Fax** number.

Defining Feature Configurations

Some system features are configured by populating options on a feature configuration.

You can access the portal from the **Admin Menu**, and then **General**, and then **Feature Configuration**. You are brought to a query portal with options for searching for a specific feature configuration. Once a feature configuration has been selected you are brought to the maintenance portal to view and maintain the selected record.

Because various options throughout the system may be controlled by settings in feature configuration, this section does not document all the disparate possible options. The administration documentation for each feature describes its options and whether an option supports multiple values. You may also refer to the detailed description of each option type when reviewing a configuration online for more information.

You can create options to control features that you develop for your implementation. To do this:

Copyright © 2026, Oracle and/or its affiliates.

- Review the lookup values for the lookup field EXT_SYS_TYP_FLG. If your new option can be logically categorized within an existing feature type, note the lookup value. If your new option warrants a new feature type, add a lookup value to this lookup field.
- Define the feature's option types. If you have identified an existing feature type to add the options to, find the lookup with the name xxxx_OPT_TYP_FLG where **xxxx** is the lookup value of EXT_SYS_TYP_FLG noted above. If you decided to create a new feature type (by adding a new lookup value to the EXT_SYS_TYP_FLG lookup, you must create a new lookup with the name xxxx_OPT_TYP_FLG where **xxxx** is the new value you defined above.
- Flush all caches.

The following zones may appear as part of the portal's **Main** tab page.

- **Feature Configuration.** This zone provides information about the selected feature configuration.
- **Options.** This zone is used to maintain various options associated with the feature type.
- **External System Messages.** If the feature exists to interface with an external system, you can use this zone to define the mapping between error and warning codes in the external system and our system. The zone is visible for feature types that their definition in the **Feature Type Configuration** Extendable Lookup supports this functionality.

> **Note**
>
> A corresponding message must be defined in the system message tables using a Message Category that is 90000 or greater so as to avoid collisions during upgrades.

Defining Master Configurations

A master configuration is an object that enables an implementation to define configuration for features in the system. It is an alternative to using feature configuration for defining options. A master configuration is defined using a business object. Typically one master configuration exists for a given business object yet some configurations may be designed to support multiple records.

Overview / Initial Setup

The product provides one or more master configuration that may be used for configuration. Some examples of base master configuration business objects are as follows

- **Hijri to Gregorian Date Mapping**. This allows an implementation that uses Hijri dates to define the mapping between those dates and Gregorian dates.
- **ILM Configuration.** For implementations that use Information Lifecycle Management, the ILM configuration record defines some parameters used by the process.
- **Migration Assistant Configuration.** For implementations that use the content migration assistant (CMA), the configuration record defines some parameters used by the process.

For a list of all the master configuration records provided by the product, navigate to the master configuration page in the application. To find help topics related to functionality controlled by the master configuration records, use the keyword ‘master configuration’ in the index.

Implementations may configure the business objects to limit the ones that are visible to users if desired.

Copyright © 2026, Oracle and/or its affiliates.

- If the master configuration does not apply to your implementation, navigate to the business object for this master configuration and update the Instance Control to **Do not allow new instances**. This ensures that the business object will not be visible to any user for the implementation.
- If a given master configuration record is only relevant for certain users in the system, application security may be used to limit the records visible by a given user. By default the base delivered business objects typically reference the Master Configuration MO application service. This may be overridden to link custom application services to the master configuration BOs to provide more granular security. Users will only see the master configurations that they have security for.

Configuration

You can access the portal from the **Admin**, and then **General**, and then **Master Configuration**. You are brought to a query portal with options for searching for a master configuration business object. If a configuration has not yet been set up for the business object use the add icon to define the record. If one or more configuration records exists for the business object you may select the master configuration business object. Once selected you are brought to this portal to view the selected record.

The following zones may appear as part of the portal's **Main** tab page.

- **Master Configuration** This zone provides information about the selected master configuration record. Please see the zone's help text for information about this zone's fields.
- **Related Configurations** This zone lists additional master configuration records for the same business object, representing different parts of the same configuration. You may add additional parts from this zone. The zone is only visible when the business object is designed to support multiple configuration parts and is explicitly marked as such by the **Multiple Configurations** business option.

What's New Portal

> **Important**
>
> This is only available for cloud implementations.

The What's New portal provides a list of maintenance pack fixes that have been applied to your environment along with a summary of the maintenance packs that have been supplied along with their dates. There is also a link for Release Notes. Clicking this link opens the Cloud Application Readiness page. Select the link corresponding to your product and version to see the new features added in your version.

To view the information, open the Help Menu in the toolbar and choose the What's New option.

The following zones appear on the Main page.

- **Maintenance Pack Details**. This zone is populated whenever a maintenance pack or a hot fix is applied to your environment. By default it shows fixes applied in your current maintenance pack and it suppresses copies if your environment already received this fix in a previous release. Use the filters to expand the search to other releases or to include copies. There are also filters to narrow the search based on various criteria. Refer to the zone inline help for more information.

Copyright © 2026, Oracle and/or its affiliates.

- **Installation History Summary**. This zone is visible if you click the 'View Installation Summary' link in the previous zone. It shows a distinct list of installation date and version information.

> **Note**
>
> This zone only has fixes that were applied to your environment after the enhancement to populate this data was fully supported. The system is not populating this zone with fixes applied to your implementation prior to the enhancement.

Security & User Options

The contents of this section describe how to maintain a user's access rights.

- Application Security
- Row Security
- Data Privacy
- Cryptography Keys

Application Security

The system restricts access to various components of the application using application services. The following sections provide more information about this mechanism.

- Understanding Application Security
- Defining Application Services
- Defining Security Types
- Defining User Groups
- Defining Users
- Application Security Query

Understanding Application Security

The contents of this section and supporting topics provide information about application security.

Users

Each user who is granted access to the application is stored in the User object. This object defines the attributes of the user including identifier, name, language, time zone, home page and display profile (which defines the preferred format of dates and amounts). In addition there is configuration related to using the product, including favorite links and scripts, portal preferences and valid To Do roles - controlling what types of To Do entries a user can work on. Additionally, Refer to Users and User Groups for additional information.

Copyright © 2026, Oracle and/or its affiliates.

Application Services

Each object in the system that may be secured is associated with an application service. For example, each page in the system defines its application service allowing you to control what users can view a given page.

An application service defines one or more Acces Modes, which typically correspond to an action associated with the application service. This allows you to control users that can only view a type of record (and grant security to the Inquire access mode) and users that can perform other actions such as add, change and delete. Refer to Secured Objects and Application Servicesto understand what objects are associated with an application service and the typical access modes associated with each type of secured object.
Application Services may also define additional configuration, called security types. Refer to Field Level Security for more information.

User Groups

User Groups are a mechanism for grouping users, usually around job roles. Users are attached to User Groups through a relationship that is effective-dated. This can be useful for the attachment of temporary employees such as contractors or for people who change roles regularly.

Each User Group is authorized to access appropriate Application Services and their access modes. In addition, user groups define access to the application service's security types, if applicable. Refer to Field Level Security for more information.

- Users and User Groups
- Secured Objects and Application Services
- Field Level Security
- Importing Security Configuration from an External Source

Users and User Groups

A user group defines the application services and access modes its users have access to. A user may be associated to more than one user group. The user groups may have overlapping security configurations and the security access for a user is the union of all security granted by all of its user groups.

A major effort for implementations is to decide how to define their user groups. User groups can be created to represent roles or other job-related categorization; or based on organizational factors; or a combination of both. Refer to Secured Objects and Application Services for some guidelines on understanding what application services secure and tools provided to help security administrators identify what application services to grant access to.

The following sections provide additional topics related to users and user groups, including some base functionality related to these objects as well as some best practices.

Base Delivered User / User Groups

The system is delivered with a base User linked to a base User Group, provided to facilitate an initial user accessing the application for an on-premises implementation.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Cloud implementations should refer to the Cloud Services Administrative Guide for information on setting up users.

The system supplied user is **SYSUSER**. This user is delivered so that when an on-premises environment is provisioned, there is at least one user authorized to use the system. It has access to the base delivered user groups. An initial administrator can sign in as this user and proceed to create the actual users that should get access to the environment.

> **Important**
>
> Once appropriate users have been defined, it is recommended to update SYSUSER to set it to disabled. (In cloud implementations, this user is delivered as disabled.)

The product provides several base user groups that serve as a kind of library of services. They are meant for documentation purposes and as a way to easily access the system in a development or testing environment. Some user groups are meant to define a list of services associated with a distinct set of features or functions in the system. The user group `ALL_SERVICES` is a catch-all user group that defines services that are not part of a specific use case. (Note that some services may be defined in ALL_SERVICES as well as another specific user group). You cannot change or remove the information delivered for the base user groups. System administrators should define user groups appropriate for their business requirements.

> **Important**
>
> It is not recommended for any production user to have access to the base delivered user groups that are provided as a library of services. The product often adds or changes records in the base user groups for a new release when there is new or changed functionality. It means that your production users would have a change in what they have access to without the ability to opt in to new features.

Defining a Read-Only User Group

The product does not supply a user group that is 'read only', but it is a common requirement for implementations. This section provides a guideline for creating such a user group.

1. Create the User Group with the desired name and description.
2. Navigate to the Service Manager tab. In the Add Application Service / Access Modes tab, you can use the filter to limit the results to those with Inquire access mode. Click the checkbox in the header to select all rows and click Add. You are prompted for an expiration date. Click Save. (Note that the zone has a limit to 1000 services so you may need to do this more than once to get all services added.)
3. In addition, you should give this user group access to all the access modes for the default application service: `F1-DFLTS` (Default Application Service). This application service is considered one that all users should have for basic access to the system.

Copyright © 2026, Oracle and/or its affiliates.

Template User

The system provides support for defining one or more users as a template user. There are two broad use cases for using a template user.

Inheriting Preferences or Favorites Configuration

Some implementations want all users of a certain team to have the same configuration of portal preferences and / or favorite links. This allows users in that team to have the same information in the same place on their respective displays, facilitating in helping each other with questions.

If your organization wants to implement this practice, define a template user with the desired portal preferences and / or favorite links. You may define different users with different configurations for more granularity.

For each user whose preferences or links should be taken from the template user, go to the User record via **Admin**, and then **Security**, and then **User.** If the portal preferences should be inherited from a template user, define the desired template user as the **Portals Profile User** . If the favorite links should be inherited from a template user, define the desired template user as the **Favorites Profile User**.

If applied, the specified user's portal preferences and / or favorite links will be those defined on the template user and they will be unable to customize those preferences.

Defining a Template User to help in User Creation

The user object in the product captures configuration related to many aspects of using the product. It includes application security configuration, such as its user groups; row level security configuration through its access groups; the To Do roles that define what types of To Do entries a user can work on, along with other configuration. Many users that are in a similar 'role' in the organization would have many of the same settings. As such, it may be useful for implementations to define a template user that represents a 'role' in the organization, with much of the configuration common for users in that role. When creating a new user, you can use the template user as the starting point and then duplicate that record to create the new user. Then you only need to adjust the information that is unique for the newly created user.

Many implementations provision their users first in an external identity management system and use an interface to import the user definition into the product. Refer to Importing Security Configuration from an External Source for more information. For these interfaces, the external system may have the ability to define custom information relevant to the product's user record, but the product also supports referencing a template user so that information from the template user is copied to the new user while creating the record.

How can you refer to a template user when creating a User from and external source?

- When the new user is uploaded to the system, the interface uses the user BO `F1-IDMUser` to create the user. The BO includes a pre-processing algorithm. This algorithm looks for the existence of a template user in the schema of the new user sent as a characteristic of type `F1-TMUSR`. If such a characteristic is found, it uses that template user. If not found, the algorithm supports defining a template user as an algorithm parameter. If a template user is identified, all the information from the template user will be copied onto the new user record except for the information passed in from the external source. The template user is captured on the newly created user via a characteristic for information / audit purposes.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bookmarks** are not included in the data that is copied from a template user.

- Once the new user is created, its configuration can now be adjusted, if applicable. Note that the template user is only a tool used when adding a user. Updates to the template user will not "ripple" to all the other users that were created based on this template.
- Not all external sources may have the ability to define a template user and send it as a characteristic or via another field. For these integrations, using a single template user on the algorithm as a parameter may be used to define basic configuration applicable to a large percentage of new users. (More information on this below). For those integrations that are able to associate a new user with a template user in the external system and provide that template user along with the new user information allows you to associate the new user with a template user that is more aligned with the user's role. Refer to Importing Security Configuration from an External Source for more information on how each integration supports the definition of a template user.

Designing your Template Users for helping Create Users

Before configuring template users, all the administrative control tables that are part of User configuration must be defined, including time zones, display profiles, To Do roles, data access roles and user groups.

If your integration will use a single template user defined as an algorithm parameter, you need to define a template user with the basic information that is applicable to all or at least most users. See below for guidance on referencing this template user as an algorithm parameter.

If the way you create users allows you to associate the user with a more specific template user, the next step is to define the user configuration for your system users. During this exercise, you will find that you have broad categories of users. But you will also see that within a given category of user there may be variations in the user privileges and preferences. For example, perhaps there are supervisors within the Customer Service role that have more security privileges than a typical Customer Service user. In addition, there may be variations based on the attributes of the users themselves. For example, maybe your organization exists in multiple time zones and some of your workers are in one time zone and some are in the other.

At this point your security users that are designing their user provisioning procedures must decide the following:

- What information about a new user will be captured in the identity system (besides the expected information like Name, Email and the User IDs)? For example, for the case of multiple time zones, maybe the best solution is to capture the time zone when defining the user.

  What information is defined on the template user and how many template users should be created to reduce the need for manual steps or additional data captured in the identity management system? In the case of multiple time zones, proliferating the template users to have one set for one time zone and another set for the other time zone may not make sense since this is one field that is different. However it may be reasonable to create additional templates in the case of striations in the levels of privileges for workers of a different category. So rather than template users for Dispatcher, Mobile Worker, System Administrator and Contractor, your organization may have template users for Dispatcher, Mobile Worker, Mobile Worker (Supervisor), System Administrator, Contractor (Short Term) and Contractor (Long Term).

  What information is must be configured on the User record in the application after the user is added? If only a small number of users have a variation from other users, it may be that

Copyright © 2026, Oracle and/or its affiliates.

the easiest way to deal with those variations is to simply update those user records manually. Using the above examples:

- If your organization covers 2 time zones but only a small group of people work in one of the time zones whereas the bulk of the users are in the other time zone, the simplest procedure may be to define the template users for the main time zone and use that for the creation of all users. Then for the small group of users in the separate time zone, navigate to the User page to adjust the time zone after the record is added.
- If only a small number of Mobile Workers are supervisors with separate privileges, rather than defining a special template user for those type of workers, the simpler procedure may be to use the Mobile Worker template and then navigate to the User page to add the additional privileges to the supervisor users after the record is added.

Defining a Template User

Regardless of which use case is applicable for your organization, the procedure for defining a template user is the same.

Navigate using **Admin**, and then **Security**, and then **User.**

- Define a User ID. If you are using a template user to help create users from an external system and the external system is able to capture the template user to send in with the new user information, this is the ID that should be referenced in the external system.
- Be sure to choose a **User Type** of **Template User**.
- Define all the information appropriate based on the use case.
  - For template users used as a Portal Profile user, be sure to define the appropriate portal profile information.
  - For template users used as a Favorites Profile user, be sure to define the appropriate favorite links.
  - For template users acting as a type of 'role' to help with creation of users, define all the information that should be copied onto a new user that references this user as its template user. (Note that **Bookmarks** are not included in the data that is copied from a template user.)

Defining a Default Template User for Integrations

Define the default template user on the algorithm that defaults data from this user as follows:

- Navigate using **Admin**, and then **System**, and then **Algorithm.**
- Search for and select the algorithm `F1-OIMUSR` (Populate User Data from a "Template" User).
- Edit the record. Add a new row in the Algorithm Versions collection. Enter an appropriate date (later than the date of the existing entry, but on or before the current date). Populate the template user you want to use as the default.

Secured Objects and Application Services

The system restricts access to transactions or explicit services using an application service. The following sections provide more information about what objects are secured, how to find out what is secured by each app service as well as additional information related to managing application services.

Copyright © 2026, Oracle and/or its affiliates.

What is Secured by an Application Service

- The following points highlight security related to viewing and modifying individual records in the system:
  - All maintenance objects define an application service that includes the basic actions available, typically **Add**, **Change**, **Delete**, and **Inquire**. The base product supplies an application service for every maintenance object. Some objects that include a fixed lifecycle, may define additional access modes on the maintenance object's application service. The application service for the maintenance object is defined on its related service program.

    > ⚠ **Caution**
    >
    > Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a maintenance object that includes access modes **Freeze**, **Complete** and **Cancel**). If you want to give a user access to any of these special actions, you must also give the user access to the **Change** action.

  - For maintenance objects whose user interface page is not portal-based, the application service also controls whether the menu entry appears. If a user doesn't have access to the maintenance object's application service, the menu item that corresponds with the application service will not be visible.
  - For portal based user interfaces, each main (stand-alone) portal defines an explicit application service with the access mode **Inquire**, allowing the user interface to be secured independently of the underlying object security. If a user doesn't have access to the portal's application service, the menu item that corresponds with the application service will not be visible. The base product supplies an application service for every portal that is accessible from the menu. Note that the application service for the portal is defined on its related service program, which is derived via its navigation option and navigation key.
  - Menu items may define an application service / access mode. Typically the security supplied for portals and maintenance objects provides enough granularity to suppress menu items that a user does not have access to. In addition, by default, the system populates application services / access modes on menu items related to a BPA script. A common use case is the 'Add' menu item for maintaining an object. In this case, typically the application service for the maintenance object is used along with the **Add** access mode. Other types of BPA related menu items may have other appropriate configuration.

    Clients may choose to linking an explicit application service / access mode to further suppress a menu item. For example, imagine your implementation creates a special BPA script to add a To Do Entry and would like users to use the special BPA rather than the base supplied Add dialogue for To Do Entry. The base delivered menu item for adding a To Do Entry uses the To Do Entry maintenance object application service with the Add access. Users need that security setting to be able to add a To Do Entry even from a different dialogue. To suppress the base Add dialogue, override the setting for the base supplied menu item for To Do Entry Add with a special application service and access mode. Then define a menu entry for the new special BPA for adding.
  - Zones define an application service.

Copyright © 2026, Oracle and/or its affiliates.

        * For zones linked to a portal, if a user doesn't have access to the zone's application service, the zone will not be visible on the portal. In most cases the zone is delivered with the same application service as its portal. In special cases, such as the zones on the Sidebar, the product supplies separate application services for each zone allowing implementations to determine at a more granular level which users should have access to which zones.
        * For query zones that are configured on a multi-query zone, if a user doesn't have access to the zone's application service, the zone will not be visible in the dropdown on the multi-query zone. In most cases all zones in a multi-query zone define the same application service as the multi-query zone. The product may supply a special application service for one or more zones in a multi-query zone if the functionality is special to certain markets or jurisdictions and not applicable to all implementations.
        * For zones that are used by business services to perform SQL queries, the product supplies a default application service (`F1-DFLTS`). Security for these zones is not checked by the product as they are used for internal purposes.
    - For portal based pages, individual elements may be shown or hidden based on security using the **oraHasSecurity** function in a UI map's HTML or in UI Hints in a schema. Refer to Check User's Security Access for more information.
    - Business objects define an application service. If the business object defines a lifecycle, the application service must include access modes that correspond to each state. In addition, the standard maintenance object access modes of **Add**, **Change**, **Delete** and **Inquire** are included. The base product business objects are supplied with appropriate application services. In addition, implementations may override the configured application service if desired.

      > ⚠ **Caution**
      >
      > Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a business object that includes lifecycle states **Freeze**, **Complete** and **Cancel**). If you want to give a user access to any of these special actions, you must also give the user access to the **Change** action.

    - Batch controls define an application service which provides the ability to secure submission of individual batch processes. The application service must include an access mode of **Execute**. The base product batch controls are supplied with appropriate application services. These services will typically have an ID that matches the batch control ID.
    - Report Definition records define an application service. The application service must include an access mode of **Submit / View Report**.
- The following objects are securable but are typically executed via internal processes. The security is provided to ensure that any access to the objects from an external source is secured.
    - BPA scripts may define an application service with the access mode **Execute**. The base BPA scripts are typically not configured with any application service. An implementation may define one. Note that as mentioned above, a menu item may also be configured with an application service and access mode. This allows for a BPA that is invoked via a menu entry to be secured in more than one way.

Copyright © 2026, Oracle and/or its affiliates.

- Business Services and Service Scripts define an application service with the access mode **Execute**. This is needed for services that may be executed from an external system, for example via an inbound web service. Base business services and service scripts that are linked to an inbound web service are configured with special application service. All other business services and service scripts are delivered with a default application service (`F1-DFLTS`), which may be overridden by an implementation.
- Scripts may also check security within an Edit Data step type if there is some functionality that should or should not execute based on a user's security access. The Edit Data step would include a call to the business service `F1-CheckApplicationSecurity`
- Service Programs define an application service. As mentioned above, for Portals and Maintenance Objects, their application service is taken from the related service program. In base, specific application services are released for each of these types of service programs. All other service programs are typically delivered with a default application service (`F1-DFLTS`), which may be overridden by an implementation. Note that for service programs linked to a Business Service, the application service on the business service takes precedence when invoking the business service.

Determining What an Application Service Secures

Because the product wants to provide as much granularity as possible so that implementations can have the greatest control over what users can do, there are a large number of application services provided out of the box and it can be daunting to understand who needs to have access to what application services to do their job. The product provides several online tools to help security administrators to understand what application services exist and what they are for.

- On the application service page itself, there is a Secured Objects zone that indicates all the metadata that is related to this application service. There are a small number of application services that are securing functionality that is not discoverable by the Secured Objects zone. For those services, the detailed description of the application service should indicate what it's for.
- A large percentage of security is related to a specific page in the system that is accessible via the menu. Because of the different layers of security, to be able to view a record you may need access to several application services: the one associated with the query portal, the one associated with the maintenance portal, the one associated with the maintenance object of the record and the one associated with the business object governing the record. When viewing a menu, there is an Application Security tab, which allows you to review each menu line and see a list of all the application services a user may need access to for viewing or maintaining data on that related page.
- The Application Security Query portal provides various filters for viewing what objects are being secured by what services. Besides menu items, it shows batch controls, reports, web services and other functionality. In addition, it allows you to view what users and what user groups currently have access to what application services and allows you to compare the security of two users.
- The User Group portal includes several zones that help security administrators to configure security options or review security configuration. The main tab includes a zone to Add Application Services. This zone only shows application services that are not already linked and provides several ways to narrow down the list of application services. The Service Manager tab has a similar zone, but breaks down the information to a combination of application service and access mode. Finally, the Secured Components tab is similar to the Application Security Query. It allows you to query the objects secured by application services in this user group.

Copyright © 2026, Oracle and/or its affiliates.

Application Services Introduced in a Release

When new functionality is provided that includes any object that is associated with an application service, new application services may be introduced. The product release notes indicate the new application services included in the release. In addition, the product includes a special Entity Tag that includes a list of the new application services. The tag's description uses the cloud release reference rather than the on-premise release reference to identify itself.

Field Level Security

Sometimes transaction and action security is not sufficient. There are situations where you may need to restrict access based on the values of data. For example, in Oracle Utilities Customer Care and Billing you might want to prevent certain users from completing a bill for more than $10,000. This is referred to as "field level security".

Field level security can be complex and idiosyncratic. Implementing field level security always requires some programming by your implementation group. This programming involves the introduction of the specific field-level logic into the respective application service(s). The recommendation is to use a business object validation algorithm for checking field level security.

> **Note**
>
> The check for field level security must exist in validation logic to ensure that data is not added or updated if it does not pass the field level security rules.
>
> It is not straight forward to implement user interface changes related to field level security, such as protecting elements under certain conditions. This is because overriding the base product delivered user interface to check for implementation specific conditions is not always supported, depending on how the user interface is built, or requires a full duplication of product artifacts, which is not recommended.

The definition of a user's access rights is performed using the same transactions used to define transaction / action level security. This is achieved as follows:

- Create a security type for each type of field-level security.
- Define the various access levels for each security type. For example, assume you have some users who can complete bills for less than $300, and other users who can complete bills for less than $1,000, and still other users who can complete bills for any value. In this scenario, you'd need 3 access levels on this security type:
  - Level 1 (lowest): May authorize bills <= $300
  - Level 2 (medium): May authorize bills <= $1,000
  - Level 3 (highest): May authorize all bills
- Link this security type to each application service where this type of field level security is implemented. This linkage is performed on the security type transaction.
- Defining each user group's access level for each security type (this is done for each application service on which the security type is applicable).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Highest value grants highest security. The system expects the highest authorization level value to represent highest security level. Moreover, authorization level is an alphanumeric field so care should be taken to ensure that it's set up correctly.

Importing Security Configuration from an External Source

The product provides support for importing security information from an external source. In all cases, the product uses a special user BO `F1-IDMUser` to create the user and the recommendation is to take advantage of the Template Userfunctionality. More information about the out of the box support for using this feature is described in the related links.

- If your organization uses Lightweight Directory Access Protocol (LDAP), you can import your existing LDAP users and groups into the system. Once imported, all user and group functions are available. You can import a user group, or a single user. You can resynchronize your LDAP users and groups at any time.

  > **NOT_SUPPORTED**
  >
  > For more information refer to LDAP Integration.

- The system provides an integration with Oracle Identity Manager. When a user is created in the identity manager product, its information can automatically be interfaced to the product. Once the user is successfully created in the system, all functions are available.

  > **NOT_SUPPORTED**
  >
  > For more information refer to Oracle Identity Manager Integration.

- The system provides a web service with add, update and get operations that follow the System for Cross-domain Identity Management (SCIM) specification for provisioning users.

  > **NOT_SUPPORTED**
  >
  > For more information refer to Provision Users with SCIM Open Standard.

Defining Application Services

Please refer to Understanding Application Security for a description of how application services are used when you grant user groups access rights to transactions and the significance of an application service's access modes.

You can access the portal using **Admin**, and then **Security**, and then **Application Service.** You are brought to a query portal with options for searching for a specific application service. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- The **Application Service** zone provides the information about the application service and its access modes.
- The **Secured Objects** zone displays the object (or objects) that are secured by this application service. Refer to Secured Objects and Application Services for details about the types of objects that may be secured.

The **Application Security** tab includes zones to view and manage which user groups have access to the application service and which ones do not have access .The following zone may appear as part of this tab page:

- The **User Groups Linked** zone lists the user groups that currently have a link to the application service. Note that expired links are included when the zone is first displayed. Use the zone filters to exclude expired links and to limit the user groups shown by user ID, user group ID, and user group description.
- The **User Groups not Linked** zone lists the user groups that do not have a link to the application service. Use the zone filters to limit the user groups shown by user ID, user group ID, and user group description.

Defining Security Types

Security types are used to define the types of field level security.

> **Note**
>
> **Programming is required.** You cannot have field level security without introducing programming code. Refer to Field Level Security for more information on how security types are used to define field level security.

- Security Type - Main

Security Type - Main

Select **Admin**, and then **Security**, and then **Security Type** to define your security types.

Description of Page

Enter a unique **Security Type** and **Description**.

Use the **Authorization Level** grid to define the different authorization levels recognized for this security type. Enter an **Authorization Level Number** and its **Description**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Programming is required.** Note that the values that you enter are not interpreted by the system itself, but by the user exit code used to implement the special security. Check with the developer of the user exit logic for the correct values. Refer to Field Level Security for more information on how security types are used to define field level security.

Use the **Application Services** grid to define the application service(s) to which this security type is applicable. If this application service is already associated with user groups, you must update each user group to define their respective security level. This is performed on the User Group portal.

Defining User Groups

A user group is a group of users who have the same degree of security access. Think of a user group as a "role"; associated with a role are:

- The users who play this role
- The application services to which the role's users have access (along with the actions they can execute for each service and their field level security authorization levels).

Refer to Secured Objects and Application Services and Field Level Security for more information about access modes and security types.

> **Caution**
>
> Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a transaction that supports actions in addition to **Add**, **Change**, and **Inquire** (e.g., **Freeze**, **Complete**, **Cancel**). If you want to give a user access to any of these additional actions, you must also give the user access to the **Inquire** and **Change** actions.

This portal allows a security administrator to view and manage information about a user group.

You can access the portal from the **Admin Menu**, and then **Security**, and then **User Group**. You are brought to a query portal with options for searching for a specific user group. Once a user group has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Application services may not be changed or removed from any Base Delivered User / User Groups.

The following zones may appear as part of the portal's **Main** tab page

- **User Group.** This zone provides information about the selected user group.

Copyright © 2026, Oracle and/or its affiliates.

- **Application Services.** This zone lists application services linked to the selected user group. You may use this zone to edit the information for an specific application service or use mass actions to remove selected services or update their effective date.
- **Add Application Services.** You may use this zone to add application services to the selected user group.

The following zones may appear as part of the portal's **Users** tab page

- **Users.** This zone lists users linked to the selected user group. You may use mass actions to remove selected users or update their effective date.
- **Add Users.** You may use this zone to add users to the selected user group.

The following zones may appear as part of the portal's **Service Manager** tab page

- **Application Service / Access Modes.** This zone lists application service and access mode combinations currently linked to the selected user group. You may use mass action to remove a selected set of records.
- **Add Application Service / Access Modes.** You may use this zone to add application service access mode records.
- **Security Type Authorization.** This zone maintains field level security for application services included in the selected user group.

The following zone may appear as part of the portal's **Secured Components** tab page

- **Application Components.** This zone allows you to review various types of application components secured by the selected user group. The query is similar to the Application Security Query restricted to the current user group.

Defining Users

The user maintenance transaction is used to define a user's user groups, data access roles, portal preferences, default values, and To Do roles. To access the user maintenance transaction, select **Admin**, and then **Security**, and then **User**.

The user maintenance transaction is the same transaction invoked when the user launches Preferences from the user menu or shortcut key.

Application Security Query

This portal allows a security administrator to review application security configurations across users, user groups and the various types of application components they secure.

While application security is configured at the granular level of an application service and its access modes, it is more intuitive to review access rights to top level broader components:

- **Menu line.** Several application services may be related to a single menu line and impact the ability to properly leverage various aspects of its functionality:
  - The application service of the transaction page or portal, which also controls the visibility of the menu line or its 'search' part when the line is composed of 'add' and 'search' items.
  - The application services associated with any zones on that portal.
  - The application service of the underlying maintenance object when applicable, which also controls the 'add' item of the menu line.
  - The application services of business objects for the underlying maintenance object.

Copyright © 2026, Oracle and/or its affiliates.

  - The application services of portals used to maintain these business objects.
  - The application services of related portals.
  - The application services of related maintenance objects, their business objects and portals.
  - and more

  You may review the list of application services associated with a menu line on the **Application Security** tab on the **Menu** portal.

- **Sidebar Zone.** This is a regular or context sensitive zone that the user has access to on the sidebar area.
- **System.** This is a Toolbar or other system wide option a user is allowed to use.
- **Batch Control.** This is a batch process a user is allowed to submit.
- **Report Definition.** This is a report a user is allowed to submit.
- **Web Service.** This is a web service a user is allowed to invoke.
- **Miscellaneous.** This is a special service that a user is allowed to invoke.
- **Internal.** This is an internal item that is secured by default but is assumed to be called in a context that does not require security check.
- **Unclassified.** Some application services may not be related to any type of application component and as such remain unclassified. For example, there are situations where previously released services are enhanced to be associated with dedicated application service in a later release. In this situation the services may retain their original or even custom application service and the new application services may appear as not associated with any component.

Refer to Secured Objects and Application Services for details about the types of objects that may be secured.

Open this page using **Admin**, and then **Security**, and then **Application Security Query**.

Row Security

Some products allow you to limit a user's access to specific rows. For example, in Oracle Utilities Customer Care and Billing, row level security prevents users without appropriate rights from accessing specific accounts.

A combination of framework configuration and configuration in your edge product is required for row level security. The following points describe the configuration:

- For each record that should be secured, associate it with an **Access Group**. Note that if your edge product supports row level security, that product is providing a link between the secure-able record and Access Group. Your access groups may be granular and only referenced by one secured record or they may be more broad and be referenced by multiple secured records that require the same type of security restriction.
- To define which users have access to the secured records, you define a **Data Access Role**. For each data access role, define which Access Groups the role has security clearance for. An access group may be linked to one or more data access roles. In addition, define the **Users** that have access rights to these secured records. When you grant a data access role rights to an access group, you are giving all users in the data access role rights to all secured records in all the referenced access groups. A user may belong to many data access roles.

Copyright © 2026, Oracle and/or its affiliates.

If your edge product supports row level security, it will include logic in the appropriate areas of the system to limit the secured rows that a user may view or maintain based on this configuration. For example, in Oracle Utilities Customer Care and Billing, throughout the system users are only able to view and maintain information about an account and any of its detail if the user is in a Data Access Role for the account's Access Group (or the account is not linked to an Access Group).

> **NOT_SUPPORTED**
>
> Refer to your product's documentation for more information on row level security, if applicable.

- Defining Access Groups
- Defining Data Access Roles

Defining Access Groups

Refer to The Big Picture of Row Security for a description of how access groups are used to restrict access to specific objects.

Access groups control which groups of users (referred to as Data Access Roles) have rights to accounts (or other objects) associated with the access group. You can access the portal using **Admin**, and then **Security**, and then **Access Group**. You are brought to a query portal with options for searching for a specific access group. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Access Mode** zone provides the information about the access group and its data access roles.

- The **Data Access Role** collection defines the data access roles whose users have access to the access group's secured objects. Keep in mind that when you add a **Data Access Role** to an **Access Group**, you are granting all users who belong to this role access to all of the objects linked to the access group.

  > **Note**
  >
  > You can also use Data Access Role - Access Group to maintain a data access role's access groups.

- The **To Do Role Assignment** collection may be used to route To Do Entries of a given To Do Type to a specific To Do Role if the objects related to the To Do Entry are associated with the access group. Refer to Adjusting the To Do Role for more information.

Defining Data Access Roles

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of Row Security for a description of how access groups are use to restrict access to specific objects.

The data access role transaction is used to define two things:

Copyright © 2026, Oracle and/or its affiliates.

- The users who belong to the data access role.
- The access groups whose accounts (or other objects) may be accessed by these users.
- Data Access Role - Main
- Data Access Role - Access Group

Data Access Role - Main

Select **Admin**, and then **Security**, and then **Data Access Role** to define the users who belong to a data access role.

Description of Page

Enter a unique **Data Access Role** code and **Description** for the data access role.

The scroll area contains the **Users** who belong to this role. A user's data access roles play a part in determining the accounts (or other objects) whose data they can access.

To add additional users to this data access role, press the add button and specify the following:

- Enter the **User** ID. Keep in mind that when you add a **User** to a **Data Access Role**, you are granting this user access to all of the accounts (or other objects) linked to the data access role's access groups.
- Use **Expiration Date** to define when the user's membership in this data access role expires.

> **Note**
>
> **Also maintained on the user page.** You can also use User - Access Security to maintain a user's membership in data access roles.

Data Access Role - Access Group

Select **Admin**, and then **Security**, and then **Data Access Role** and navigate to the **Access Groups** tab to define the access groups whose accounts (or other objects) may be accessed by the users in this data access role.

Description of Page

Use the **Access Group** collection to define the access groups whose objects can be accessed by this role's users. Keep in mind that when you add an **Access Group** to a **Data Access Role**, you are granting all users who belong to this role access to all of the accounts (or other objects) linked to the access groups.

> **Note**
>
> You can also use Access Group - Main to maintain an access group's data access roles.

Data Privacy

Personally Identifiable Information (PII) is any information that identifies or could be used in combination with other information to identify an individual. Many countries have data privacy

Copyright © 2026, Oracle and/or its affiliates.

laws governing the use of such information. A key aspect of these laws is an organization's obligation to erase personal information when there is no compelling reason to retain it. This obligation can be fulfilled either by deleting data or altering the data in such a way that it is no longer possible to associate the data with the individual.

Organizations are likely to have other legal obligations that impact how long they retain personal data. For example, there could be financial audit requirements that oblige the organization to retain relevant information for a fixed number of years after termination of a contract.

Object Erasure is designed to address right to erasure issues, with a combination of configuration and processes that may be extended to implement the appropriate policies for the PII within your data.

- Encryption and Masking
- The Approach to Implementing Object Erasure
- Data Redaction for Export and Logging
- Defining Redaction Rules
- Obfuscation Configuration

Encryption and Masking

"Encryption" refers to encrypting data stored in the database using an encryption key. There are two different types of encryption described in the sections below. System encryption refers to columns in the system identified by the product to use encryption. Application encryption refers to the ability for an implementation to configure fields and elements that should be encrypted in the database.

"Masking" refers to overwriting all or part of an un-encrypted field value with a masking character. For example, perhaps only the last 4 digits of a tax ID number are visible with the other digits changed to an asterisk. The system provides support for masking fields on the user interface that may be stored as plain text in the database. In addition, there are cases where encrypted fields are shown to the user interface using masked values rather than the encrypted value.

The following sections provide more information about each feature.

- System Encryption
- User Interface Masking
- Application Encryption

System Encryption

The system automatically encrypts certain fields captured in various option tables or context tables. This is mainly used for passwords. For example, passwords captured in Message Sender context or password

In addition, batch control supports configuring a security option for parameters that capture sensitive information, such as a password. Refer to Defining Batch Controls for more information.

It is also possible to enable system encryption using the characteristic type `F1-PWD`. However, the maintenance object must include specific code to enable system encryption for characteristics of this type. In Oracle Utilities Application Framework, the only maintenance

Copyright © 2026, Oracle and/or its affiliates.

object that supports this is extendable lookup. Refer to Extendable Lookup Advanced Topics for more information.

User Interface Masking

The functionality described in this section is used to take data that is stored in plain text in the database and mask the value before it is presented to a user (or an external system). This feature includes the ability to allow some users to view the data unmasked using security configuration. The system allows different masking rules to be applied to different fields. For example, a credit card number can be masked differently than a social security number.

The following topics describe how to mask field values.

Identify the Data to be Masked

Identify the data that is stored as plain text, but should be masked for display to users. For example, imagine that you have identified that Credit Card Numbers and a person's federal ID number (for example, in the United States, the Social Security Number or SSN). Each field identified may be displayed and maintained in different user interfaces throughout the system, but the masking rules for a given field are probably uniform regardless of where the data is displayed.

Primary keys cannot be masked. A field defined as a unique identifier of a row cannot be configured for masking. Masking a field that is part of the primary key causes a problem when attempting to update the record. This restriction also applies to elements that are part of a "list" in an XML column on a maintenance object. One or more elements in the list must be defined as a primary identifier of the list. Be sure that primary key elements in the list are not ones that require masking.

List members that contain different "types". Consider a page with a list that contains a person's identification numbers. You can set up the system so that a person's social security number has different masking rules than their drivers license number. If your implementation has this type of requirement, the list of masked fields should contain an entry for each masking rule.

For each field, if there are some users that may see the data unmasked on one or more of the user interfaces, then security configuration is required. If the value of a field should be masked for all users across all pages in the application, then the security configuration is not needed.

Security Configuration

Define a security type for each field with two authorization levels:

- 1 - Can only see the element masked
- 2 - Can only see the element unmasked

Link all of the security types to an application service of your choosing. We recommend linking every masking-oriented security type to a single application service (e.g., `CM_MASK`) as it makes granting access easier.

For each security type, identify which users can see its data unmasked and which users can only see its data masked. If the masked and unmasked users fit into existing user groups, no additional user groups are necessary. Otherwise, create new user groups for the masked and unmasked users.

After the user groups for each security type are defined, link each user group to the application service defined above. When a user group is linked to the application service, you will define the authorization level for each security type linked to the application service. If a user group's

Copyright © 2026, Oracle and/or its affiliates.

users should see the security type's field values unmasked, set the authorization level to 2; otherwise set it to 1.

> **Note**
>
> Updating masked values. The system supports allowing a user that sees a masked value to update the value. For numbers that are masked, the masked value is typically all 9s. If there is some chance that a user is allowed to update a masked number, the recommendation is that the user is configured with authorization to see the number unmasked. Otherwise, there is a corner case where the user sees the masked number and needs to update the number to a value that exactly matches the masked value. In this case, the update won't go through because the system doesn't recognize it as a change.

> **Note**
>
> Flush the cache. Remember that any time you change access rights you should flush the security cache (by entering flushAll.jsp on the URL of the application) if you want the change to take effect immediately.

Configure a Masking Algorithm

A data masking algorithm (using the algorithm entity value of **Feature Configuration - Data Masking**) must be created for each combination of masking rules and security type. These algorithms determine if a user has the rights to view a given field unmasked, and, if not, how the field should be masked.

The base package provides two algorithm types. Both support configuration to allow a set of users to see the data unmasked. Parameters capture the application service, security type and authorization level defined above used to evaluate this.

- `F1-MASK`: this is for masking alphanumeric data. Its parameters are designed to handle most masking needs. Parameters allow you to configure how much of the data to mask and what masking character to use. Refer to the algorithm type description for more information.
- `F1-MASKNBR`: this is for masking numbers. Refer to the algorithm type description for more information.

Determine How the Fields are Displayed

The masking configuration differs based on how a field is retrieved for access to the user interface. So for the masking of one "logical" field (like a person's SSN), there may be multiple configuration entries required to cover all the access methods. Review each user interface where a given field is displayed and create the following categories:

- The field is an element that is retrieved by invoking a business object, a business service, or a service script
- The field is stored as an ad hoc characteristic
- The field is displayed on a fixed maintenance page (and is therefore retrieved by invoking a page service)
- The field is displayed on a fixed search page (and is therefore retrieved by invoking a search service)

Copyright © 2026, Oracle and/or its affiliates.

Create a Feature Configuration for Each Masked Element

Create a feature configuration with a Feature Type of **Data Masking**. An option entry with option type of **Field Masking** is needed for every combination of field to mask and the method used to display the data. The value will contain mnemonics that reference the appropriate data masking algorithm along with configuration that differs depending on how the field is retrieved for display as described below.

Schema Based Object Field Masking

For data that is accessed via a schema-based object call and displayed in a UI map, the field to be masked must reference a meta-data field name in its schema definition: `field="fld_name", alg="algorithm name"`

If the element references an mdField in the schema, that is the field used to identify the masking rule. If there is no mdField reference but only a mapField reference, that is the field used to identify the masking rule. For example, if you want to mask a credit card number, let's assume that field is defined in the schema is the following:

```
<creditCard mdField="CCNBR" mapField="EXT_ACCT_ID"/>
```

In this case, the option value should be `field="CCNBR", alg="algorithm name"`. An option value of `field="EXT_ACCT_ID", alg="algorithm name"` would not result in masking.

A "where" clause may also be specified. This is useful for data that resides in a list where only data of a certain type needs to be masked: `field="fld_name", alg="algorithm name", where="fld_name='value'"`

For example, person can have a collection of IDs and only IDs of type 'SSN' (social security number) should be masked. If the person data including its collection of person IDs is displayed on a UI map via a business object call, let's assume the collection is defined in the following way:

```
<personIDs type="list" mapChild=CI_PER_ID">
  <isPrimaryId mapField="PRIM_SW"/>
  <idType mapField="ID_TYPE_CD"/>
  <personIdNumber mapField="PER_ID_NBR"/>
</personIds>
```

The option value may look like this: `field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`

Please note the following important points for schema based masking:

- **Limitation of 'where' field** Although the main use of a 'where' clause for schema oriented elements is to mask certain elements in a list based on a 'type', it is also possible to mask a single field in the schema based on the value of another field. For example, imagine that a customer submits a registration form that defines an ID type and ID value. Although this data is not in a list, the implementation may still want to only mask the ID value if the ID type is "SSN". The framework is only able to mask an element in the schema based on a 'where' clause if the element in the 'where' clause is a "sibling" in the schema.
  - If the element to be masked is in a list, the element in the 'where' clause must be in the same list.

Copyright © 2026, Oracle and/or its affiliates.

- If an element to be masked maps to a real column in a table, the element in the 'where' clause must also map to a real column in the table.
- If an element to be masked maps to and XML column in the table as a single element, the element in the 'where' clause must map to the same XML column as a single element.

- **Multiple feature option entries for the same field.** It's possible that different schemas in the system have a similar type of data that may be masked based on different conditions. For example, imagine that an implementation has different schemas that captured or referenced person identifiers in different ways:
  - One schema captures a single person ID without any corresponding "type" record and it should always be masked using Algorithm CM_SSN_MASK:

    ```
    <personSSN mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  - One schema captures a person ID and a corresponding ID Type and it should be masked with Algorithm CM_SSN_MASK if the type is "SSN" and masked with algorithm CM_FEIN_MASK if the type is "FEIN".

    ```
    <personIdType mapXML=BO_DATA_AREA mdField=ID_TYPE_CD/>
    <personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  - One schema captures a person ID and a corresponding ID Type and it has the same masking rules as the previous schema, but a different field name is used for the ID Type code. (This scenario could happen if for example a different label is desired for ID Type on the user interface for this schema.)

    ```
    <personIdType mapXML=BO_DATA_AREA mdField=CM_ID_TYPE/>
    <personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  For this scenario, the feature options may look like this:

  1. `field="PER_ID_NBR", alg="CM_SSN_MASK"`
  2. `field="PER_ID_NBR", alg="CM_SSN_MASK", where="ID_TYPE_CD='SSN'"`
  3. `field="PER_ID_NBR", alg="CM_FEIN_MASK", where="ID_TYPE_CD='FEIN'"`
  4. `field="PER_ID_NBR", alg="CM_SSN_MASK", where="CM_ID_TYPE='SSN'"`
  5. `field="PER_ID_NBR", alg="CM_FEIN_MASK", where="CM_ID_TYPE='FEIN'"`

  For each schema, the system will first find whether the element applies to any masking option. It will find 5 masking options for the field PER_ID_NBR. Then it will determine if any sibling elements match the 'where' clause.

  - If more than one sibling element matches a 'where' clause, a runtime error is issued. For example if a schema has an element that references "mdField=ID_TYPE_CD" and an element that references "mdField=CM_ID_TYPE", this is an error. Additionally, if multiple elements reference mdField=ID_TYPE_CD", this is an error.
  - If one and only one sibling element matches a 'where' clause, the value of the element is compared to the values defined in the 'where' clause. If it finds a match on the value, the appropriate masking algorithm is applied. If no match is found (for example, the Person ID Type is "LICENSE") the element is displayed as is.

Copyright © 2026, Oracle and/or its affiliates.

- If no sibling element matches a 'where' clause and a feature option exists with no 'where' clause (option 1 above), then the masking algorithm of the option with no 'where' clause is applied.

- **Changing the value in the 'where' clause.** If your implementation has some users that are allowed to change records where some data is masked based on a condition, it is recommended to design the user interface to reset the masked value when the value in the 'where' clause changes. For example, if a user is prevented from viewing a person's social security number, but the user is allowed to make updates to the person's record, changing the value of the Person ID Type should reset the Person ID Number. This would ensure that the user does not 'unmask' the social security number by simply changing the ID Type.

Records Maintained Using Page Maintenance

For data that is accessed via a page maintenance service call, indicate the table name and the field name where the data resides: `table="table_name", field="fld_name", alg="algorithm name"`

For example if the Person record and its collection of identifiers are displayed and maintained using page maintenance, the option value should be `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name"`

A "where" clause may also be specified: `table="table_name", field="fld_name", where="fld_name='value'", alg="algorithm name"`

This is useful for data that resides in a child table where only data of a certain type needs to be masked. For the person ID example, `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`

Characteristic Data

For data that is stored as a characteristic, simply indicate the characteristic type: **CHAR_TYPE_CD='char type', alg="algorithm name"**

This needs to be defined only once regardless of which characteristic entity the char type may reside on. Note that only Ad hoc characteristics are supported.

Masking Fields in Explorer Zones or Info Strings

In explorer zones data is often retrieved using SQL directly from the database. No masking is applied automatically in this case. If there is data in the explorer zone results that should be masked, the masking must be applied by calling a business service.

Similarly, an MO Info algorithm may not use BO interaction to get data. It may access data using SQL for efficiency purposes. No masking in applied when retrieving data via SQL. To apply masking to a string prior to including it in an info string, the masking must be applied by calling a business service.

The system supplies two business services to be called to determine if masking rules apply for a specific field.

- `F1-TableFieldMask`. Mask a Table field. This business service receives a table name, field name and one or more field values. If masking applies it returns the masked value.
- `F1-SchemaFieldMask`. Mask a Schema field. This business service receives a schema name and type, XPath and field value. If masking applies it returns the masked value.

Search Service Results

For data that is displayed on a 'fixed' search page, it is retrieved via a search service call. Indicate the search name and the appropriate field to mask along with the masking algorithm.

Copyright © 2026, Oracle and/or its affiliates.

For example: `search="SearchServiceName", field="PER_ID_NBR", where="ID_TYPE_CD='SSN'", alg="algorithm name"`

To find the name of the search service, launch the search in question, right click in the filter area and choose **Inspect**. Search for "serviceName". The service name is listed there. To find the field name to mask, search for "Widget Info". There should be two results found, one for the filter area and one for the results area. This results area has the text "SEARCH_RESULTS" as prefix for each field. The field names are the ones after the x$. Do not reference the x$ when defining the field name. Note, the "where" statement can only apply to fields that are also part of the search results.

Additional Masking Information

The following points provide additional information to assist in your masking configuration:

- If the demonstration database includes a **Data Masking** feature configuration, review the settings because it will probably contain masking rules that will match your own.
- On data input pages, a user might be able to enter or change masked data, such as a bank account number, but not be able to subsequently see what they added or changed.
- External systems can request information by performing a service call via a web service. Please keep in mind that some web service requests require data to be masked and some do not. For example, a request from an external system to synchronize person information needs the person's social security number unmasked; whereas a request from a web self service application to retrieve the same person information for display purposes needs the person's social security number masked. To implement this type of requirement, different users must be associated with each of the requests and these users must belong to separate user groups with different access rights.
- If a maintenance object (MO) contains a field that holds an XML document and a service call invokes the MO's service program directly, the system will mask individual XML elements in the field if a **Determine BO** algorithm has been plugged into the maintenance object and the element(s) in the respective BO schema have been secured as described above.

Application Encryption

The functionality described in this section allows implementations to configure fields to encrypt when storing it in the database. This functionality is mutually exclusive from the User Interface Masking functionality. This feature also supports encrypting specific elements stored within a CLOB or XML column.

The following points highlight the features of the encryption functionality:

- The encryption key is defined using a key ring. Refer to Field Encryption for information about defining encryption key rings.
- When a field is configured to be encrypted, the encrypted data is stored in a special encryption field that is not the source field (the one exposed to the user on the user interface). The source field captures the data as masked. Because a special field is required to support encryption, the product must provide support for that field to be encrypted.
- For encrypted data that must allow searching, the system supports capturing a hash value in a special field. The product must provide support for this functionality. Besides providing a special field to capture the hash value, base search functionality for that data must also cater for this configuration.

Copyright © 2026, Oracle and/or its affiliates.

- The system supports encrypting data that is captured as an element within an XML field. If the XML field is provided in a schema owned by the product, then the product must provide specific support for the capture of the encrypted data.
- For backward compatibility, the system also supports encryption keys and hash keys defined using a keystore. This is not the recommended method going forward, but further information about defining the keystore in the system can be found in the Installation Guide. Refer to Moving from Keystore to Key Ring for information about adopting the key ring functionality if you are currently using the keystore.

The following sections provide additional information about the support for encryption provided by the framework. Refer to the security chapter of the administration guide for your particular product for more information.

Encrypting and Masking the Data

When a product enables encryption for a given field, a special encryption field should be created to capture the encrypted value. Because encrypting is optional, the source field (the one exposed to the user) should not be this special encrypted field. If encryption is configured, the system will internally populate the encrypted field. The source field will be populated with asterisks by default. That way the masked data is what is shown to the user on the page rather than the encrypted value.

The following points highlight how the system behaves when encryption is configured and when it is not. Assume as an example, the field is a credit card number. The user views and populates a field with the field name CC_NBR. The table also has a second field ENCR_CC_NBR. A user populates the credit card number:

- If encryption is not configured, CC_NBR will be updated with the entered credit card number and ENCR_CC_NBR will be empty. Note that in this case, an implementation may choose to configure user interface masking.
- If encryption is configured, CC_NBR will be updated with '**********' and ENCR_CC_NBR will contain the encrypted value. The asterisks for the standard field will fill the full field size up to 50 characters.

If for some reason the standard masking using all asterisks is not desired, the system supports supplying an explicit masking algorithm using the same Feature Configuration - Data Masking plug-in spot used for User Interface Masking.

> **Warning**
>
> Unlike user interface masking, the masking of encrypted fields is not driven by security. The data stored in the source field for all encrypted data should be masked. Be sure not to configure security authorization logic in algorithms used for this type of masking.

Feature Option Configuration

Create a feature configuration with a Feature Type of **Encryption**. For each source field you are encrypting, enter an option with option type of **Field Encryption**. The value will contain mnemonics that reference the appropriate encryption key ring along with configuration related to the field and its table location. Unlike the user interface data masking, the configuration for data encryption is related to how the data is stored rather than how it is displayed. In addition, each entry may define an explicit masking algorithm to override the default and if supported, may also define a hash field and hash key ring.

Refer to Field Encryption for information about defining encryption key rings.

Copyright © 2026, Oracle and/or its affiliates.

For data that is stored in a specific column on a table, an explicit field to capture the encrypted value must exist. Indicate the table name, source field name and encrypted field name along with the key ring: **table='table_name', field='fld_name', encryptedField='encr_fld_name', keyRing='key ring code'**

A "where" clause may also be specified when data resides in a child table and only data of a certain type needs to be encrypted.

Example, **table='CI_PER_ID', field='PER_ID_NBR', encryptedField='ENCR_PER_ID_NBR', keyRing='CM-SymmetricKey', where='ID_TYPE_CD='SSN''**

For data that is stored in an XML column in a record, the source field to be encrypted must reference a meta-data field name in its schema definition along with the element that captures the encrypted data and the alias: **field='field_name', encryptedField='encr_field_name', keyRing='key ring code'**

The syntax for adding a reference to a masking algorithm is **maskAlg='algorithm name'** .

The syntax for adding configuration for capturing a hash value for searching purposes is **hashKeyRing='key ring code' hashField='HASH_FLD_NAME'**.

The following is an example of configuration that uses all the possible options (specific masking algorithm, where clause and hash field support):

table='CI_PER_ID', field='PER_ID_NBR', keyRing='CM-SymmetricKey', encryptedField='ENCR_PER_ID_NBR', hashKeyRing='CM-HashKey' hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN', maskAlg='CM-PERIDMASK'
For backward compatiblity, the product supports defining an encryption key using a keystore. For implementations that continue to use this configuration, you reference the alias or hash alias instead of the key ring. The following is an example of the configuration that uses all the possible options, but uses the alias instead of a key ring.

table='CI_PER_ID', field='PER_ID_NBR', alias='aliasKey', encryptedField='ENCR_PER_ID_NBR', hashAlias='hashAliasKey' hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN', maskAlg='CM-PERIDMASK'
In addition, when using an alias, the encrypted field may optionally be "wrapped" with a special marker (e.g., ENC{}) to indicate that the field value is encrypted. This can be specified by the setting wrap="true" or wrap="false". The default is false. This should be set to false unless additional processing in your code is included to handle the special marker.

Refer to Moving from Keystore to Key Ring for information about adopting the key ring functionality if you are currently using the keystore.

Searching by an Encrypted Value

If the product supports a hashed value for an encrypted field for searching purposes, the following points highlight explorer zone configuration for this purpose

- The user filter value should reference the source field and should include an additional `encrypt=` mnemonic. For example

  ```
  type=STRING
  label=PER_ID_NBR
  encrypt=[CI_PER_ID,PER_ID_NBR,ID_TYPE_CD,F1]
  ```

  Refer to User Filters for more information.
- The SQL should include the hashed value in the WHERE clause. Note that because encryption is optional, a product zone that includes searching by a field eligible for

Copyright © 2026, Oracle and/or its affiliates.

encryption will include finding a match for the filter in the source field (as plain text) or in the hashed field. For example:

```
WHERE
   [(F2) (ID.PER_ID_NBR =:F2 OR ID.HASH_PER_ID_NBR = :F2)]
```

Customizing Encryption Algorithm

This feature is only applicable to implementations that use the keystore. This is not the recommended configuration going forward. The text is kept here for information purposes.

Although the encryption algorithm to use with a given key can be gleaned from the key in the keystore, there is sometimes extra information associated with an algorithm that might need to be used to encrypt or decrypt data.

The system provides a feature configuration option for the **Encryption** feature type using the option type **Algorithm Info** that can be used to adjust the behavior of the encryption.

- You can modify the default mode and padding of the encryption algorithm.
- If a key will be used to digitally sign anything, the signing algorithm can also be specified for the key.

For details about the syntax, refer to the feature option type's detailed description.

- Managing Encrypted Data

Managing Encrypted Data

The sections in this topic provide information about encrypting data when first implementing the application along with various use cases that may require re-encrypting data, such as key rotation.

The batch job `F1-FLENC` (Field Encryption for a Maintenance Object) is provided to handle encryption / re-encryption of all data for a given Maintenance Object. Review your encryption configuration and identify all the maitnenance objects that should have data encrypted. Run the batch once for each maintenance object. The below sections highlight various use cases that warrant running this batch for each maintenance object.

Initially Encrypting Data

Once your encryption / hash configuration is complete, any data added or updated from that point on is encrypted / hashed per the configuration. However, for all existing records that have data that should be encrypted / hashed, you need to run the batch job `F1-FLENC` for each maintenance object to go through and encrypt / hash the data in the existing rows.

Mass Key Rotation

If you introduce a new encryption key or hash key, as described in Key Rotation, or you introduced a new key ring as described in Key Ring Rotation, records added or updated from that point on will be re-encrypted / re-hashed with the latest key for the respective key ring. However, to rotate the key for all existing records that are encrypted / hashed with the previous key, you need to run the batch job `F1-FLENC` for each maintenance object to go through and re-encrypt or re-hash the data in the existing rows.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can independently choose to introduce a new encryption key or a new hash key. The batch program will check each encrypted field or hashed field to confirm that they use the current active key and if not, the data is adjusted.

> **Note**
>
> When rotating a Hash Key, from the time the new key is activated until the time when all records have been successfully updated to use the new hash key, if any end users attempt to search for records via a hashed field, there may be inconsistent results. You should schedule rotation of hash keys to be during a time where end users are not impacted. Ideally you activate the new key and run the key rotation batch control for all maintenance objects during off hours.

Moving from Keystore to Key Ring

Using the key ring for defining your encryption and hash keys allows for key rotation. If your implementation has been using a keystore for field encryption / hashing and would like to use the key ring instead, you can do this as described in the following steps.

- Define a key ring for the Symmetric Key and a key ring for the Hash Key, if applicable.
- Generate and activate a key for each key ring.
- Go to the Feature Option Configuration. For each entry, add references to keyRing / hashKeyRing to define the newly created key rings. Be sure to keep the configuration for the alias mnemonic in place. The system needs this information to know how to unencrypt the existing data if it was encrypted using the keystore. Note that the system doesn't need the hashAlias if you are moving from using the keystore for the hash key to using the key ring. The system simply applies the new hash key.
- Records added or updated from that point on will be re-encrypted / re-hashed using the indicated key ring and its active key. To rotate the key for all existing records that are encrypted / hashed with the keystore alias, you need to run the batch job `F1-FLENC` for each maintenance object to go through and re-encrypt or re-hash the data in the existing rows.
- Once you have confirmed that all the data is using the key ring for encryption and hashing, you can update the feature option configuration to remove the reference to the old alias keys.

The "notes" listed in the Mass Key Rotation section also apply to this use case.

The Approach to Implementing Object Erasure

This section describes the product approach to implementing object erasure for its maintenance objects (MOs).

> **Note**
>
> The approach outlined is intended for managing erasure of master data objects. The assumption is that Information Lifecycle Management will be used for archiving any related transactional data.

Copyright © 2026, Oracle and/or its affiliates.

Various events in the system may signal the need to erase data at some future date. For example, closing an account may indicate the need to erase both the account and person data once there is no longer a need to maintain related financial details. There may also be events that re-establish the business relationship with a person and signal that data marked for erasure should now be retained.

The Object Erasure Schedule is used to capture key details of objects with data that needs to be erased. Records in the schedule are monitored periodically to determine if the date for erasure has been reached and if so, execute the erasure logic. The system will create and maintain records in the schedule when specific events occur for an MO that is marked for erasure.

The system provides the ability to mark an MO as eligible for erasure via a number of MO configuration entries. Included in these is an option that defines the business object for the MO's erasure schedule records. In addition to defining the erasure record elements, this BO defines the monitoring process and the specific erasure logic for the MO.

The following sections provide more information about configuring and managing object erasure.

- Configuring a Maintenance Object for Erasure
- Manage Erasure Schedule Algorithm
- Monitoring the Schedule and Performing Erasure
- Erasing User Information By Obfuscation
- Viewing an Object’s Erasure Status
- Viewing Erasure Configuration
- Archiving the Object Erasure Schedule

Configuring a Maintenance Object for Erasure

There are three types of maintenance object configuration entries that define how to manage erasure:

- An option to define the business object for erasure schedule records.
- An option to define the period between the date on which the need for erasure is detected and the date on which erasure should occur. The common logic provided by the system to add or update erasure records uses this option to determine the erasure date.
- One or more entries in the MO algorithm collection for the algorithms that determine whether a record should be scheduled for erasure, according to the current state of the record, and create or update the erasure schedule entries.

Some MOs that would commonly be regarded as eligible for erasure will be installed with a base Erasure BO already configured. The expectation is that the corresponding Erasure Period option and the algorithm for the Manage Erasure Schedule plug-in spot will be set up by your implementation with the appropriate entries for your organization's business rules.

> **Note**
>
> The presence of an Erasure Period option is the signal to the system that your implementation has enabled an object for erasure.

Copyright © 2026, Oracle and/or its affiliates.

It may not be necessary for all master data objects that contain personally identifiable information to manage their own erasure schedule. For example, the event which triggers erasure for a person record may also trigger the creation of erasure schedule entries for the person's other master data records. In this case, only the person maintenance object will need to be configured with a Manage Erasure algorithm (or algorithms). Refer to your product specific documentation for more information on the recommended approaches for your product's master data objects.

The portal Maintenance Object Configuration includes a zone that shows all the maintenance objects eligible for object erasure. You may use this zone to review or change the configuration for any of the maintenance objects.

Manage Erasure Schedule Algorithm

Algorithms of this type are responsible for determining a record's erasure status and creating or maintaining an entry in the Object Erasure Schedule for the record. They are triggered when certain events occur in the system for maintenance objects that are eligible for erasure. Multiple algorithms can be configured for the plug-in spot.

Depending on the determination made by the algorithm (or algorithms), a number of different actions may need to be taken. For example:

- A new schedule record may need to be added
- An existing schedule record may need to be updated with a new erasure date
- A pending record may need to be discarded if erasure is no longer applicable
- A discarded record may need to be reactivated if erasure now applies again

The base business service `F1-ManageErasureSchedule` performs the common logic to handle the possible actions and is recommended for use by these types of algorithms. The service can also optionally create a log entry for information purposes. Refer to the description of the business service for full details.

In certain circumstances, if one record is scheduled for erasure, other related records may need to be evaluated. The base business service `F1-ManageErasureScheduleDriver` performs the logic to execute an MO's Manage Erasure Schedule algorithms. It can be used to manage any related records from within another algorithm.

This type of algorithm is plugged into the Maintenance Object — Algorithm collection.

Your product may supply algorithms to manage the schedule for base MOs that are regarded as eligible for erasure. If your organization has special business rules that are evaluated to determine the erasure schedule for an MO, a custom algorithm can be created and applied by the implementation team.

Monitoring the Schedule and Performing Erasure

The erasure schedule is managed using the lifecycle of the erasure record's business object. A deferred monitor process is used to periodically check for records that are due for erasure and transition them from the 'pending erasure' state to the 'erased' state. An enter plug in on the 'erased' state is responsible for performing the erasure logic.

The system provides a 'root' business object for the object erasure schedule (`F1-ErasureScheduleRoot`) which defines the lifecycle that erasure schedule business objects should follow. The system also provides a monitor process (`F1-OESMN`) which is configured on the 'pending erasure' state of the 'root' business object. This process is configured to monitor object erasure schedule records whose erasure date is on or before the batch business date.

Copyright © 2026, Oracle and/or its affiliates.

Maintenance objects that are eligible for erasure should be configured with an erasure schedule business object that uses the ‘root’ business object as its parent. The algorithm that performs the erasure processing applicable to that maintenance object must be plugged in on the ‘erased’ state of the ‘child’ business object.

The system provides an erasure schedule business object (`F1-ErasureScheduleCommon`) which is designed to erase an object by deleting the main record and any child records. Your product may supply additional erasure schedule business objects for certain use cases. Refer to your product specific documentation for more information.

Erasing User Information By Obfuscation

Some implementations consider the base User object to have personally identifiable information. Erasing user information by deleting the records is not advisable as it can cause referential integrity problems. The recommended approach is to obfuscate the data instead.

The system provides a user erasure business object (`F1-ErasureScheduleUser`) which is designed to support obfuscating a user’s identifiable information by removing it or replacing it with a non-identifiable value if the field is required.

If the User maintenance object is configured to be eligible for erasure in your implementation, the manage erasure schedule algorithm for the record will be invoked whenever the **Enable** flag is changed for a record on the User maintenance page. The system provides a base algorithm (`F1-OBJERSUSR`) specifically for the User MO. This algorithm will add an entry to the object erasure schedule if the user record is disabled or deactivate an existing erasure schedule record if a disabled user record is enabled again.

Viewing an Object’s Erasure Status

The system provides a portal for viewing and editing an erasure schedule record. The portal is accessed from the **Object Erasure Schedule** sidebar zone. This is a context sensitive zone which only appears when accessing a record for a maintenance object that is configured to be eligible for erasure. It displays the information for the erasure schedule entry for the record in context, if applicable.

To add the zone to an eligible maintenance object:

- Navigate to Context Sensitive Zone and search for the navigation key for the maintenance object.
- Add the Object Erasure Schedule zone `F1-OBJESRSCD` to that navigation key.

Viewing Erasure Configuration

The erasure configuration may be viewed in the Maintenance Object Configuration portal.

Archiving the Object Erasure Schedule

The system provides the ability to archive older records in the object erasure schedule using Information Lifecycle Management. The object erasure schedule maintenance object is configured with a base ILM eligibility algorithm and an ILM crawler batch control.

The base object erasure schedule BO is configured to set the ILM archive switch when an erasure schedule record enters a final state and to reset it if the record becomes active again.

Refer to The Approach to Implementing Information Lifecycle Management for more information on archiving functionality.

Copyright © 2026, Oracle and/or its affiliates.

Data Redaction for Export and Logging

Personal identifiable information (PII) requires special handling when exported out of the application and written to files, including log files. Information that is encrypted at the database level remain encrypted and therefore protected when exported. With respect to non-encrypted information, the need to obfuscate it for added protection depends on how the exported data is intended to be used. If the exported information is part of an integration flow with an external system then the data should remain unchanged. However, when exporting data from a production environment for testing purposes sensitive information should be anonymized before they can be imported to a lower environment or written to logs for troubleshooting purposes.

The following are key aspects of this functionality:

- Redaction rules define how specific fields should be redacted when needed.
- When redaction rules exist, data redaction is performed by default by specific tools for export and logging. In absence of redaction rules, data is exported in its original form.
- Redaction rules are customer specific and may only be relevant to environments that contain sensitive data. As such, they are not released with the base product. Review your data redaction requirements and configure appropriate rules in relevant environments.
- Certain users may be allowed to export data in a non-redacted form based on application security.

The following sections provide additional information about this topic.

Redaction Rules Are Customer Specific

Redaction rules identify which fields should be redacted and how they should be anonymized. The list of fields to redact and the function to use to manipulate their values are customer specific and therefore are not released with the base product. A sample set of rules are provided as accelerator data for reference only.

It is important that you review your data redaction requirements and configure rules that reflect your business needs.

Redaction By Default

Processes that support data redaction on export and when logging do so by default when redaction rules exists. When rules are not defined, data is exported as is. Note that for this reason it is unnecessary to define such rules in lower environments that do not contain sensitive information.

Specific Export Tools Only

Data redaction is supported by the following tools only:

- **Content Migration Assistant (CMA).** Depending on application security, a user may indicate on a specific data set export request that data should be exported in its original form.
- **General Data Export (GDE) batch processes.** A batch parameter may be used to explicitly request that data is exported in its original form. This option is also protected by application security and enforced at batch submission time.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This type of process establishes a single export stream that may be sent to multiple targets. Review these integrations to ensure they do not have conflicting data redaction requirements and set the batch parameter accordingly.

- **Debug logging**. By default, data is redacted base on redaction rules when writing details to the various logs.
  - In an environment that is not a live production environment, a user with appropriate security may use a feature configuration to turn this off. Find the Feature Configuration record for the **General System Configuration** feature type. (It may need to be defined if it does not exist). Choose the option type **Turn Off Log Redaction** and define a value of **Y**.
  - The system uses a property (isLiveProduction) to check if an environment is a live production environment. In a cloud implementation, the development operations team sets the property when your implementation has indicated that you are live. In an on premises implementation, it is recommended to set this property when you are live. Refer to the system administration guide for more information.

Different Types of Redaction Rules

Sensitive information may reside in various types of storage fields which impact the way they are identified for redaction. You may define rules that target fields on specific tables or provide a more global rules that applies to all references of a specific field.

The application supports the following types of redaction rules:

- **Physical Field.** This type of rule applies to a value residing in a field on a specific table. An optional filter expression may be provided when only records of a certain type needs to be redacted.
- **XML Storage.** This type of rule applies a value associated with the specified field name that resides within an XML storage field on a specific table. This option also supports a filter expression.
- **Reference Field.** This type of rule applies to a value residing anywhere and associated with the specified field name. It does not include fields referenced in XML storage fields.

Redaction Function

A redaction function defines the format by which a value is anonymized. The type of function used depends on the field's data type. For example, to anonymize a numeric field the redaction function should only use numeric values etc.

Refer to the **Redaction Function** extendable lookup for a list of available functions. The base product provides a sample of common functions but the list may be extended by your organization as needed.

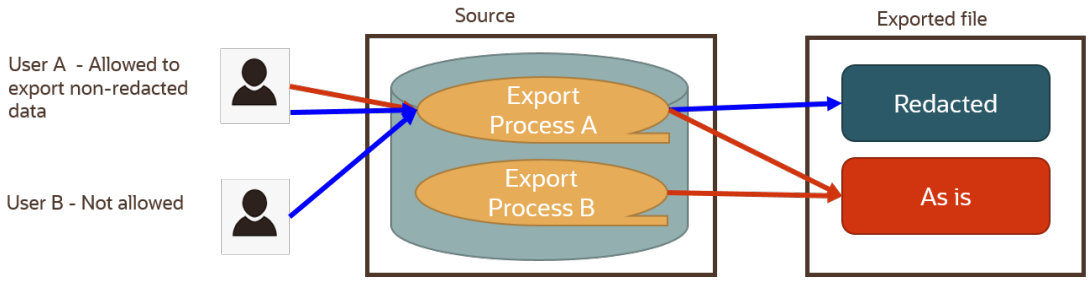
Fields May Contain Sensitive Information

The field definition record includes an indication as to whether the field may contained sensitive information or not. Note that the indication does not imply that the field is actually used to capture such data but rather that it can potentially be used this way. Use this indication to review fields that may need to be associated with a redaction rule.

Copyright © 2026, Oracle and/or its affiliates.

Exporting Non-Redacted Data Requires Security Rights

By default, the export tools redact data based on existing redaction rules. There are situations where certain users may be allowed to export data in its original form, that is, to request to not apply the redaction rules.



The following points describe the processes for exporting data:

- Some processes always export data as is, in case the target consumer needs the raw data. This is irrespective of the user requesting the export.
- Some processes perform data redaction by default and may offer an option for exporting data as is. The option to export data in its original form is secured by application security.
- In this example, process A supports data redaction by default with an option to export non-redacted data. User A has the right to export the data redacted (default) or explicitly request to export it in its original form. User B does not have the right to do so hence their export content is redacted by default.

The option to export data in its original form is associated with the **Do Not Apply Redaction Rules** access mode associated with the following relevant application services:

- For CMA, the additional access mode is associated with the migration data set export's application service. When granted, the user may select the special non-redacted export option when they create the export data set request.
- For GDE batch processes, the access mode is associated with their application service and is tied to the **Do Not Apply Redaction Rules** batch parameter they support. Only users with the proper application security can submit these batches using the special batch parameter option

Defining Redaction Rules

This portal is used to view and maintain rules that define how specific fields that containing personal information should be redacted when exported out of the application or written to a debug log. For example, an e-mail address may be redacted by replacing some or all characters with '*'.

Refer to Data Redaction for Export and Logging for more information.

You can access the portal from the**Admin Menu**, and then **Security**, and then **Redaction Rule**. You are brought to a query portal with options for searching for a specific redaction rule. Once a redaction rule has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Redaction Rule.** This zone provides information about the selected rule.

Copyright © 2026, Oracle and/or its affiliates.

Obfuscation Configuration

This portal is used to view and maintain all configuration options related to personal information obfuscation. It includes all masking, encryption and data redaction rules.

You can access the portal from the**Admin Menu**, and then **Security**, and then **Obfuscation Configuration**.

Refer to Encryption and Masking for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Redaction Rules.** This zone lists all redaction rules. You may add a new rule from this zone.
- **Masking.** This zone lists masking options. You may navigate to the masking related feature configuration to review and edit these options.
- **Encryption.** This zone lists encryption options. You may navigate to the encryption related feature configuration to review and edit these options.
- **Review Fields for Redaction Rules.** This zone lists fields that may contain personal identifiable information and are not associated with a redaction rule. Review these fields and add redaction rules when applicable.

Cryptography Keys

The system provides support for defining various types of cryptography keys to handle different use cases in the system related to encrypting / decrypting data, providing credentials and providing and reading signatures when exchanging data between two parties. This may be used when communicating information with an external system or for internal processing.

The following sections include information about the functionality provided to support this type of security.

- Understanding Key Rings
- Defining Key Rings

Understanding Key Rings

The Key Ring object is provided to define the specific use case and to reference the keys that are used over time for a given business use case. Each key ring references a specific business object, which defines all the rules related to managing the keys. For example:

- If the key ring is used to define a pair of private and public keys, then the business object includes actions for generating a key pair and for viewing the public key.
- If the key ring is used to capture the public key of a third party, then the business object includes actions for capturing the external public key.
- Each key ring defines the rules related to the possible lifecycle of the keys.

The following sections include information about the functionality provided to support different key ring classes for particular use cases.

- Field Encryption
- OAuth Keys
- RSA Signature Keys

Copyright © 2026, Oracle and/or its affiliates.

- File Encryption and Signing Keys

Field Encryption

The system supports encryption of certain sensitive fields in the database. This support includes the ability to capture hash values of encrypted data for searching purposes.

Refer to Application Encryption to understand how to configure which fields you want to encrypt.

The sections in this topic provide more information about the defining the encryption and hash key values and the key rotation support.

Symmetric Key

The product provides an AES Symmetric Key business object. It generates an encryption key using an AES 256 algorithm.

When defining a key ring for this business object, you provide a 3-character "wrap" prefix that is unique across key rings for the 'symmetric key' class. This aids in key rotation, described below.

The business object supports key rotation with respect to the keys. See the section below for more information about key rotation.

Once your symmetric key ring is defined and you have at least one active key, you can configure the fields that you want to encrypt and reference this key ring in the configuration. Refer to Application Encryption for more information.

Hash Key

The product provides an HMAC Hash Key business object. It generates an HMAC key and stores it in PEM format.

When defining a key ring for this business object, you provide a 3-character "wrap" prefix that is unique across key rings for the 'hash key' class. This aids in key rotation, described below.

Once your hash key ring is defined and you have at least one active key, for encrypted fields that support a hash value, you can reference this key ring in the configuration. Refer to Application Encryption for more information.

Key Rotation

Defining your encryption and hash keys using key rings allows for key rotation.

- When the system encrypts the data or creates the hash field, it uses a marker so that the system knows what key ring and key were used to encrypt / hash the data. Data is encrypted / hashed with the current Active key.
- If your organization wants to introduce a new key for either your symmetric key ring or your hash key ring, use the Generate Key button to create a new key. Click the Activate button adjacent to that Key to make it the key to use for all data that gets added or updated going forward. The previous key is marked Inactive.
- All records added or updated going forward use the new key.
- To rotate the keys across all the existing data encrypted / hashed with the previous key, you need to submit a batch. Refer to Mass Key Rotation for more information.

Copyright © 2026, Oracle and/or its affiliates.

Key Ring Rotation

The respective business objects for the symmetric key ring and the hash key ring define the algorithms to use for generating the key. If there comes a point where you want to use a different algorithm for generating the symmetric encryption key or the hash key, a new business object is needed so that the new desired algorithm can be defined. The following points highlight the steps needed to change the key generation algorithm.

- Define the new business object for the use case (symmetric encryption key or hash key). Ensure that the algorithm related to the Generate Key state has the desired key generation logic. This business object should include a field for the 'wrap prefix', similar to the delivered product business objects for the symmetric key ring or hash key ring.
- Define a new key ring for the new business object and be sure to choose a different wrap prefix value. This is needed for key rotation.
- Generate a key for the key ring and activate it.
- Go to the Feature Option Configuration and update the keyRing or hashKeyRing references for all the entries to point to the new key ring.
- All records added or updated going forward use the new key ring and its active key.
- To rotate the keys across all the existing data encrypted / hashed with the previous key ring and its key, you need to submit a batch. Refer to Mass Key Rotation for more information.

OAuth Keys

The system provides support for sending messages secured by OAuth authentication. To deliver the message, the application is required to provide a client ID and secret. The OAuth secret is sensitive data and needs to be stored securely in the application. The Message Sender configuration provides the ability to store the client ID and an encrypted secret directly in the sender context. Alternatively, the sender context can refer to a key ring that stores those details. Using a key ring allows users to apply new secrets over time without having to update the affected message sender configurations.

Refer to Message Sender - Context for more information about configuring a message sender for OAuth security.

The product provides an OAuth Keys business object that supports the following functions:

- Defining the external system related to the key ring
- Adding a client ID and secret key. The secret key is stored in an appropriate "secret store".
- Key rotation. For increased security, a new secret key may be entered periodically. The new key automatically becomes the active key.

RSA Signature Keys

RSA signature key pairs are used in the following cases:

- Certain processes in the system may need to write files to or read files from Oracle Cloud Object Storage. Cloud object storage has its own method for encrypting files, and products using the framework application do not need to define additional configuration for that. However, when communicating with cloud object storage, the application is required to provide a signature key.

  Refer to External File Storage for more information about configuring the system to connect to cloud object storage.

Copyright © 2026, Oracle and/or its affiliates.

- Customers may need the ability to authorize a customer service representative (CSR) to view the customer's account information in a customer facing portal. A signature key is required to support secure communication with the endpoint.

  Refer to User Redirection for more information about configuring the system to enable this type of redirection.

- Signature keys are used for outbound messages when using the OAuth JWT assertion grant type. The key is used to sign the assertion.

The product provides an RSA Signature Key Pair business object that supports the following functions:

- Generating a key pair, public and private. The private key is stored in an appropriate "secret store" and the public key is available to copy and share with a third party.
- Viewing the public key, allowing a user to copy it to register it with cloud object storage.
- Activating the new key pair (to be done after registering the new public key).
- Inactivating an expired key.
- Key rotation. For increased security, a new key pair should be generated periodically.
- Deactivating all keys for the key ring.
- Requesting a Certificate Signing Request
- Certificate Signing Request

Certificate Signing Request

If a certificate signing request is required for any use case, the following steps should be followed.

- Navigate to the signature key ring that requires a certificate.
- In the Keys zone, identify the pending or active key that requires a certificate. Click **Generate CSR**.
- You are prompted to provide appropriate distinguished name parameter values and set the duration.

> **Note**
>
> The distinguished name parameters are defined in the `F1-DistinguishedNameParameter` extendable lookup. The visible parameters are those that marked as 'default in grid'. Additional parameter names are available, if needed.

- Optionally check the Self Signed Certificate switch, for example in a development or test environment.
- Click Save

If you didn't check the self signed switch, copy the returned Certificate Signing Request text to a .pem file and use a certificate authority to provide the certificate. Otherwise, use the returned Self Signed Certificate.

Once you have the certificate, provide it to the relevant external party, according to the use case.

Copyright © 2026, Oracle and/or its affiliates.

File Encryption and Signing Keys

For security, files exchanged with third parties may be encrypted, digitally signed or both. This requires the sender and recipient to maintain public/private key pairs and share public keys. The product supports the maintenance and generation of public and private keys in a format suitable for encryption using PGP standards.

The following sections provide additional information about encryption keys.

Encryption Key Pairs

Internal encryption key pairs are used in the following cases:

- Creating a digital signature for an extract file. The signature is encrypted with the private key of the key pair and decrypted by the recipient using the shared public key .
- Decrypting an import file. The expectation is that the sender has encrypted the file data using a temporary key. The temporary key is encrypted using the shared public key and decrypted using the private key.

The product provides an Encryption Key Pair business object that supports the following functions:

- Generating a key pair, public and private. The private key is stored in an appropriate "secret store" and the public key is available to copy and share with a third party. Both keys are in formats suitable for PGP encryption.
- Viewing the public key, allowing a user to copy it to register it with third parties.
- Activating the new key pair (to be done after sharing the new public key).
- Inactivating an expired key.
- Key rotation. For increased security, a new key pair should be generated periodically.
- Deactivating all keys for the key ring.

Public Encryption Keys

External public encryption keys are used in the following cases:

- Verifying the signature of an imported file. The signature is decrypted using the public key provided by the sender.
- Encrypting an extract file. The batch file adapters create a temporary key which is used to encrypt the file data. The temporary key is encrypted using the public key provided by the recipient

The product provides a Public Encryption Key business object that supports the following functions:

- Defining the external system and reference ID related to the key ring.
- Adding a public key to the key ring. The product expects the public key to be in a format suitable for PGP encryption.
- Viewing the public key.
- Inactivating an expired key.
- Key rotation. When new public keys are provided by the external system, the previously active key is set to expiring status. It may be manually inactivated after all messages using that key have been processed.

Copyright © 2026, Oracle and/or its affiliates.

- Deactivating all keys for the key ring.

Defining Key Rings

Refer to Understanding Key Rings for an overview of key ring functionality.

To maintain the key rings applicable to your product or implementation, open **Admin**, and then **Security**, and then **Key Ring.**

This is an All-in-One portal and includes the standard List and display zones for a key ring.

The information captured on the key ring depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

Key Zones

The key zones provide information about the keys that are associated with the key ring. The zones displayed and the available actions depend on the type of key. Refer to embedded help for more information.

Key Rotation

For internal keys, it is recommended to practice key rotation and generate new keys periodically. New key pairs are generated in the **Pending** state. The user should register the public key of the new key pair with the associated third party. Once that is done, the new key should be activated.

For external keys, new public keys are added in the **Active** state. The previously active key is set to **Expiring** and may be inactivated when no further processing needs to refer to it.

User Interface Tools

This section describes tools that impact many aspects of the user interface.

- Understanding Portals and Zones
- Setting Up Portals and Zones
- Defining Menu Options
- Defining Navigation Keys
- Defining Navigation Options
- Navigation to External Pages
- Understanding System Messages
- Understanding Unified Search
- Trees
- Contextual Insights
- Knowledge Articles
- Color Configuration
- Defining Display Icons
- Custom Look and Feel Options

Copyright © 2026, Oracle and/or its affiliates.

Understanding Portals and Zones

A portal is a page that is comprised of one or more information zones. The base product pages are built using either a fixed page metaphor or using portals and zones.

There are three broad classes of portals:

- **Standalone.** Standalone portals are separate pages where the main tab of the page is built using a portal. These pages are opened using any of the standard methods (e.g., by selecting a menu item, by selecting a favorite link, etc.). Additional tabs for a stand-alone portal may be included using tab page portals.
- **Tab Page.** These types of portals cannot be attached to a menu. They simply define the zones for a tab on a standalone portal.
- **Sidebar.** A sidebar portal is a special portal that dictates what may appear in the Sidebar Area. Its zones contain tools and information that exist on the user's desktop regardless of the transaction. Several sidebar portals and zones are delivered as part of the base-package. Your implementation can add additional zones to these portal.
- **Assistant.** This is a special portal that defines what may appear in the Assistant Area if configured by your specific product.

The contents of this section describe general information about portals and zones.

Portal Page Structure

A portal page has a main tab and may have additional tabs. The page is constructed by first creating a **Standalone** portal, which is also the main portal, followed by creating a **Tab Page** portal for each additional tab and linking it to the main portal. Content is organized on each tab using independent zones.

The entire page is secured by the single application service associated with the main portal.

Portal Roles

An optional attribute called a portal role may be assigned to a portal. The role is typically for information purposes, but in some cases, a role may support additional configuration.

Certain roles may support predefined **UI Template** layouts to choose from.

Portals Are Made Up of Zones

A portal is a page that contains one or more zones, and each zone contains data of some sort. All zones reference a **Zone Type**. The zone type controls the behavior of the zone and the parameters available to configure the zone.

A portal may also use zones to present additional information in a drawer. These zones are not part of the page layout and are only presented upon a user request.

Not All Zones Make Up Portals

Zones may or may not be associated with a portal. The following describes additional uses of zones:

- A data explorer type of zone may implement a search for an entity on an input user interface map. These zones may be referenced on a FK reference record or directly referenced on a UI map.

Copyright © 2026, Oracle and/or its affiliates.

- A data explorer type of zone may implement a Business Service. The latter may be used for an internal process or used to build a user interface dropdown.
- A zone may be defined as a **Context Sensitive Zone** that appears on the Sidebar when a specific page or portal is in context. This type of zone is not linked to the Sidebar portal but associated with the context page or portal's navigation key.
- A 'view detail' zone may be launched into a drawer without being configured as part of the portal it is launched from.

Configuring Zones for a Portal

The portal includes configuration of how the zones should appear on the portal by default. This includes the following options, all of which may be overridden by an implementation.

- The order in which the zone should appear. An implementation may configure an override sequence to change the order zones on a base delivered portal.
- Whether the zone is visible on the portal. Zones delivered in the base product should be configured to be visible. But an implementation may override this if desired.
- Whether the zone should display initially collapsed or not. A zone's data is only retrieved when it is expanded. As such, a zone may be configured to be initially collapsed when the data is not needed very often. A user can expand the zone when the information is required. Implementations may change the collapsed setting of a base product portal / zone. Refer to Zones May Appear Initially Collapsed When a Page Opens for more information.

Flexible Layout Control

By default a zone spans the full width of the portal's display area but can be configured in a number of ways using the layout control features on the portal.

These features allow portal designers to have finer-grained control over how zones are laid out, and enables them to achieve a better user experience and a more optimal layout that reduces unused space.

The following highlight key layout control options:

- In addition to the standard half and full width options, 1/4, 3/4, 1/3 and 2/3 width options are also available.
- Better handling of zones that are dynamically hidden by a visibility script. When a zone becomes invisible its space may be reclaimed by a next zone as long as it is of same width or smaller.
- A zone may define the beginning of a new row. This indication allows for a more fixed layout that is not impacted by hidden zones. The new row indication may establish a layout of one, two or three zones displayed vertically.
- Support the following zone height options:
  - **Content (Unlimited Height)** lets the height be determined by content.
  - **Content (Limited Height)** lets the height be determined by content up to a specified maximum limit. This option may be used to achieve a more consistent portal layout, especially when a zone is rendered on long or small devices.
  - **Fixed Height** sets the zone at a specific height regardless of content.

    This option too may be used to achieve a more consistent portal layout.
  - When a height option is not explicitly selected, zone height is unlimited and determined by content. However, for a data explorer zone, the results grid area itself (not the entire

Copyright © 2026, Oracle and/or its affiliates.

zone) is limited by the zone parameter "height of report" (which has a default value of 50%).

You may also control the visibility of the following page components:

- **Page Title.** Use the **Page Title Visibility** portal option on a standalone portal to indicate that the page title should not be visible for this portal and its related tab portals if any.

> **Note**
>
> The page actions displayed in the same row as the page title are also suppressed. As such, you should only use this option if the actions in that area are not needed, including Bookmark, Clear and Refresh.

- **Tab Menu.** Use the **Tab Menu/Tab Visibility** portal option to suppress one or more tabs on the page based on conditions. It also may be configured to suppress the tab area altogether.
- **Zone Header.** You may use the **Zone Header Visibility** general zone parameter to suppress an explicit map zone's header area to allow for more vertical space when the page only has one zone and the standard actions on the zone header are not relevant.

Portal Specific Page Actions

The system supports defining actions that appear in the Page Actions Toolbar that are specific to a portal. Portal specific actions are first defined using a menu with the menu type of **Page Action Menu**. Each menu line in a page action menu should define one item. The menu line's item may reference a navigation option or a sub-menu (which in turn defines one or more lines with menu items that define navigation options). Use the **Page Action Menu** portal option to associate the menu with the portal. More than one page action menu option may be defined. If the lines on the referenced menus define navigation options, they are shown as distinct buttons. If any line references a sub-menu, then an action menu is shown, using the description on the menu line.

> **Note**
>
> Standard menu security is checked. For any menu item in any of the page action menus defines an application service / access mode, you only see the action if you have security.

Advanced Configuration for Portal Title / Tabs

You may configure your portal to provide additional information to be displayed either in the portal's Page Title Area or in the tab area.

Information String. If the portal is configured to display a single record, you can configure the **Page Info FK Reference** portal option to include the FK reference for the related maintenance object. The system will automatically display the information string for the FK reference when viewing a record on this type of portal.

Advanced Title Configuration. You may provide a service script in the **Enhanced Page Title** portal option to override the page title or to add additional information to be displayed. The service script should include the data area F1PageTitleDetails. Refer to the detailed description of the data area for more information on what the service script can return. This option may only be configured on the standalone ("main") portal, but the information is visible on any of the portal tabs.

Copyright © 2026, Oracle and/or its affiliates.

Scorecard Configuration. For portals that are configured with the role of Dashboard, you may configure a scorecard for each tab in the portal. To do this, you provide a service script in the **Scorecard** portal option. The service script should include the data area F1ScorecardDetails. Refer to the detailed description of the data area for more information on what the service script can return. If the dashboard has multiple tabs, if any of the tabs configure a scorecard service script, all tabs must configure one.

Portal Banners

Important messages related to the context of a page may be displayed in the form of banners presented in the title area. For example, a customer dashboard may show a banner message alerting the agent of an existing outage along with the ability to view outage details in a drawer.

These messages are composed by a script referenced on a **Portal Banners** portal option. Banner messages may be associated with various severity levels and optionally support related actions. The option is defined for the main tab of the portal and applies to the entire page.

The product provides a generic 'Portal Banners by Insight Class' (F1BannesInst) script that composes messages based on insights for an insight class and a given context. Using this script allows banners for a portal to be configurable by the customer.

Portal Templates

When defining a portal with the portal role of **Dashboard**, you may optionally choose a **UI Template**. The UI templates define a predefined layout. You simply need to choose the zones that you wish to include and the template will render the zones based on the template's configuration.

User Preferences

A **Standalone** portal includes configuration to indicate whether or not the portal should appear on a user’s portal preferences. The portal will not appear on the user's portal preferences if the **Show on Portal Preferences** setting is set to No. Note that an implementation may change this value for a product delivered portal. This setting is typically enabled for a portal that provides disparate information where not all zones are applicable to all users or where users may wish to adjust the order of the zones. An example of a portal enabled for portal preferences is the Sidebar portal.

The user can override zone oriented configuration for the portal:

- Which zones appear on that portal
- The order in which the zones appear
- Whether the zones should be initially collapsed when the portal opens.
- The refresh seconds. This is applicable to zones displaying data that changes often.

An implementation can optionally configure the system to define portal preferences on one or more "template" users. When a template user is linked to a "real" user, the real user's preferences are inherited from the "template" user and the "real" user cannot change their preferences. Some implementations opt to work this way to enforce a standard look and feel for users in the same business area.

Refer to User — Portal Preferences for more information about how users configure their zones.

Copyright © 2026, Oracle and/or its affiliates.

View Detail Drawers

A zone may be designed to display related extended information in a drawer. This type of zone is designed and implemented like any other zone. It is recommended to associate it with the 'View Details Drawer' role for reporting purposes.

View detail zones can be configured as follows:

- **Portal panels.** An optional list of 'view detail' zones can be defined as part of a portal's zone configuration. These zones are not rendered as part of the portal's layout. Instead, they represent groups of zones to be presented in a drawer upon user request. A 'view detail' zone is associated with a **Panel Name** which represents a uniquely named drawer content. When the panel name is launched all zones that belong to that panel are presented in the drawer area in sequential order.
- **Not related to a portal.** A zone can be launched directly into a 'view detail' drawer, without being listed on any portal. This technique is used when the zone displays information that is not specific to the current portal. For example, the Knowledge Article drawer may be launched from any portal that supports a default knowledge article.

The product supports small, medium and large drawer sizes. When launched as a panel, the largest drawer size setting of all panel zones is used. When a zone is directly launched, not via an association to a panel, the width of the zone sets the drawer width.

A drawer may be launched from another drawer, forming a nested display of panels.

Support for launching a drawer is available from a display UI map, a data explorer column, and a chart. Refer to Launch a View Details Drawer for more information.

> **Note**
>
> Drawers may not be launched from the Sidebar and Assistant areas.

View Entity Drawer

In some pages, the product supports the ability to review an entity's information in a drawer without navigating to the entity's page.

The default drawer for viewing an entity's information may be dynamically determined using the **View Entity Drawer** maintenance object or business object option. If a business object option exists it overrides the maintenance object option if any. When applicable, you may use these options to override the 'view details' zone used by the product an entity. To do so, implement your custom 'view detail' zone and reference it as an option with a higher sequence number.

This feature is only available in select pages. By default, a link to an entity navigates to its portal, and does not launch the entity's view details drawer. To implement such functionality in a custom user interface, refer to product examples that use the 'Get Drawer Zone for an Entity' (F1GetDrwEnt) script.

Removing Zones From Portal

You cannot remove a base product zone from a base product portal. An implementation may override the **Display** setting to prevent a zone from displaying on the portal. In addition, you cannot remove a zone if a user has enabled it on their portal preferences. To remove a zone from the portal list, first make sure that no user has it enabled in their portal preferences.

Copyright © 2026, Oracle and/or its affiliates.

Granting Access to A Portal

An application service is associated with each portal. A user must be granted access rights to the respective application service in order to see a portal.

When you add a new **Standalone** portal, the system automatically creates its application service behind the scenes and displays it as part of the portal definition. You'll need to use this application service to grant access to the portal. **Tab Page** portals do not have separate security access. If a user has access to the main stand-alone portal, then the user will have security access to all its tabs.

Please note the following in respect of how application security impacts a user's portals:

- A user's Portal Preferences page only shows the portals configured to show on user preferences and where they have security access.
- The system's menus only show portals to which a user has security access.
- Users can set up favorite links to all portals, but they must have security rights to the portal's application service in order to invoke the favorite link.

Granting Access to Zones

An application service is associated with each zone. A user must be granted access rights to the respective application service in order to see a zone on a portal. Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

Please note the following with respect to zone application security:

- For most base product portals, all the zones for all the portals reference the same application service that is used to grant access to the portal. In other words, if the user has access to the page, then the user has access to all the zones on all portals for the page. There may be exceptions to this rule for certain portals.
- For a base product multi-query zones, typically the individual query zones and the multi-query zone reference the same application service that is used to grant access to the main (standalone) portal for the page. However, there may be individual query zones provided with a unique application service. This may occur when the query option is unusual and not applicable to all users or even to all implementations. If a user does not have security access to an individual query zone, that option will not be available in the dropdown.
- For base product portals that are configured to show on portal preferences, it is common that the portal contains different types of zones that may be applicable to different types of users. Typically these types of portals will deliver a unique application service for each zone so that an implementation may configure which user groups are allowed to view each zone. For these types of portals, please note the following:
  - A user's Portal Preferences page contains a row for a zone regardless of whether the user has access rights to the zone. Because of this, the system displays an indication of the user's access rights to each zone.
  - If a user's access rights to a zone are revoked, the zone will be suppressed when the user navigates to the respective portal.
  - Revoking a user's access rights does not change the user's portal preferences (i.e., a user can indicate they want to see a zone even if they don't have access to the zone - such a zone just won't appear when the respective portal appears).

Copyright © 2026, Oracle and/or its affiliates.

Putting Portals on Menus

When you add a new **Standalone** portal, the system automatically creates its navigation option behind the scenes and displays it as part of the portal definition. You'll need to use this navigation option to add the portal to a menu or to allow any other type of navigation method to the portal.

To add a portal to a menu, you must add a menu item to the desired menu and reference the portal's navigation option on the new menu item. A portal's navigation option can appear on any number of menu items (i.e., you can create several menu items that reference the same portal).

Your users can set up their preferences to include the portal's navigation option on their Favorite Links. This way, they can easily navigate to the portal without going through menus.

Portal Options

Portal options are used to enable various features available for each portal. Several are mentioned in previous sections in this topic. Review the detailed descriptions of each portal option to understand what other features are available.

> **Note**
>
> **You can add new option types.** Your implementation may want to add additional portal option types. To do that, add your new values to the customizable lookup field `PORTAL_OPT_FLG`.

Setting Up Portals and Zones

The topics in this section describe how to set up portals and zones. Please refer to The Big Picture of Portals and Zones for background information.

- Defining Zone Types
- Defining Zones
- Defining Context-Sensitive Zones
- Defining Portals

Defining Zone Types

A Zone Types represents a particular type of zone with a specific behavior. For example, a data explorer zone type is used to select data using a specific SQL statement and display the data based on parameter configuration. The zone type defines the Java Class that controls the behavior of the zone and defines the parameters that the Java Class supports. The base product supports many zone types used to build the portal / zone user interface. Implementations may introduce their own zone types.

> **Note**
>
> It is not very common for an implementation to introduce their own zone types.

Select **Admin**, and then **System**, and then **Zone Type** to maintain zone types.

Copyright © 2026, Oracle and/or its affiliates.

Description of Page

Specify an easily recognizable **Zone Type** code and **Description**. Use the **Detailed Description** to describe in detail what the zone type does.

> ⚠ **Caution**
>
> When adding new zone types, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Owner indicates if this zone type is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a zone type. This information is display-only.

Java Class Name is the Java class responsible for building the zone using the parameters defined below.

Two types of parameters are specified when defining a zone type:

- Parameter values that have a **Usage** of **Zone** are defined on the zones and control the functionality of each zone governed by the zone type. A **Usage** value of **Zone - Override Allowed** indicates that an implementation may override the parameter value for a base zone.
- Parameter values that have a **Usage** of **Zone Type** are defined directly on the zone type and control how the zone type operates (e.g., the name of the XSL template, the name of the application service). A **Usage** value of **Zone Type - Override Allowed** indicates that an implementation may override the parameter value for a base zone type.

The following points describe the fields that are defined for each parameter:

- **Sequence** defines the relative position of the parameter.
- **Parameter Name** is the name of the parameter.
- **Description** is a short description that allows you to easily identify the purpose of the parameter.
- **Comments** contain information that you must know about the parameter or its implementation. For parameters with a usage of **Zone** or **Zone - Override Allowed**, this information is visible to the user when viewing or defining this parameter for a zone of this type.
- **Usage** indicates whether the parameter value is defined in a **Zone** of this type or in the **Zone Type**. **Zone - Override Allowed** and **Zone Type - Override Allowed** indicate that override values for the parameters defined in a base zone or base zone type can be entered.
- **Required** is checked to indicate that a zone must define a value for the parameter. It is not checked if a value for the parameter is optional. This field is protected if the **Usage** is **Zone Type** or **Zone Type - Override Allowed**.
- **Parameter Value** is used to define the value of zone type parameters. This field is protected if the **Usage** is **Zone** or **Zone - Override Allowed**.
- **Owner** indicates if this parameter is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a parameter. This information is display-only.

Copyright © 2026, Oracle and/or its affiliates.

Zone Type Parameter Comments

For the product owned zone type parameters, the parameter's detailed description provides the detail needed for properly configuring the parameter. For the Action parameters (IMPLEMENTOR_ACTION_n), the parameter description is abbreviated. Additional detail about configuring this parameter may be found in the Zone Action Parameter detailed information. The same details apply.

Defining Zones

This portal is used to view and maintain the configuration of a zone. Implementations may define custom zones or override the descriptions and some parameter values for base product zones.

You can access the portal from the**Admin Menu**, and then **System**, and then **Zone**. You are brought to a query portal with options for searching for a specific zone. Once a zone record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding Portals and Zones for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Zone.** This zone provides general information about the selected zone.
- **Query Options.** This zone maintains the list of query options for a Multi-Query zone. The zone is only visible for Multi-Query zones.
- **Filters.** This zone maintains various zone parameters that handle filtering options for a data explorer zone. The zone is only visible for data explorer zones.
- **General Parameters.** This zone maintains various general purpose zone parameters for a data explorer zone.
- **SQLs.** This zone maintains individual SQLs and their related parameters. The zone is only visible for data explorer zones.
- **SQL Configuration.** This zone maintains all the zone parameters related to a specific data explorer SQL. The zone is only visible when a specific SQL is broadcast from the **SQLs** zone.

The following zones may appear as part of the portal's **References** tab page

- **Zone References.** This zone list various configuration entities referencing the current zone.
- Overriding Parameter Values
- Overriding Zone Parameters
- Zone Parameter Details
- Data Explorer Graph Configuration
- Pagination Configuration

Copyright © 2026, Oracle and/or its affiliates.

- Auto-filtering Configuration
- Use Data Explorer for Derived Data
- Configuring Timeline Zones

Overriding Parameter Values

Implementation teams may override the value of selected types of base provided zone parameters. The override parameter value field for base owned zone parameters is only enabled when the related zone type parameter usage is **Zone - Override Allowed**.

For multi-query zones, you may use the override parameter value to add custom query options to a base product owned zone.

Overriding Zone Parameters

Implementations may override the value of a base owned zone parameter only if the zone type definition allows such override. The override parameter value is enabled for a specific base owned parameter when the related zone type parameter usage is set to **Zone - Override Allowed**.

You may use the override parameter capability to provide alternative or additional custom query options for a base owned multi-query zone.

Zone Parameter Details

For most zone parameters, the embedded help for the parameter provides the detailed information needed for configuring the parameter values. For some parameters with very detailed descriptions, the embedded help is abbreviated and more detail is provided here.

- Zone Visibility Service Script
- SQL Statement
- Column Parameters
- Zone Action
- User Filters
- Hidden Filters
- Multi-Select Action

Zone Visibility Service Script

All zones support a visibility script that is used to determine if the zone should be displayed to the user or not based on conditions. The script may receive input parameters and is expected to return a Boolean value indicating if the zone should be displayed or not. The embedded help for the **Zone Visibility Service Script** parameter provides details related to the syntax.

The following table highlights some service scripts provided by the product that may be used if applicable to your zone's requirements. This is not an exhaustive list of visibility scripts. There may be others that are specific to a given zone.

Script Code	Description	Comments
`F1-ShldShwZn`	Zone Visibility - Display Zone in Portal	This script simply returns a value of 'true' and is used when the zone should always appear.

Copyright © 2026, Oracle and/or its affiliates.

Script Code	Description	Comments
F1-CondShwZn	Zone Visibility - Display Zone in Portal Conditionally	This is used when the condition for showing the zone is based on the population of a context value. This is commonly used when one zone in the portal should only appear after a broadcast of a record from another zone in the portal. For an example of a zone the uses this visibility script, refer to F1-BSFTYPE .
F1-RwCtShwZn	Zone Visibility - Based on Row Count	This is used when the condition for showing the zone is based on the existence of one or more rows that can be determined using SQL. This script accepts a zone code, user filters 1 through 25 and hidden filters 1 through 10. The script returns an indication of 'true' if at least one row count is returned by the zone. To use this visibility script, a specific data explorer zone must be developed for the specific use case. For an example of a zone the uses this visibility script, refer to F1-MIGRREQEL .

SQL Statement

Data explorer zones are used to select data to display using one or more SQL statements. The SQL parameters are applicable to the following zone types

- Info Data Explorer - Single SQL (F1–DE-SINGLE). The parameter has the description **SQL Statement**.
- Info Data Explorer - Multiple SQLs (F1–DE). The parameters follow the description pattern of **SQL Statement x**.
- Query Data Explorer - Multiple SQLs (F1–DE-QUERY). The parameters follow the description pattern of **SQL Statement x**.

> **Note**
>
> If your implementation has been configured to restrict the functions that may be used when defining an SQL then an error is issued at runtime if there are functions found that are not in the allowlist. The allowlist may be viewed using the **SQL function allowlist** link in the SQLs zone on the zone maintenance page.

> **Note**
>
> Zone SQL should take care not to use the following reserved keywords for the table alias. (For example, the table alias in this SQL is 'XX': SELECT 'X' FROM W1_ACTIVITY_TYPE **XX**.)

Keyword
AT

Copyright © 2026, Oracle and/or its affiliates.

The following table provides a list of SQL substituted keywords that may be used in the SQL Statement parameters in explorer zones. At execution time, the system determines the database and substitutes the keyword with the database specific syntax:

Keyword	Description	Examples
`@toCharacter()`	Converts the input to Character data type.	select **@toCharacter**(batch_cd) as batchCode from ci_batch_ctrl
`@toDate()`	Converts the input to Date data type.	select **@toDate**(last_update_dttm) as lastUpdateDate from ci_batch_ctrl
`@toNumber()`	Converts the input to Number data type.	select **@toNumber**(next_batch_nbr) from ci_batch_ctrl

Copyright © 2026, Oracle and/or its affiliates.

Keyword	Description	Examples
`@currentDate`	Fetches the current date.	select batch_cd, **@currentDate** as today from ci_batch_ctrl

> **⚠ Caution**
>
> The Oracle functions **SYSDATE** and `CURRENT_DATE` should not be used because they do not properly cater for adjusting dates from the database time zone to the installation time zone, if needed.

Copyright © 2026, Oracle and/or its affiliates.

Keyword	Description	Examples
`@currentTimestamp`	Fetches the current date / time. **⚠ Caution** The Oracle functions **SYSTIMESTAMP** and `CURRENT_TIMESTAMP` should not be used because they do not properly cater for adjusting the date / time from the database time zone to the installation time zone, if	select batch_cd from ci_batch_ctrl where last_update_dttm > **@currentTimestamp**

Copyright © 2026, Oracle and/or its affiliates.

Keyword	Description	Examples
	⚠ need ed.	
`@concat`	Combines the result list of two or more columns.	select batch_cd **@concat** next_batch_nbr concatNbr from ci_batch_ctrl
@substr(string, start)	**String** is the input String that you are trying to get a substring of. **Start** is the position of the character for the output results.	select batch_cd batchCode from ci_batch_ctrl Result: TESTCD select **@substr**(batch_cd,3) batchCode from ci_batch_ctrl Result: STCD
@substr(string, start, end)	**String** is the input String that you are trying to get a substring of. **Start** is the position of the character for the output results. **End** is the number of characters required in the output from starting position.	Select batch_cd batchCode from ci_batch_ctrl Result: TESTCD select **@substr**(batch_cd,3,2) batchCode from ci_batch_ctrl Result: ST
`@trim`	Trims the white spaces of the output on both sides.	select **@trim**(batch_cd) as batchCode from ci_batch_ctrl
The following syntax is related to 'fuzzy' searching. It is only applicable if Oracle DB Text is enabled and a context text index has been created. Refer to Advanced Search Options for more information.		
@fuzzy(string, score, numresult, 'weight')	**String** is the input value for the search. **Score** is the degree of 'fuzziness'. Valid values are between 1 - 80. The higher the number the more precise the search. Default is 60. **Numresults** is the number of variations to consider for the string. Valid values are between 1 and 5000. Default is 100. Indicate `'weight'` to signal that the results are returned in order of weight. Leave this setting off to indicate that the results are returned in order of score.	Set score to 70, number results to 6, and specify weight. select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70, 6, 'weight')) > 0
`@fuzzy(string)`	This returns a string result from the fuzzy expansion operation where the default value of 60 is assumed for the **score** and the default value of 100 is assumed for the **numresult**.	To use default values: select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1))> 0

Copyright © 2026, Oracle and/or its affiliates.

Keyword	Description	Examples
@fuzzy(string, score)	This returns a string result from the fuzzy expansion operation with the **score** specified and the default value of 100 for the **numresult**.	Set score to 70. select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70)) > 0
@fuzzy(string, score, numresult)	This returns a string resulted from the fuzzy expansion operation with the similarity score and the **numresults** specified.	Set score to 70, number results to 6. select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70, 6)) > 0

Column Parameters

Data explorer zones are used to select data to display using one or more SQL statements. For each SQL statement, the zone may configure up to 20 Columns that contain the formatting definition for displaying the output data.

These parameters are applicable to the zone types

- Info Data Explorer - Single SQL (`F1-DE-SINGLE`). The parameters follow the description pattern of **Column x**.
- Info Data Explorer - Multiple SQLs (`F1-DE`). For this zone type, all SQLs are executed and the zone displays a union of all the results. The parameters follow the description pattern of **Column x for SQL y**. There are some mnemonics that do not make sense to differ within rows of the same column. For example, the column label. For these mnemonics, only the value in SQL 1 is considered for that column. The table below indicates which mnemonics follow this rule.
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`). For this zone type, only one SQL is executed. The **SQL y Condition** parameter may be used to control this. The system will execute the first SQL whose condition is satisfied (or with no condition populated). The parameters follow the description pattern of **Column x for SQL y**.

The following sections describe the various types of mnemonics.

Source Mnemonics

This table describe the mnemonics that control how the data in a column is derived.

Mnemonic	Description	Valid Values	Comments
`source=`	Defines how the column's value is derived.	`SQLCOL`	Indicates that the source of the column's value comes from a column in the SQL statement. This type of column must also reference the `sqlcol=` mnemonic.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		BO	Indicates that the source of the column's value comes from a business object. This type of column must also reference the `bo=`, `input=` and `output=` mnemonics to define how to interact with the business object.
		BS	Indicates that the source of the column's value comes from a business service. This type of column must also reference the `bs=`, `input=` and `output=` mnemonics to define how to interact with the business service.
		SS	Indicates that the source of the column's value comes from a service script. This type of column must also reference the `ss=`, `input=` and `output=` mnemonics to define how to interact with the service script.
		FORMULA	Indicates that the source of this column's value is calculated using a formula. This type of column must also reference the `formula=` mnemonic.
		SETFUNC	Indicates that the source of this column's value is calculated using a superset of values from the rows in the SQL statement. This type of column must also reference the `setfunc=` mnemonic.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		ICON	Indicates that the source of this column's value is a display icon reference (meaning that an icon will be displayed in the column). This type of column must also reference the icon= mnemonic to define the icon reference.

Note

When using this source mnemonic, the format

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			ngmnemonic type=isnotapplicable.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		FKREF	Indicates that the source of this column's value is an FK reference (meaning that the FK reference's context menu and information string will be displayed in the column and will be enabled for navigation). This type of column must also reference the `fkref=` and `input=` mnemonics to define how the FK reference is called.

> **Note**
>
> When using this source mnemonic, the

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments

formattingmnemonic type=isnotapplicable.

Note

Toshowaf

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			oreignkey'sinformationbutnotenablenavigation,referto Comm

Mnemonic	Description	Valid Values	Comments
			onUseCases for the recommended technique.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		SPECIFIED	Indicates that the source of this column's value is specified by concatenating literals and other column values. This type of column must also reference the spec= mnemonic. When constructing an HTML element for a web component or a function, the syntax may be slightly different for a data explorer column. Refer to UI Map Attributes and Functions for the appropriate syntax.
		MSG	Indicates that the source of this column is a message from the message table (along with any substitution variables). This type of column must also reference the msg= mnemonic.
sqlcol=	Defines the column in the SQL statement when source=SQLCOL.	COLUMN_NAME	Enter the name of a column that is retrieved in the SELECT statement. Note that if the select statement uses an alias for a column, then the alias should be referenced here.
		x	Where x is an integer value that references a column by its relative position in the SELECT statement. For example, sqlcol=3 would display the 3rd column in the SELECT statement).

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
bo=	Defines the business object to invoke when source=BO. This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the business object.	'Business Object Code'	
bs=	Defines the business service to invoke when source=BS. This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to/received from the business service.	'Business Service Code'	
ss=	Defines the service script to invoke when source=SS. This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the service script.	'Service Script Code'	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`icon=`	Defines the icon to use when `source=ICON`.	Cx	This means the Display Icon code is defined in an earlier column. For example, define `C1` if column 1 defines the icon code value.
		COLUMN_NAME	This means the Display Icon code was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'image'	This means the image file is defined directly. For example '/images/treeRootObject.gif'.
		'Display Icon Code'	This means the Display Icon code is defined directly. For example 'F1EDIT.
`fkref=`	Defines the FK reference used to retrieve the column's information when `source=FKREF`. This mnemonic must be used in conjunction with the `input=` mnemonic to define how information is sent to the FK reference to build the information.	Cx	This means FK reference code is defined in an earlier column. For example, define `C1` if column 1 defines the FK reference value.
		COLUMN_NAME	This means the FK reference was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'FK Reference Code'	This means the FK Reference is defined directly. For example 'F1-ROLE'.
`formula=`	Defines the formula to use when `source=FORMULA`. Examples: • `formula=C1*.90/C2`	The formula can contain numeric constants, operators and column references.	For column references, use the format Cx where x represents the column number.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	• `formula =(C1/ C2)*100`		Refer to Expression Parser for information about the functions supported.
`setfunc=`	Defines the function to apply to the rows of a given column when `source=SETFUNC`.	function(Cx)	Where Cx represents a column whose rows should have the function applied and the function is one of the following: • **MAX**. This derives the maximum value of all rows in the column. • **MIN**. This derives the minimum value of all rows in the column. • **TOT**. This derives the sum (total value) of all rows in the column. • **ACC**. This derives the cumulative total of all rows up to an including the current row.
`input=`	This is used to define one or more input fields and values passed to business objects, business services, service scripts, and FK references. The syntax is as follows: [ELEMENT_NAME=ELEMENT_REF ELEMENT_NAME=ELEMENT_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the	Cx	Where Cx represents the value of a previous column. If the value to pass is in the first column, reference C1.
		COLUMN_NAME	This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'literal value'	This means a literal value within the single quotes should be passed in.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	ELEMENT_NAME, which is the name of the element / field in the target. ELEMENT_REF is the value passed in. The next column indicates the possible values for ELEMENT_REF.	`userTimeZone`	This means the current user's time zone should be passed in. This is typically used with the business service `F1-ShiftDateTime` to convert data in the storage time zone to the user's time zone for display.
		`installationTimeZone`	This means the installation time zone should be passed in. This is typically used with the business service `F1-ShiftDateTime` to convert data in the storage time zone to the installation time zone for display.
		Examples: • `input=[USER_ID=C1]` • `input=[USER_ID=USER_ID]` • `input=[input/targetTimeZone=userTimeZone]`	
`output=`	This is used to define the name of the element retrieved from the business object, business service or service script used to populate this column.	elementName	Example: `output=personInfo`
`pagingkey=`	This mnemonic is only applicable when the **Enable Paging** parameter has been configured. It indicates that this column is one of the keys used by the SQL statement to orchestrate paging through	`Y`	

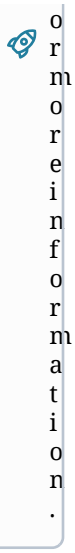
Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	results. This mnemonic can only be specified when the `source=SQLCOL`.	`N`	This is the default, meaning that you don't need to indicate `pagingkey=N` at all to indicate that the column is not one of the paging keys.

NOT_SUPPORTED

Refer to Pagination Configuration

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			

> **Note**
>
> If multiple columns are configured with the same source BO, BS or SS and the same input data, the system caches the output from the first call and reuses the results for subsequent columns.

Formatting Mnemonics

This table describe the mnemonics that control how a column is formatted.

> **Note**
>
> For the `F1-DE` zone type, zone displays the union of all the different SQLs.

Mnemonic	Description	Valid Values	Comments
`type=`	Defines how the column's value is formatted.	`STRING`	Columns of this type capture a string. This is the default value.
		`DATE`	Columns of this type capture a date and will be displayed using the user's display profile.
		`TIME`	Columns of this type capture a time (in database format) and will be displayed using the user's display profile.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		`DATE/TIME`	Columns of this type capture a date and time (in database format) and will be displayed using the user's display profile.
		`MONEY`	Columns of this type capture a monetary field. This type of column may also reference the `cur=` mnemonic. If the cur mnemonic is not specified, the currency code on the installation record is used.

on and Foreign Key columns. The source=source mnemonic may be use

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		`NUMBER` ⓘ dtoindicateacolumnshouldbederivedfromaniconreferenceoraf	Columns of this type capture a numeric field. This type of column may also reference the `dec=` mnemonic.

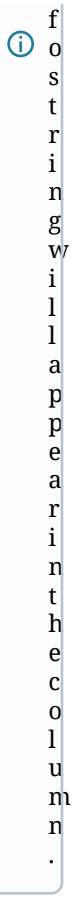
Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		oreignkey(FK)referenceIfyouuseeitheroftheseresources,thet	

Copyright © 2026, Oracle and/or its affiliates.

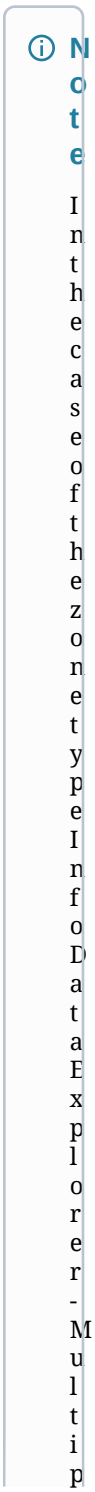
Mnemonic	Description	Valid Values	Comments
		ype=mnemonicisnotrelevantaseitheranicononracontextmenu/in	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			
label=	Defines the column's override label. The label appears in the column's heading and in the zone's drag and drop area. If this mnemonic is not defined, the system uses the column's default label. The source of a column's default label differs depending on the column's source. Note that some sources don't have a default	FIELD_NAME	Enter a valid field name whose label should be used for the column label. This should always be the option used if multiple languages are needed.

Copyright © 2026, Oracle and/or its affiliates.

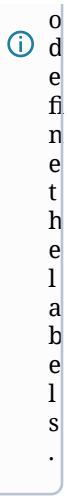
Mnemonic	Description	Valid Values	Comments
	value and omitting this mnemonic will result in a blank label.	'text'	Defines the text directly.



Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		leSQLs(F1_DE), thesystemwillusethecolumndefinitionsinSQL1t	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		 odefinethelabels.	
`cur=`	Defines the currency code applied when `type=MONEY` if the installation record's currency should not be used.	Cx	This means currency code value is defined in an earlier column. For example, define `C1` if column 1 defines the currency code.
		COLUMN_NAME	This means the currency code was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'Currency Code'	This means the currency code is defined directly. For example 'USD'.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`dec=`	Defines the number of decimal places when `type=NUMBER`. It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.	nR	Where n is the number of decimal places to show. Suffixing the number of decimal places with R means that the system should round up / down. Simply specifying n (without an R) means that decimal places should be truncated. For example, entering `dec=4` will display 4 decimal places and truncate the remainder.

Note

Formatting only. This mnemonic is only

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			used for formatting, it does not impact the precision used for subs

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			equentcalculations. For example, if a column retrieved from the

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			database contains 6 significant digits and dec=0, the column will be

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			show with decimal places (truncated), however any references

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			to the column in subsequent calculations will use 6 decimal places

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			. For example, if the column is referenced in a formula or set function

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			on, all 6 decimal places will be used.
`char=`	This mnemonic applies special character(s) to the column's value.	'x[]x'	Where x references the literal value to display and **[ ]** defines the relative position of the characters (before or after the value). You need only include the **[ ]** if you want to position characters in front of the value. For example, `char='%'` will place a percent sign after the value. If you want to position the word 'minutes' before a value, enter **char='minutes [ ]'**. If you want to output a value like **BUDGET $123.12 (YTD)**, enter **char='BUDGET [ ] (YTD)'**.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`suppress=`	This is used to indicate a column should not be displayed. A column would be suppressed if it's only defined for use by subsequent columns, for example, if there is a formula that derives a column using two other columns. In this scenario, the columns referenced in the formula can be suppressed.	`true`	

Note

In the case of the zone type Inf

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		`false` Data Explorer-MultipleSQLs(F1_DE), the system will apply the s	This is the default, meaning that you don't need to indicate `suppress=false` at all to indicate that the field should be shown.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		settings for the column definitions in SQL 1 to all subsequent SQLs.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`suppressSearch=`	This is used to indicate a column should not be displayed when the zone is invoked in search mode only.	`true`	

Note

In the case of the zone type InfoDataExplorer-Mu

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		`false` ltipleSQLs(F1-DE), thesystemwillapplythesettingsforthecol	This is the default, meaning that you don't need to indicate `suppressSearch=false` at all to indicate that the field should be shown.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		undefinitionsinSQL1toallsubsequentSQLs.	
`suppressExport=`	This is used to indicate a column should not be downloaded when exporting.	`true`	
		No	

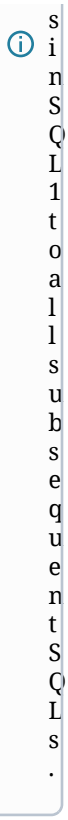
Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		`false` te In the case of the zone type InfoDataExplorer-MultipleSQLs (F	This is the default, meaning that you don't need to indicate `suppressExport=false` at all to indicate that the field should be included in a download.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		1 – DE), the system will apply the setting for the column definition	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			 sinSQL1toallsubsequentSQLs.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`width=`	This is used to override the width of a column (number of pixels). The default value is the maximum width of any cell in the column.	n	Where n is a number between 0 and 999.

Note

If there is no available break point in the data, the co

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			lumn will be longer than the specified number of pixels. The length

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			f t h e c o l u m n ' s l a b e l (w h i c h a p p e a r s i n t h e c o l u m n ' s h e a d i n g) m a y a l

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			some the width wider than specified. In the case of the zone type In

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			fo D at aE xp lo re r- M ul ti pl eS QL s(F1 _ DE), th es ys te m w il ls et th ew i

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments

dthbasedonthecolumndefinitionsinSQL1.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`color=`	This is used to override the column's text color.	A valid HTML "named" color	For example `color=yellow`.

Note

Refer to Color Contrast for information about the use of t

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			he HTML color 'red' and its impact on accessibility.
		A valid RGB color model combination	For example `color=#E0292F` or `color=#CCCCCC`. Note that the # is required.
`bgcolor=`	This is used to override the column's background color.	A valid HTML "named" color	Similar to the `color=` mnemonic.
		A valid RGB color model combination	Similar to the `color=` mnemonic.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`order=`	Defines the column's default sort order.	ASC	Indicates that the order is ascending. This is the default meaning that it is not necessary to indicate `order=ASC`.
		DESC	Indicates that the order is descending.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`rowHeader=`	Designates the column as a row header for accessibility purposes.	`true`	By default, the first data column of the data explorer results is identified as the row header, for accessibility tools. If the data in the first column does not uniquely identify the row, use this mnemonic to explicitly mark a different column or multiple columns as the row header.

Note

In the case of the zoneTypeInfoDataExplo

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			rer-MultipleSQLs(F1_DE),thesystemwilllookonlyatthecolumn

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			sinSQL1thathavethismnemonicdefinedandapplyittoallresultsf

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			orthosecolumns.

Click Mnemonics

This table describe the mnemonics that define whether a column value may be clicked and if so, what should happen.

Mnemonic	Description	Valid Values	Comments
`navopt=`	Defines the navigation option that references the target transaction or script when the user clicks a column. Note, this mnemonic should be used in conjunction with the `context=` mnemonic to define what information is sent to the navigation option's target transaction. This mnemonic is ignored if `source=FKREF` because the FK reference code defines the	Cx	This means navigation option code is defined in an earlier column. For example, define `C1` if column 1 defines the navigation option.
		COLUMN_NAME	This means the navigation option was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. Example: `navopt=MAIN_PORTAL`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	hyperlink's destination.	'Navigation Option Code'	This means the navigation option code is defined directly. For example `navopt='userMaint'`.
`context=`	This is used to define one or more context fields and values passed to the target navigation option to go along with the `navopt=` mnemonic. The syntax is as follows: [FIELD_NAME =FIELD_REF FIELD_NAME= FIELD_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME, which is the name of the context field in the navigation option. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF.	Cx	Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference `C1`.
		COLUMN_NAME	This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'literal value'	This means a literal value within the single quotes should be passed in.
`bpa=`	Indicates that a BPA script should be executed with the user clicks the column	Cx	Indicates that the BPA script is defined in a previous column.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	and indicates the BPA to execute. Note, this mnemonic should be used in conjunction with the `tempstorage=` mnemonic to define the temporary storage values that will be initiated when the script is executed. This mnemonic is ignored if `source=FKREF` because the FK reference code defines the hyperlink's destination.	COLUMN_NAME	This means the BPA script to execute was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'BPA Script Code'	This means that the BPA script to execute is defined directly.
`tempstorage=`	This is used to define how temporary storage variables are initiated when the `bpa=` mnemonic is used. The syntax is as follows: [FIELD_NAME =FIELD_REF FIELD_NAME= FIELD_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME,	Cx	Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference `C1`.
		COLUMN_NAME	This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	which is the name of the field in temporary storage. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF.	'literal value'	This means a literal value within the single quotes should be passed in.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`list=`	This is used to enable work list capability for this column. You may optionally populate the `listdesc=` mnemonic to override the text that will be placed in the worklist zone.	`true`	Setting `list=true` will cause the work list icon to appear in the column's header. If a user clicks the column, it will populate all the rows in the output into the work list zone.

Note

In the case of the zone type InfoDataExplorer-M

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			multipleSQLs(F1-DE),theoutputmaybeshowingaunionoftheresul

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			tsoftmultipleSQLstatements.Inthiscase,ifsomeoftheSQLstate

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments

mentsconfigureagivencolumnwithlist=true,butnotall,onlyth

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			edata in the cells for the statements that configure this mnemonica

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments

reputintotheworklistwhentheuserclickstheicon.Alsonotetha

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			t w h e n d e t e r m i n i n g w h i c h c o l u m n s s h o u l d h a v e t h e w o r k l i s t i c o n w h e

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			nbuildingthezone,thesystemonlylooksatthecon figurationfort

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			hecolumnsinSQL1.
`listdesc=`	This is an optional mnemonic when using the `list=` mnemonic. It can be used to override the text that is placed in the work list zone.	Cx	Where Cx represents the value of a previous column. For example, if the text to use is in the first column, reference `C1`.
`listbroadcast=`	Indicates that the broadcast information for the column is also to be made available in the work list zone. This means that the work list can be used to broadcast information to a portal in the same manner as a data explorer.	`true`	Use this setting to turn on the feature.

Common Use Cases

This section provides some common configuration that can be copied and pasted for newly created zones.

Copyright © 2026, Oracle and/or its affiliates.

Use Case	Code Snippet
Displaying the description of a lookup value that has been retrieved in the SQL statement. The business service checks for an override label.	**source=BS** **bs='F1-GetLookupDescription'** **label=FLD_NAME** **input=[fieldName='FLD_NAME' fieldValue=FLD_NAME]** **output=description**
Display a foreign key with its information but suppressing the navigation. (This is useful when you have a foreign key that is additional information in a zone but would cause confusion to enable navigation. To display foreign key reference with its information and enabled for navigation, use the source of FKREF as described above.) The first example assumes that you know the FK Reference. The second example uses a business service that receives the maintenance object as input.	**source=BS** **bs='F1-GetForeignKeyReference'** **input=[input/fkReference=FK_REF_CD input/fkValue1=BUS_OBJ_CD]** **output=output/infoDescription** **label=DESCR** **source=BS** **bs='F1-GetFKReferenceDetails'** **input=[input/maintenanceObject='F1-EXTLKUP' input/pkValue1=BUS_OBJ_CD input/pkValue2=C1]** **output=output/infoDescription** **label=DESCR**
Display a foreign key, including a corresponding navigation link, for generic objects that may or may not have a foreign key reference. If the FK reference exists, its information is shown, otherwise not. Standard foreign key services assume all rows have foreign key details. The service script **F1FKInfoOpt** provides the same foreign key details but does not error when foreign key details do not exist. To build the foreign key description and link, the script needs to be used twice; one column gets the navigation details and another gets the description. The latter references the navigation details column and invokes a BPA script to perform the dynamic navigation.	This column gets the navigation details: **source=SS** **ss='F1FKInfoOpt'** **input=[input/fkReference=C9 input/fkValue1=C6 input/fkValue2=C7]** **output=output/navigationOption** **suppress=true** This column constructs the foreign key description and link: **source=SS** **ss='F1FKInfoOpt'** **input=[input/fkReference=C9 input/fkValue1=C6 input/fkValue2=C7]** **output=output/infoDescription** **label=F1_APPSEC_COMP_DESCR** **bpa='F1ApsecNav'** **tempstorage=[navopt=C15 pkField1=C10 pkValue1=C6 pkField2=C11 pkValue2=C7]** **list=true**

Zone Action

Most zone types provided by the product allow for one or more Zone Actions to be defined to appear in the zone header. An action can appear as a hyperlink, icon or button. The action can also be provided as an HTML string.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Zone types also include parameters for actions defined at the zone type level using IMPLEMENTOR_ACTION_n (Action n) parameters. These are rarely used by the product zone types. The actions defined here override any actions defined on the zone type (if present). The details below apply to the zone type level actions as well.

A zone action is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
`type=`	This mnemonic defines the appearance of the action in the zone header.	**LINK**	Indicates that the action is shown as a textual hyperlink.
		ICON	Indicates that the action is shown as a graphical icon.
		BUTTON	Indicates that the action is shown as an HTML button.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		ASIS	Indicates that the parameter will provide the HTML to be used for the action.
`action=`	This mnemonic defines the action to take when the link/icon/button is clicked. This is ignored when the type=ASIS.	**NAVIGATION**	Indicates that the action is navigation to a page.
		SCRIPT	Indicates that the action is to run a BPA script.
`navopt=`	Defines the navigation option to use when the `action=NAVIGATION`.	'NAV_OPT_CD'	Enter a reference to a valid navigation option in single quotes.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
bpa=	Defines the script to run when the action=SCRIPT.	'SCRIPT_CD'	Enter a reference to a valid BPA script in single quotes.
icon=	Indicates the icon to use when type=ICON.	DISP_ICON_CD	Enter a reference to a valid display icon.
		'path'	Enter an explicit path to the icon, for example 'images/gotoZone.gif'.
asis=	This is required when the type=ASIS. This provides the ability to precisely define the HTML you wish to have included in the header. All valid HTML is permitted including the use of "ora" css classes and JavaScript functions.	['HTML']	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
label=	By default, the label or tooltip will come from the navigation option or BPA script description. Use this mnemonic to override that label.	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.
		'text'	Enter the text directly in single quotes.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`context=`[target1=source1 target2=source2]	This is used to pass context data when navigating to a page or executing a BPA script. The mnemonic supports passing multiple values. In each case the target context field or BPA script variable is defined first followed by an equal sign, followed by source data defined using one of the valid values defined in the next column. One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets.	FIELD_NAME	Indicates that the value should be taken from the field with this name from portal context, global context or the page data model. The mnemonic sourceLoc is used for defining the source.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		xpath	Indicates that the value should be taken from a schema field, represented by the Xpath, displayed in this zone. This is valid when the zone is displaying a UI Map.
		'constant'	Indicates that the value defined in single quotes should be passed.
sourceLoc=	This mnemonic defines the source of the FIELD_NAME's value in the context mnemonic. If this mnemonic is left blank, the default behavior is as follows: - The portal context is checked.	G	Indicates that the field's value is retrieved from the global context.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	- If no portal context value is found, the global context is checked. - If neither value is available, the field is ignored.	P	Indicates that the field's value is retrieved from the portal context.
		D	Indicates that the field's value is retrieved from the page data model.
class=	Use this mnemonic to override the look and feel of the link / icon / button using a different CSS style.	'className1' 'className2'	Enter one or more classes in single quotes. Multiple class names may be provided.
style=	Use this mnemonic to override the look and feel of the action element using the indicated css style.	Standard style= format.	All allowed css style definitions may be used.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
vss=	This mnemonic, along with **vinput**= and **voutput**=, may be used to conditionally hide the action. Use this mnemonic to define the service script responsible for determining wherher the action should be visible or not.	'SCRIPT_CD'	Enter a reference to a valid service script in single quotes.
vinput=[target1=source1 target2=source2]	This is used to pass input data to the visibility script. The mnemonic supports passing multiple values. In each case the target field is defined first followed by an equal sign, followed by source data defined using one of the valid values defined in the next column.	FIELD_NAME	Indicates that the value should be taken from the field with this name from portal context.
	One or more input values may be defined. Each value is defined separated by spaces. The whole set of values should be surrounded by square brackets.	'constant'	Indicates that the value defined in single quotes should be passed.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`voutput=`	Use this mnemonic to define the XPath name of a Boolean element set by the visibility script indicating whether to show the action or not.	xpath	If the visibility script sets the element to true the action is visible. If it is set to false the action is not visible.

Examples:

- **type=BUTTON action=SCRIPT bpa='F1-SET-USER' context=[USER_ID=USER_ID] label=UPDATE_LBL**
- **type=LINK action=NAVIGATION navopt='gotoUser' context=[USER_ID=path(schema/userdId)]**
- **type=LINK … vss='F1BtrActVis' vinput=[batchControl=BATCH_CD batchNumber=BATCH_NBR batchRerunNumber=BATCH_RERUN_NBR batchRunAction='CLOSE_PARMS'] voutput=showAction**
- **type=ASIS asis=['<A class="oraLink" href="www.google.com">Search']**

> **Note**
>
> If the zone type has actions defined and there is a desire to simply remove the zone type actions, the Zone Action can be set with the following configuration: **type=ASIS asis=[]**

User Filters

Data explorer zones include the ability to define User filters to allow a user to enter data to restrict the zone's rows and / or columns. The filters may be defined individually using User Filter parameters 1–25. Alternatively, a UI map may be defined for capturing filters. In this case, the map's input fields must be associated with the zone's filters by specifying the **xpath=** mnemonic on the respective User Filter parameters.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (`F1-DE`)

Copyright © 2026, Oracle and/or its affiliates.

- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`)
- Info Data Explorer - Single SQL (`F1-DE-SINGLE`)

A user filter is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
`name=`	This mnemonic is used if the zone's filter should be pre-populated with a value from global context, portal context or broadcast from another zone.	MD Field Name	
`datasource=`	This mnemonic defines the source of the filter's pre-populated value defined in the name mnemonic. If this mnemonic is left blank, the default behavior is as follows: - If the field has been broadcast from another zone, the broadcast value is used. - If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken). - If still no value, the global context is checked. - If still no value, no default value is shown.	`G`	Indicates that the zone should look for the filter value in global context.
		`P`	Indicates that the zone should look for the filter value in portal context.
		`D`	Indicates that the zone should look for the filter value in the page data model.
`type=`	Defines the visual metaphor used to capture the filter values.	`DATE`	Filters of this type capture a date.
		`DATE/TIME`	Filters of this type capture a date and time.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		STRING	Filters of this type capture a string
		MONEY	Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic.
		NUMBER	Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic.
		LOOKUP	Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic.
		TABLE	Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic.
		CHARTYPE	Filters of this type capture predefined characteristic values for a characteristic type (code and description). This type of filter must also reference the chartype mnemonic.
		ASIS	Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement.
label=	Defines the filter's label that appears in the zone's description bar and in the input area.	MD Field Name	Enter a valid field name whose label should be used for the filter label. This should always be the option used if multiple languages are needed.
		'text'	Defines the text directly.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
cur=	Defines the currency code applied when type=MONEY.	Currency Code	Enter a reference to a valid currency code.
dec=	Defines the number of decimal places when type=NUMBER.	Valid number	It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.
lookup=	Defines the lookup flag whose values appear when type=LOOKUP.	Lookup Field Name	Enter a reference to a valid lookup field name.
table=	Defines the admin table whose values appear when type=TABLE.	Table Name	Enter a reference to a valid control table name.
chartype=	Defines the characteristic type code whose values appear when type=CHARTYPE.	Char Type code	Enter a reference to a valid characteristic type code.
xpath=	This mnemonic is used in conjunction with a Filter Area UI Map. For each filter, you must specify the XPath to the corresponding UI map schema element.	XPath	The type= mnemonic must also be appropriate for the map's input field, otherwise the query's SQL could fail.
likeable=	This mnemonic defines if a likeable search is performed on the entered value when type=STRING.	S	The query will add % to the suffix of the filter value.
		P	The query will add % to the prefix of the filter value.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		PS	The query will add % to the prefix and suffix of the filter value.
`divide=`	The mnemonic controls if a divider line appears above and/or below the filter. Note, you can specify this parameter twice if you want divider lines placed above and below a filter, e.g., **divide=above divide=below**.	`above`	This results in a divider line placed above the filter.
		`below`	This results in a divider line placed below the filter.
`searchField=`	This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map.	MD Field Name	Enter the field name that exactly matches the searchField name specified in the oraSearchField HTML element in the UI map.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`oraSearchFilter`	This mnemonic is only applicable when the zone is used as a foreign key search. It indicates that this filter is the one that the user is prompted for in the auto-filtering functionality. No additional values are needed.		

Note

Be sure that a filter that includes this m

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		nemonicalsodefinea'label='mnemonicsothatttheautto-filteringp	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		rompt shows the label. If no label is defined, the user sees the generer	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		iterm'Filter'.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
`encrypt=`	This mnemonic defines if the user filter is encrypted and needs to be searched by hashed value.	[TBL_NAME,FLD_NAME,WHERE_FLD,WHERE_VALUE]	A valid table name and field name are required. The WHERE_FLD and WHERE_VALUE are optional, but if entered, both are required. Use this to only encrypt the field if another field has a certain value. The following is an example. `encrypt=[CI_PERSON,PER_ID_NBR,ID_TYPE_CD,'SSN']`. The WHERE_VALUE may also reference another filter. The following is an example. `encrypt=[CI_PERSON,PER_ID_NBR,ID_TYPE_CD,F1]`. The field referenced should be the source value u

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			e of the field. However, the SQL should use the hashed value in its filt

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
			er.

Examples:

- `label=F1_NBR_DAYS type=NUMBER`
- `label=F1_SHOW_ALL_REQ_FLG type=LOOKUP lookup=F1_SHOW_ALL_REQ_FLG`
- Filter value where a Filter UI Map is defined and Description is one of the filters. `type=STRING xpath=description likeable=S`
  - `type=STRING label=DESCR likeable=S divide=below`
  - `label=REQ_TYPE_CD type=TABLE table=F1_REQ_TYPE`

Hidden Filters

Data explorer zones include the ability to define Hidden filters to restrict the rows and / or columns that appear in the zone. The following are the potential sources of a hidden filter's value:

- The global area contains the fields whose values are maintained in global context.
- The portal area contains the fields describing the object currently displayed in a portal.
- Other zones on a portal can broadcast information to the portal area, which can then in turn be used by the zone as a hidden filter.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (`F1-DE`)
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`)
- Info Data Explorer - Single SQL (`F1-DE-SINGLE`)

A hidden filter is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
`name=`	This mnemonic defines the name of the field that needs to be broadcast from other zones or populated in the portal context	FIELD_NAME	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
datasource=	This mnemonic defines the source of the hidden filter's value. If this mnemonic is left blank, the default behavior is as follows: - If the field has been broadcast from another zone, the broadcast value is used. - If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken). - If still no value, the global context is checked. - If still no value, the zone appears as per the poprule mnemonic.	G	Indicates that the zone should look for the filter value in global context.
		P	Indicates that the zone should look for the filter value in portal context.
		D	Indicates that the zone should look for the filter value in the page data model.
poprule=	This mnemonic controls what happens if the hidden filter is not present.	R	Indicates that a value for the filter is required. The zone will be set to the "empty state" and the "please broadcast" message will appear in the zone. This is the default value.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		0	Indicates that the value is optional. If no value is required, the zone is still built without that value.
type=	Defines the visual metaphor used to capture the filter values.	**DATE**	Filters of this type capture a date.
		DATE/TIME	Filters of this type capture a date and time.
		STRING	Filters of this type capture a string
		MONEY	Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic.
		NUMBER	Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic.
		LOOKUP	Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
		TABLE	Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic.
		CHARTYPE	Filters of this type capture predefined characteristic values for a characteristic type (code and description). This type of filter must also reference the chartype mnemonic.
		ASIS	Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement.
`label=`	Defines the filter's label that appears in the zone's description bar.	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.
		'text'	Defines the text directly.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
cur=	Defines the currency code applied when type=MONEY.	CURRENCY_CD	Enter a reference to a valid currency code.
dec=	Defines the number of decimal places when type=NUMBER.	N	It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.
lookup=	Defines the lookup flag whose values appear when type=LOOKUP.	LOOKUP_FIELD_NAME	Enter a reference to a valid lookup field name.
table=	Defines the admin table whose values appear when type=TABLE.	TABLE_NAME	Enter a reference to a valid admin table name.
chartype=	Defines the characteristic type code whose values appear when type=CHARTYPE.	CHAR_TYPE_CD	Enter a reference to a valid characteristic type code.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
searchField=	This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map.	FIELD_NAME	Enter the field name that exactly matches the searchField name specified in the oraSearchField html element in the UI map.

Multi-Select Action

This parameter defines an action to be included in the action area for multi-selection processing. Note that a multi-selection action can only be used if the Multi Select parameter has been set to YES, which causes a checkbox to appear on each row displayed. The action defined here will trigger against all rows selected by the user via the checkbox.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (F1–DE)
- Query Data Explorer - Multiple SQLs (F1–DE-QUERY)
- Info Data Explorer - Single SQL (F1–DE-SINGLE)

A multi select action has the following mnemonics:

Mnemonic	Description	Valid Values	Comments
script=	This mnemonic defines the script to be invoked when the action is clicked. This is required.	SCR_CD	Enter a reference to a valid BPA script or Service Script.
type=	This mnemonic	**BUTTON**	The action is rendered as a button. This is the default.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	defines how the action should be rendered.	**LINK**	The action is rendered as hypertext.
		ICON	The action is rendered as a graphic icon. For this option, the icon mnemonic is required.
`icon=`	This mnemonic defines the icon to display when `type=ICON`.	DISPLAY_ICON_CD	Enter a reference to a valid display icon code.
`refresh=`	This mnemonic indicates how and if a refresh should occur after the script completes.	**NO**	Indicates that no refresh is performed. This is the default.
		ZONE	Indicates that a refresh of the zone is performed.
		PORTAL	Indicates that a refresh of the entire portal is performed.
`label=`	By default, the button label, link text or icon toolti	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	p will come from the script description. Use this mnemonic to override that label.	'text'	Enter the text directly in single quotes.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
list=	When executing the script, the framework builds an XML list containing information from each row selected. This list must be defined in the script's schema and referenced in this mnemonic.	listElementName	Enter a valid list element name from the script schema.
context=[elementName1=rowData1 elementName2=rowData2]	This mnemonic is used to populate the list with the appr	Cx	Indicates that the element should be populated with a value in the referenced column parameter.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	opriate information from each selected row. The mnemonic supports passing multiple values. In each case the element in the schema list is defined first followed by an equal sign, followed by information about the data used to populate the element defin	Px	Indicates that the element should be populated with a value in the referenced post processing parameter.
		COLUMN_NAME	Indicates that the element should be populated with a value from a column in the SQL statement.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	ed using one of the valid values defined in the next column. One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets. Example of a	'constant'	Indicates that the value defined in single quotes should be passed.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	schema: <schema> <accountInfo type="list"> <accountId/> <name/> <amount/> <process/> </accountInfo> </schema> Example of		

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
	list and context mnemonics. `list=accountInfo` **context=[accountId=ACCT_ID name=C2 amount=P3 process='O']**		
`class=`	Use this mnemonic to override the look and feel of the action using a different CSS style.	'className1' 'className2'	Enter one or more classes in single quotes to be appended to the standard class(es). Multiple class names may be provided.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Valid Values	Comments
style=	Use this mnemonic to override the look and feel of the action element using the indicated css style.	Standard style= format.	All allowed css style definitions may be used.

Data Explorer Graph Configuration

For some data explorer zones, it is more effective to display the results in a graphical form such as a pie or line chart. Visualization option parameters enable the graph configuration to be integrated with the data explorer zone that retrieves the data.

The visualization option parameters support three options:

- **Grid** - displays the data in a standard data explorer grid
- **Chart** - displays the data in the form of a chart
- **Map** - provides the ability to invoke a UI Map to display the data

Up to five visualization option parameters may be specified. One of the options must be the standard grid, allowing for a maximum of four graphical options. The zone will display icons corresponding to the visualization types to allow you to switch between the various views. These icons are presented in the same order are they are defined in the parameter list.

> **Note**
>
> If the only display option is a standard grid, it is not necessary to specify the grid as a visualization parameter.

Visualization option parameters are provided on the Info Data Explorer - Single SQL (`F1-DE-SINGLE`) and Info Data Explorer - Multiple SQLs (`F1-DE`) zone types.

Visualization Option Parameter Configuration

Mnemonics are used to define the attributes for an option type within the parameter value.

Copyright © 2026, Oracle and/or its affiliates.

For a **grid**, the mnemonic is ‘optiontype=GRID’. No other mnemonics are required.

For a **map**, the mnemonics are ‘optiontype=MAP map=MAP_CD schemaPath=xpath’, where ‘MAP_CD’ is the ID of the UI Map and ‘xpath’ is the schema path of the UI Map. In most use cases the schema elements will match the result set from the data explorer, in which case no schemaPath mnemonic is needed. If the UI Map schema has additional elements present, the schemaPath attribute will define the group node that the data explorer information will be copied into.

For a **chart**, the option type mnemonics is ‘optiontype=CHART’. This is followed by the appropriate chart configuration attributes, which are described in the section below.

Chart Configuration

The following attributes are used to configure a graphical representation of a set of data.

The ‘`oraChart:`’ attribute defines the type of graph to display and its general configuration. The set of configuration parameters available for this attribute are as follows:

Parameter	Values	Description

`type:`	`pie`
	`stacked`
	`cluster`
	`line`
	`area`
	`combo`
		Defines the type of chart to display.
		Required

`showLegend:`	`true` `false`	Defines if the chart should have a legend displayed. Optional (default is **true**)

`legendPosition:`	`left`
	`right`
	`bottom`
	`top`
		Defines where the legend should appear.
		Optional (default is **right**)
		Setting position to **left** or **right** will automatically render it vertically.
		Setting position to **top** or **bottom** will automatically render it horizontally.

`legendBorder:`	`true` `false`	Defines if the legend should display with a border around it. Optional (default is **false**)
`depth:`	`true` `false`	A value of **true** indicates a 3 dimensional depth for the chart. Optional (default **false**, which is a 2 dimensional chart)
`animate:`	`true` `false`	A value of **true** indicates that the graph should animate when displayed. Optional (default is **true**). When using large data sets, consider disabling animation.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`dataCursor:`	`on` `off`	A value of **on** enables hovering anywhere in the graph. Optional (default is **off**). It is not applicable to pie charts.
`orientation:`	`horizontal`	Defines the chart orientation. This only applies to bar, line, area, combo charts. E.g., `oraChart="type:cluster;orientation:horizontal"`, defines horizontal cluster chart. Optional (default is vertical).

The **oraChartSeries** attribute defines the source information for the graph. Most charts support 5 of these attributes, with the following mnemonics:

- `oraChartSeries1`
- `oraChartSeries2`
- `oraChartSeries3`
- `oraChartSeries4`
- `oraChartSeries5`

Stacked charts support an unlimited number of series by continuing to add attributes `oraChartSeries6` and above, but beware of performance implications and memory limits when using an excessively high number of series. All attributes are identical in format and accept the same parameters, as described below.

> **Note**
>
> All the charts require `oraChartSeries1`. Stacked, Cluster and Line charts may optionally include the additional series attributes (for multiple bars/lines).

If you define multiple series, then data must be provided for all series defined. The data amounts can be 0 (zero) but they have to be present in order for the chart to display correctly.

The set of configuration parameters available for the **oraChartSeriesN** attribute are:

Parameter	Values	Description
`list:`	XPath value	Defines the XPath to the list in the XML that contains the data to chart. Required only for HTML definitions in UI Maps. Data explorer values are derived from the columns for the zone.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`amount:`	Cn Pn XPath value	Defines the element containing the amount to chart. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. Required
`xaxis:`	Cn Pn XPath value	Defines the element containing the x-axis data. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element is specified. Required for Stacked, Cluster, Line, Area and Combo charts.

`xaxisFormat:`	`date`
	`dateTime`
	`time`
	`localDate`
	`string`
		Defines the x-axis data format.
		If it is **date**, **dateTime** or **time** then the value is presented in the format as defined on the user's display profile.
		In case of **localDate** or **string** the data is displayed as is with no special formatting.
		Optional (Default is **date**).

`label:`	Text value	Defines the label for the amount being charted. Either this setting or `labelPath:` must be defined.
`labelPath:`	Cn Pn XPath value	Defines the element that provides the label for the amount being charted. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. Either this setting or `label:` must be defined.
`currency:`	A valid Currency code	Defines the currency code for the amount being charted. Optional.
`currencyPath:`	Cn Pn XPath value	Defines the element that provides the currency code for the amount being charted. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. Optional.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`hoverText:`	Text value	Defines the hover text for the chart element. Optional (A default hover text is always available.) Ignored if `hoverTextPath:` is defined. The following substitution variables are available. • `$label` This will be replaced with the label text as determined by the `label:` or `labelPath:` setting. • `$amount` This will be replaced with the amount text as specified by the `amount:` setting. • `$axis` This will be replaced with the x-axis text. • $% This will be replaced by the percentage "slice" of the pie or bar. • `$newline` This will force a line break. If the hover text defined contains any of the above values, they will be replaced by the equivalent text prior to being displayed. Example: `"hoverText:$label$newline$amount"`
`hoverTextPath:`	Cn Pn XPath value	Defines the element that provides the hover text for the chart element. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. The hover text in the element can utilize all the substitution functionality described above in the `hoverText:` description. Optional.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description

`type:`	`bar`
	`line`
	`area`
		This attribute is used for the **combo** chart type only. It defines how each series on the **combo** chart should be presented.
		The following example defines a **combo** chart where one series is rendered as bars and another one as area.
		
		`oraChart="type:combo;" oraChartSeries1="xaxis:C3; label:C2; amount:C1;type:bar" oraChartSeries2="xaxis:C3; label:C2; amount:C1;type:area"`

`navOpt:`	A valid Navigation Option code.	Defines a navigation option to be activated when the chart element is clicked. Optional.
`navOptPath:`	Cn Pn XPath value	Defines the element that provides the navigation option to be activated when the chart element is clicked. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. Optional Note that both `navOpt:` and `navOptPath:` may be configured. The `navOptPath:` navigation option is processed first. If a value is found it is used, otherwise the value in the `navOpt:` setting is used. This means that the chart definition can include a default navigation option and a navigation option present in the data will override it.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`key:`	Cn Pn XPath value	Defines the element that contains the key field data to be used in a navigation option. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. This is required if either `navOpt:` or `navOptPath:` is defined. **Note** Only one key field can be configured for a navigation option.
`script:`	A BPA script name	Defines a BPA script to be activated when the chart element is clicked. Optional When a script is executed, all the elements from the chart data will be made available to the script as temporary variables.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`scriptPath:`	Cn Pn XPath value	Defines the element that provides the BPA script to be activated when the chart element is clicked. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. Optional Note that both `script:` and `scriptPath:` can be configured. The `scriptPath:` script option is processed first. If a value is found it is used, otherwise the value in the `script:` setting is used. This means the chart definition can include a default script and a script present in the data will override it.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description
`color:`	HTML Color code / RGB value	Optional (default colors applied) Defines the color for the series. The format is a valid HTML color code, e.g. **green** or **blue**. All valid color names are defined in this link: http://www.w3schools.com/htmL/html_colornames.asp. Alternatively an RGB format may be used. (`00FF00` is green and `0000FF` is blue) **Note** Refer to Color Contrast for information about the use of the HTML color 'red' and its impact on accessibility.
`colorPath:`	Cn Pn XPath value	Defines the element containing a color for the series. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified. The valid formats as described in the `color:` setting apply. Optional (default colors applied)

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Values	Description

`pieColors:`	HTML color code / RGB values	Defines the colors for the pie series. Any number of HTML color codes or RGB color values can be specified, separated by spaces. Examples:
		
		`pieColors: green blue`
		
		`pieColors: 00FF00 0000FF`
		
		Optional (default colors are applied if the series exceeds the values specified)

The `oraChartBroadcast=`attribute provides the ability to broadcast a value from a chart to the portal context. An unlimited number of fields can be broadcast, configured as name/value pairs; Broadcast is triggered by clicking on a corresponding section of the chart.

The syntax is as follows where FIELD_NAMEn is the name of the portal context field to be broadcast and Cn (or Pn) is the element that contains the data value to be broadcast.

```
oraChartBroadcast="FIELD_NAME1:C1;FIELD_NAME2:C2;"
```

For HTML definitions in UI Maps, ‘xpathn’ is the element in the list schema that contains the data value to be broadcast.

```
oraChartBroadcast="FIELD_NAME1:xpath1;FIELD_NAME2:xpath2;"
```

The `oraChartViewDetails=`attribute provides the ability to launch a view details drawer for an entry in a chart. The syntax is similar to launching a drawer from a UI map or a data explorer column but without rendering an icon. Refer to Launch a View Details Drawer for more information.

UI map examples:

```
oraChartViewDetails="detailPanelName:VIEWDET; USERID:x[user];"
oraChartViewDetails="detailPanelName:x[panelName]; USERID:x[user];"
```

Data explorer examples:

```
oraChartViewDetails="detailPanelName:VIEWDET; USERID:C1;"
oraChartViewDetails="detailPanelName:C2; USERID:C1;"
```

Zone Configuration Examples

Zone with a simple pie chart

In this example, the zone SQL summarizes To Do entries by Role. The column definitions are:

- C1 - role description

Copyright © 2026, Oracle and/or its affiliates.

- C2 - count of To Do entries by role
- C3 - role ID

Visualization Option 1

This option shows To Do entries by Role as a pie chart. The role ID will be broadcast to the portal when the corresponding section of the pie chart is clicked.

```
optiontype=CHART
oraChart="type:pie;"
oraChartSeries1="labelPath:C1; amount:C2;hoverText:$label$newline$amount;"
oraChartBroadcast="ROLE_ID:C3"
```

Zone with two chart types

In this example, the zone SQL summarizes To Do entries by Priority and also calculates separate counts by To Do status within priority. The column definitions are:

- C1 - priority description
- C2 - count of To Do entries by priority
- C3 - priority flag value
- C4 - count of Open To Dos by priority
- C5 - count of Assigned To Dos by priority
- C6 - Label for the Open To Dos bar in the cluster chart
- C7 - Label for the Assigned To Dos bar in the cluster chart

Visualization Option 1

This option shows To Do entries by Priority as a pie chart. The priority value will be broadcast to the portal when the corresponding section of the pie chart is clicked.

```
optiontype=CHART
oraChart="type:pie; legendBorder:true;"
oraChartSeries1="labelPath:C1; amount:C2;hoverText:$label$newline$amount;"
oraChartBroadcast="TD_PRIORITY_FLG:C3"
```

Visualization Option 2

This option shows To Do entries by Priority as a cluster chart with two data series, one for Open To Dos and another for Assigned To Dos . The corresponding priority value will be broadcast to the portal when clicking either the Open or Assigned bar of the cluster chart.

```
optiontype=CHART oraChart="type:cluster;  showLegend:true;"
oraChartSeries1="xaxis:C1; xaxisFormat:string; labelPath:C6;
amount:C4;hoverText:$label$newline$amount;"
oraChartSeries2="xaxis:C1; xaxisFormat:string; labelPath:C7;
amount:C5;hoverText:$label$newline$amount;"
oraChartBroadcast="TD_PRIORITY_FLG:C3"
```

Pagination Configuration

Copyright © 2026, Oracle and/or its affiliates.

The various data explorer zones in the product support the ability to configure pagination so that a user can 'page through' a large set of results using "previous" and "next" buttons or links.

There are several zone parameters that are impacted when attempting to configure this functionality. The following steps highlight the configuration.

- The **Enable Pagination** parameter must be configured to define the basic setup for pagination for the zone. This parameter defines whether the "previous" and "next" actions are defined as buttons, links or icons and indicates the location of the actions. It also allows an indication as to whether the additional rows are simply appended, rather than shown in a new "page". Refer to the parameter's embedded help for information about the specific syntax.
- It is recommended that the zone is configured with record count and page information by properly configuring the **Record Count Display** parameter. Refer to the parameter's embedded help for information about the specific syntax.
- Configure the **Number of Rows to Retrieve for SQL**. parameter to define the number of records displayed per page. If this parameter is not specified the value in the **Number of Rows to Display** parameter is used.
- Configure the key that will be used for paging so that the system can keep track of the 'page break'. The data must be sorted by the paging key; as a result, the decision for identifying the paging key must take into account the design for the zone and the data being displayed. In addition, the paging key must be unique to ensure that the page breaks occur correctly. See below for configuration examples.
  - The **SQL Statement** must includes additional clauses **PAGENEXT** and **PAGEPREV** based on the paging key. In addition, as mentioned above, the paging key must be used in the ORDER BY clause.
  - The SQL **Column** parameters must define the paging key mnemonic to be used in conjunction with the SQL statement paging clauses.
  - It is recommended to configure the **SQL Display Column** parameter to show the data in the same order as the ORDER BY clause.

The following zone types support this capability:

- Info Data Explorer - Single SQL (`F1-DE-SINGLE`).
- Info Data Explorer - Multiple SQLs (`F1-DE`). Note that zones of this type support a union of the results of all the SQL statements. As a result, pagination may only be enabled for zones of this type if a single SQL is used. The system is not able to keep track of the pagination across disparate SQL statements.
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`).
- Multi Query Data Explorer (`F1-DE-MULQRY`). Zones of this type do not include configuration for SQL statements or column display. However, they do include configuration for the **Enable Pagination**. This parameter must be configured in order for pagination on the individual zones to work.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Zones used for a Business Service.** Note that pagination is ignored when invoking a data explorer zone via a business service. In this scenario, the zone will return the first "chunk" of rows as defined by the Number of Rows parameters.
>
> **Zones with data visualization options.** Pagination is not supported for zones that define visualization options to show data in a graphical form.

Examples

Simple Paging Key

In this example, the Extendable Lookup Value is defined as Column 1 (C1) and is marked as the paging key. This field is unique for the table and works well as a simple paging key.

```
SELECT A.F1_EXT_LOOKUP_VALUE,A.BUS_OBJ_CD
FROM
  F1_EXT_LOOKUP_VAL A,
  F1_EXT_LOOKUP_VAL_L B
WHERE
A.BUS_OBJ_CD = :H1
AND A.BUS_OBJ_CD = B.BUS_OBJ_CD
AND A.F1_EXT_LOOKUP_VALUE = B.F1_EXT_LOOKUP_VALUE
AND B.LANGUAGE_CD = :LANGUAGE
[(F1) AND UPPER(A.F1_EXT_LOOKUP_VALUE) like UPPER(:F1)]
[(F2) AND ((UPPER(B.DESCR_OVRD) like UPPER(:F2))
OR (B.DESCR_OVRD = ' ' AND UPPER(B.DESCR) like UPPER(:F2)))]
[(PAGENEXT) AND A.F1_EXT_LOOKUP_VALUE > :C1]
[(PAGEPREV) AND A.F1_EXT_LOOKUP_VALUE < :C1]
ORDER BY A.F1_EXT_LOOKUP_VALUE
```

Complex Paging Key

Most queries however do not sort by a unique value. In this case, the paging key needs to be set based on the sorting of the query and should include a unique field, such as the primary key, as the last paging key. In this example, the query is showing results sorted by To Do Type, Role and User. All fields, including the To Do Entry ID (the primary key) are marked as paging keys.

```
SELECT TD_TYPE_CD, ROLE_ID, ASSIGNED_TO, ASSIGNED_DTTM, TD_PRIORITY_FLG,
TD_ENTRY_ID
FROM  CI_TD_ENTRY
WHERE
ENTRY_STATUS_FLG IN ('O', 'W')
[(F1) and TD_TYPE_CD = :F1]
[(F2) AND ASSIGNED_TO = :F2]
[(F3) AND ROLE_ID = :F3]
[(PAGENEXT) and ((TD_TYPE_CD>:C1) or (TD_TYPE_CD=:C1 and ROLE_ID>:C2) or
(TD_TYPE_CD=:C1 and ROLE_ID=:C2
and ASSIGNED_TO>:C3) or (TD_TYPE_CD=:C1 and ROLE_ID=:C2 and ASSIGNED_TO=:C3
AND TD_ENTRY_ID>:C4))]
[(PAGEPREV) and ((TD_TYPE_CD<:C1) or (TD_TYPE_CD=:C1 and ROLE_ID<:C2) or
(TD_TYPE_CD=:C1 and ROLE_ID=:C2
```


Copyright © 2026, Oracle and/or its affiliates.

```
and ASSIGNED_TO<:C3) or (TD_TYPE_CD=:C1 and ROLE_ID=:C2 and ASSIGNED_TO=:C3
AND TD_ENTRY_ID<:C4))]
ORDER BY TD_TYPE_CD, ROLE_ID, ASSIGNED_TO, TD_ENTRY_ID
```

Auto-filtering Configuration

Auto-filtering is the term used to describe functionality available in search fields where a user can type search text without having to open the search. Refer to Input Field with Search for examples of this feature. This section describes how to enable this feature and how to configure a search zone to enable this functionality.

The feature is opt-in. To turn on the feature, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type `Custom Look and Feel`. If one does not exist, create a feature configuration of this type. Select the `Enable Auto-Filtering` option and set the value to 'Y'.

The feature supports designating a single user filter in the zone as the one to expose to the user for auto-filtering. For example, when searching for a control table record, like To Do Type, the system configures the Description filter as the one that supports auto-filtering. For an object like Person, the person name filter may be the one that is configured for this feature.

To indicate the filter to support auto-filtering, add the mnemonic `oraSearchFilter` to the filter definition. For example, here is the configuration for the description user filter in the To Do Type search zone.

```
type=STRING label=DESCR likeable=S xpath=descr oraSearchFilter
```

In addition, when turning on this feature, the system supports displaying up to 3 columns of content for each record to help the user choose the correct value. The best practice is ensure that at least one of the columns configured to display matches the filter the user is typing. For example, the system has configured that all control table searches show both the code and the description. This configuration goes into the `Initial Display Columns` parameter, or for Query data explorers, if individual SQLs have different display columns, add the configuration to the appropriate `SQL n Display Column` parameter. The syntax is the mnemonic `oraSearchDisplay=` followed by a comma delimited list of up to 3 column references in the order to display. Note that the columns in the search display mnemonic are limited to those referenced in the display parameter. Example:

```
C1 C2 C3 C4 C5 oraSearchDisplay='C2,C1'
```

Use Data Explorer for Derived Data

There are times when a design warrants displaying data in a data explorer zone that is not accessible via SQL. For example, perhaps the data is from another system and it requires a web service call. The JMS Message Browser is another example.

The product provides functionality in the data explorer that allows you to call a script after the user filters are populated but before the SQL is executed. The script can retrieve the data as appropriate, store the data in table format so that the SQL can retrieve the data from the table.

The following points provide more detail:

- Create a service script that retrieves the data as needed. This script should store the retrieved data in a temporary table.

Copyright © 2026, Oracle and/or its affiliates.

  - The product provides a table that may be used. It is called F1_GENERIC_GTT (Generic Global Temporary Table). There is a business service — Create Global Temporary Table Records (`F1-InsertGTTRecords`) that the service script may call to insert the records.
  - Note that if the data is accessed via a web service call, it may be appropriate to execute the web service in a separate session using the business service F1-ExecuteScriptInNewSession to trap errors that may be issued by the web service call and provide a better error.
- In the data explorer zone use this service script in the zone's pre-processing script parameter. If any user or hidden filters should be passed into the script, the parameter supports mnemonics for this purpose. Refer to the parameter's embedded help for the supported syntax.
- The SQL for the data explorer should access the temporary table that was populated by the service script.

Configuring Timeline Zones

This topic highlights information related to configuring a timeline zone. The zone type is `F1-TIMELINE`. A timeline zone contains one or more "lines" where each line shows when significant events have occurred. The output of each line is driven by an algorithm configured on a timeline zone. Each algorithm is responsible for retrieving a single type of data. For example, one algorithm may retrieve bills for an account in a given time period whereas another algorithm may retrieve payments for that account for the same time period.

The algorithms to configure for a timeline zone use the **Zone - Timeline** plug-in spot. Please note the following details about the behavior for algorithms for this plug-in spot.

- The timeline algorithm receives all the global context values currently populated. In addition, it receives a start and end date from the zone, based on the time period chosen by the user, along with the maximum number of events that can be reasonably displayed for the chosen time period. The algorithm should use this information to retrieve data for a given type of transaction related to one or more of the input context values for the provided time period.
- For each event found, the algorithm returns information about the event along with many options that assist the user in getting more detail about each event or acting on an event.
  - Event date
  - Primary key of the record (key / value pairs)
  - FK Reference. With this information, the timeline zone will display the appropriate info string to display in the zone's info area when clicking on the event. In addition, the FK reference identifies the appropriate navigation option to use when a user clicks the info string hypertext to view the record on its maintenance page.
  - Background Color and Text Color to use for the event. (Obsolete). This information is no longer used by the timeline zone and will be ignored.
  - Icon use for the event. (Obsolete). This information is no longer used by the timeline zone and will be ignored.
  - BPA script to launch when a user clicks on an event. (Optional). The algorithm may return one or more BPA scripts that a user may launch to act on an event. For example, for an event that has a status of **Error**, perhaps a BPA is provided to walk a user through resolving the error.

Copyright © 2026, Oracle and/or its affiliates.

When a script is initiated from a timeline, the system puts the prime key of the event into a field in the page data model. The name of the field is the column name(s) of the event's prime key. For example, when a script associated with a payment event is kicked off, the system populates a field called PAY_ID with the prime-key of the selected payment.

Note that your specific edge application may supply algorithm types for a timeline zone as part of the base product. Although algorithm types may be provided, typically the product does not deliver algorithms because the parameters for the algorithms are driven by a particular implementation's business rules and preferences. As a result, the product will also not deliver pre-configured timeline zones. Please refer to your edge application's documentation for more information about what timeline algorithm types are delivered, if any and recommendations for configuration.

Defining Context-Sensitive Zones

A context-sensitive zone allows you to associate a zone with a specific user-interface transaction. A context-sensitive zone appears at the top of the Sidebar when a user accesses a page for which the zone is specified as the context.

> ⚠ **Caution**
>
> Make sure that the zone is appropriate for the transaction on which you are specifying it.

Select **Admin**, and then **Context Sensitive Zone** to maintain context-sensitive zones.

Description of Page

The **Navigation Key** is a unique identifier of a tab page within the system. **Owner** indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**).

> ⚠ **Caution**
>
> Important! When introducing a new context sensitive zone, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The grid contains the list of context-sensitive zones and the sequence in which they appear in the sidebar for the selected navigation key. The grid contains the following fields:

- **Zone** is the name of the zone to display in the Sidebar.
- **Sequence** is the sequence in which the zone is displayed (if multiple context-sensitive zones are defined).
- **Owner** indicates if this context sensitive zone is owned by the base product or by your implementation (**Customer Modification**).

Where Used

A context-sensitive zone displays at the top of the Sidebar whenever a user accesses the transaction for with the zone is specified.

Copyright © 2026, Oracle and/or its affiliates.

Defining Portals

This portal is used to view and maintain the configuration of portals. An implementation may define their own portals or may override some of the settings for base product provided portals.

You can access the portal from the **Admin Menu**, and then **System**, and then **Portal**. You are brought to a query portal with options for searching for a specific record. Once a portal record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding Portals and Zones for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Portal.** This zone provides information about the selected portal definition.
- **Tabs.** This zone lists other tab portals if any are configured for a standalone portal. This zone is only visible when such tabs exist for the current standalone portal.
- **Context Sensitive Zones.** This zone lists context sensitive zones that appear on the sidebar when this portal is displayed . This zone is only visible when such zones exist for the current standalone portal.

The following zones may appear as part of the portal's **Options** tab page

- **Options.** This zone lists various configuration options for the selected portal record.

Some portals may be associated with configurable settings that are maintained on a **Configuration** tab. These settings are defined using a master configuration business object referenced as portal option. The tab is visible when such option exists for the portal.

Defining Menu Options

This portal is used to define / change any menu in the system. You can access the portal from the **Admin Menu**, and then **System**, and then **Menu**. You are brought to a query portal with options for searching for a specific menu. Once a menu has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Updating menus requires technical knowledge of the system. This is an implementation and delivery issue and should not be attempted if you do not have previous experience with menus.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Security and menus.** Refer to Secured Objects and Application Services for a discussion of how application security can prevent menu items (or an entire menu) from appearing.

> **Note**
>
> **Module configuration and menus.** Your module configuration can prevent menu items (or an entire menu) from appearing.

The following zones may appear as part of the portal's **Main** tab page:

- **Menu.** This zone provides information about the selected menu.
- **Menu Lines.** This zone lists the menu's lines and supports actions to add, edit and delete lines. You may also broadcast a line to review more information about its items configuration.
- **Menu Line Items.** This zone lists item information for a selected menu line.

Application security information for the menu can be reviewed and managed from the **Application Security** tab. The following zones may appear as part of this tab page:

- **Menu Lines.** This zone lists the menu's lines. Broadcast a line to review its application security information.
- **Menu Line Security.** This zone lists application services related to the selected menu line. You may use this zone to review and configure security access for the selected menu line and a specific user group.
- **Menu Line Security Details.** This zone lists the entities secured by a selected application service.

The following zone may appear as part of the portal's **References** tab page:

- **Menu References.** This zone lists entities that reference the selected menu.

The following sections provides more information.

Menu Types

The following types of menus are supported:

- **Main** menu type defines the menu items under the Main Menu entry visible when the menu is initially expanded.
- **Admin** menu type defines the menu items under the Admin Menu entry visible when the menu is intially expanded.
- **Submenu** defines a menu group that appears when the main menu or admin menu is selected. For the Admin menu, this is only visible when it's organized functionally. Refer to Admin Menu for more information about options for organizing the admin menu.
- **Context** refers to a context menu.
- **Page Action Menu** defines buttons that appear in the Page Title Area.
- **User Menu** refers to the menu items that appear on the user menu; for example, User Preferences.

Copyright © 2026, Oracle and/or its affiliates.

Extending Product Owned Menus

An implementation may choose to add custom menu lines along with its menu item (or items) to a base product owned menu.

Overriding Menu Lines Sequence

An implementation may override the sequence of a base product owned menu line. Also note that the sequence is defined on the menu line language table, allowing for different orders to be used for different languages (or to let the menu be sorted alphabetically in one language and in a specified order in a different one).

Menu Information is Cached

Use the **Flush Menu** button to flush the cached menu items so you can see any modified or newly created menus. Refer to Caching Overview for more information.

Defining Navigation Keys

Each location to which a user can navigate (e.g., transactions, tab pages, tab menus, online help links, etc.) is identified by a navigation key. A navigation key is a logical identifier for a URL.

- Navigation Key Types
- Navigation Key vs Navigation Option
- The Flexibility of Navigation Keys
- Linking to External Locations
- Overriding Navigation Keys
- Maintaining Navigation Keys

Navigation Key Types

There are three types of navigation keys:

- **System navigation keys** define locations where the target for the navigation is a transaction or portal within the system. The navigation keys define the program component that identifies the page to navigate to.
- **External navigation keys** define a URL that identifies the target location. External URLs can be specified as relative to the product web server or fully qualified. External navigation keys always launch in a new instance of a browser window. Examples of external navigation keys include URLs to external systems.
- **Help navigation keys** define a online help topic that identifies the specific page within the online help to launch. Help navigation keys may be related to a program component when the help is related to a specific page in system.

Navigation Key vs Navigation Option

The system has two entities that work in conjunction with each other to specify how navigation works:

- Navigation Key defines a unique location to which a user can navigate. For example, each page in the system has a unique navigation key. Navigation keys can also define locations

Copyright © 2026, Oracle and/or its affiliates.

that are "outside" of the system. For example, you can create a navigation key that references an external URL. Think of a navigation key as defining "where to go".

- Navigation Option defines how a page is opened when a user wants to navigate someplace. For example, you might have a navigation key that identifies a specific page. This navigation key can then be referenced on two navigation options; the first navigation option may allow users to navigate to the page with no context included, while the second navigates to the page with context data provided to automatically display information related to that context.
- Please note that a wide variety of options can be defined on a navigation option. In addition to defining if data is passed to the page, it could also define search options. In addition, there are some navigation options that do not reference a navigation key but rather refer to a BPA script that should be launched.

The Flexibility of Navigation Keys

Navigation keys provide a great deal of functionality to your users. Use navigation keys to:

- Allow users to navigate to new pages or search programs
- Allow users to transfer to an external system or web page. After setting up this data, your users may be able to access this external URL from a menu, a context menu, their favorite links, etc. Refer to Linking to External Locations for more information.

Refer to the Tool Suite Guide for more information on developing program components.

> **Note**
>
> **Replacing Base-Package Pages or Searches**. If your new page or search has been designed to replace a module in the base-package, the navigation key must indicate that it is overriding an existing navigation key.

Linking to External Locations

If you want to include links to external systems or locations from within the system, you need to:

- Define a navigation key that specifies the URL of the location. For example, define an external navigation key that as a URL of **http://www.oracle.com/**.
- Define a navigation option that specifies from where in the system a user can go to your external location. For example, define a navigation option with a usage of **Favorites** or with a usage of **Menu**. Your navigation option points to the navigation key you defined above.
- Add your navigation option to the appropriate location within the system. For example, have users add the navigation option to their Favorite Links or add the navigation option as an item on a menu.

Overriding Navigation Keys

Your implementation may choose to design a program component (e.g., a maintenance transaction or search page) to replace a component provided by the system. When doing this, the new navigation key must indicate that it is overriding the system navigation key. As a result, any menu entry or navigation options that reference this overridden navigation key automatically navigates to the custom component.

Copyright © 2026, Oracle and/or its affiliates.

For example, if you have a custom On-line Batch Submission page and would like users to use this page rather than the one provided by the system, setting up an override navigation key ensures that if a user chooses to navigate to the On-line Batch Submission from Menu or a context menu, the user is brought to the custom On-line Batch Submission page.

To create an override navigation key, you need to:

- Define a navigation key using an appropriate naming convention.
- If the Navigation Key Type of the navigation key being overridden is **External**, specify a Navigation Key Type of **Override (Other)** and define the appropriate URL Component.
- If the Navigation Key Type of navigation key being overridden is **System**, specify a Navigation Key Type of **Override (System)** and populate the Program Component ID with your custom program component ID.
- Specify the navigation key that you are overriding in the **Overridden Navigation Key** field.

Refer to the Tool Suite Guide for more information about developing your own program components.

Maintaining Navigation Keys

Select **Admin**, and then **System**, and then **Navigation Key** to maintain navigation keys.

Use the **Navigation Key Query** portal to search for an existing navigation key. Once a navigation key is selected, you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! When introducing a new navigation key, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Description of Page

The **Navigation Key** is a unique name of the navigation key for internal use.

Owner indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

Navigation Key Type includes the following possible values:

- **External** indicates that the location is specified in the **URL Component** field.
- **Help** indicates that the navigation key is used to launch online help where the specific help topic is defined in the **URL Component** field.
- **Override (Other)** indicates that the navigation key overrides another navigation key of type **External** or **Help**. For this option, the name of the navigation key being overridden is populated in the **Overridden Navigation Key** field.
- **Override (System)** indicates that the navigation key overrides a system navigation key. For this option, the name of the navigation key being overridden is populated in the **Overridden Navigation Key** field.
- **System** indicates that the navigation key refers to a transaction in the system identified by its program component.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Navigation Key Types for more information about navigation key types.

NOT_SUPPORTED

Refer to Overriding Navigation Keys for more information about settings required to override a system navigation key.

Program Component ID is the name of the program component identified by the key (for system navigation keys). The program component ID can also be used to specify the transaction with which an online help link is associated.

Overridden Navigation Key is the name of the navigation key that the current navigation key is overriding (if **Override (Other)** or **Override (System)** is selected for the **Navigation Key Type**). Refer to Overriding Navigation Keys for more information.

URL Component is the specific URL or portion of the URL for the navigation key (external and help navigation keys only). The URL can be relative to the product web server or fully qualified.

Open Window Options allows you to specify options (e.g., width and height) for opening a browser window for an external navigation key. (External navigation keys always launch in a new browser window.) You can use any valid features available in the Window.open( ) JavaScript method. The string should be formatted the same way that it would be for the features argument (e.g., `height=600,width=800,resizeable=yes,scrollbars=yes,toolbar=no`). Refer to a JavaScript reference book for a complete list of available features.

Application Service is the application service that is used to secure access to transactions associated with **External** navigation keys. If a user has access to the specified application service, the user can navigate to the URL defined on the navigation key. Refer to The Big Picture of Application Security for more information.

The grid displays menu items that reference the navigation key (actually, it shows menu items that reference navigations options that, in turn, reference the navigation key).

Defining Navigation Options

Every time a user navigates to a transaction, the system retrieves a navigation option to determine which transaction should open. For example:

- A navigation option is associated with every menu item. When a user selects a menu item, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every favorite link. When a user selects a favorite link, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every node in the various trees. When a user clicks a node in a tree, the system retrieves the related navigation option to determine which transaction to open.
- Etc.

Copyright © 2026, Oracle and/or its affiliates.

Many navigation options are shipped with the base product and cannot be modified as these options support core functionality. As part of your implementation, you may add additional navigation options to support your specific business processes.

Navigation options may also be used to launch a BPA script.

The topics in this section describe how to maintain navigation options.

> ⚠ **Caution**
>
> In order to improve response times, navigation options are cached the first time they are used after a web server is started. If you change a navigation option and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. A special button has been provided on the Main tab of the navigation option transaction that performs this function. Please refer to Caching Overview for information on the various caches.

- Navigation Option - Main
- Navigation Option - Tree

Navigation Option - Main

Select **Admin**, and then **System**, and then **Navigation Option** to maintain a navigation option.

Use the **Navigation Option Query** portal to search for an existing navigation option. Once a navigation option is selected, you are brought to the maintenance portal to view and maintain the selected record.

Description of Page

Enter a unique **Navigation Option** code and **Description**.

> ⚠ **Caution**
>
> When introducing a new navigation option, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The **Flush System Login Info** button is used to flush the cached navigation options so you can use any modified navigation options. Refer to Caching Overview for more information.

Owner indicates if this navigation option is owned by the base product or by your implementation (**Customer Modification**). This field is display-only. The system sets the owner to **Customer Modification** when you add a navigation option.

> ⓘ **Note**
>
> You may not change navigation options that are owned by the base product.

Use **Navigation Option Type** to define if the navigation option navigates to a **Transaction**, launches a BPA **Script** or opens an **Attachment**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The **Attachment** option type is only applicable to navigation options that are used to launch a file attached to a record in the Attachment maintenance object.

For navigation option types of **script**, indicate the **Script** to launch. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to temporary storage variables available to the script. The script engine creates temporary storage variables with names that match the Context Field names.

For navigation option types of **transaction**, define the **Target Transaction** (navigation key) and optionally a specific **Tab Page** (also a navigation key) if a specific tab on the transaction (other than the Main tab) should be opened when navigating to that transaction.

> **Note**
>
> **Finding transaction navigation keys**. When populating the **Target Transaction** and **Tab Page** you are populating an appropriate navigation key. Because the system has a large number of transactions, we recommend using the "%" metaphor when you search for the transaction identifier. For example, if you want to find the currency maintenance transaction, enter "%currency" in the search criteria.

The additional information depends on whether the target transaction is a fixed page or a portal-based page:

- For portal-based pages:
  - **Navigation Mode** is not applicable and should just be set to **Add Mode**.
  - If navigating to a query portal, by default the query portal will open with the default search option defined. If the navigation should open a different search option, define the **Multi-Query Zone** for that query portal and indicate the **Sub-Query Zone** to open by default. Note that for this configuration, it is common to define **Context Fields** to pre-populate search criteria in the target query zone. When using this configuration, be sure that the target query zone's user filters are defined to populate data from context.
- For fixed pages:
  - **Navigation Mode** indicates if the **Target Transaction** should be opened in **Add Mode** or **Change Mode**.
  - **Add Mode** should be used if the option is used to navigate to a transaction ready to add a new object. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to the transaction when it opens.
  - **Change Mode** is only applicable for fixed pages and should be used if the option is used to navigate to a transaction ready to update an object. You have two ways to define the object to be changed:
    - Define the name of the fields that make up the unique identifier of the object in the **Context Fields** (and make sure to turn on **Key Field** for each such field).
    - Define the **Search Transaction** (navigation key) if you want to open a search window to retrieve an object before the target transaction opens. Select the appropriate **Search Type** to define which search method should be used. The options in the drop down correspond with the sections in the search (where **Main** is the first section, **Alternate** is the 2nd section, **Alternate 2** is the 3rd section, etc.).

Copyright © 2026, Oracle and/or its affiliates.

You should execute the search window in order to determine what each section does.

When you select a **Search Type**, define appropriate **Context Fields** so that the system will try to pre-populate the search transaction with these field values when the search first opens. Keep in mind that if a search is populated with field values the search is automatically triggered and, if only one object is found that matches the search criteria, it is selected and the search window closes.

- **Search Group** is only visible if the **Development Tools** module is not turned off. It is used to define the correlation between fields on the search page and the tab page. You can view a tab page's **Search Groups** by viewing the HTML source and scanning for **allFieldPairs**.

The **Go To Tooltip** is used to specify the label associated with the tool tip that appears when hovering over a **Go To** object. Refer to the **Usage** grid below.

The **Usage** grid defines the objects on which this navigation option is used:

- Choose **Favorites** if the navigation option can be used as a favorite link.
- Choose **Menus** if the navigation option can be used as a user's home page or as a menu or context menu item.
- Choose **Script** if the navigation option can be used in a script.
- Choose **Foreign Key** if the navigation option can be used as a foreign key reference.
- Choose **Go To** if the navigation option can be used as a "go to" destination ("go to" destinations are used on Go To buttons, tree nodes, and hyperlinks).
- If your product supports marketing campaigns, you can choose **Campaign** if the navigation option can be used as a "post completion" transaction on a campaign. For more information refer to that product's documentation for campaigns.

The **Context Fields** grid contains the names of the fields whose contents will be passed to the **Target Transaction** or **Script** or used to launch an **Attachment**. The system retrieves the values of these fields from the "current" page and transfers them to the target transaction or to the script's temporary storage. Turn on **Key Field** for each context field that makes up the unique identifier when navigating to a transaction in **Change Mode**.

> **Note**
>
> For an **Attachment**, the grid should contain the Attachment ID.

> **Note**
>
> **Navigating from a Menu**. The standard followed for many base main **menu** navigation options for fixed transactions is that navigation options launched from the Menu dropdown are configured with no context.

Navigation Option - Tree

This page contains a tree that shows how a navigation option is used. Select **Admin**, and then **System**, and then **Navigation Option** and navigate to the **Tree** tab to view this page.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

The tree shows every menu item, favorite link, and tree node that references the navigation option. This information is provided to make you aware of the ramifications of changing a navigation option.

Navigation to External Pages

The URL Navigation (F1-URLNavigation) extendable lookup defines named external pages to which a user may navigate to from the application. For example, a named code for the Service Request page in Oracle Fusion.

The actual URL for the page is specific to your implementation and as such is not delivered on the extendable lookup record. When applicable, update the entry to reference the URL for the corresponding page. The UI location that references the name external page shows the corresponding URL only when a URL is provided.

Understanding System Messages

All error, warning and informational messages that are displayed in the system are maintained on the message table. Every message is identified by a combination of two fields:

- **Message category number.** Think of a message category as a library of messages related to a given functional area. For example, there is a message category for billing messages and another one for payment messages.
- **Message number**. A unique number identifies each message within a category.

Every message has two components: a brief text message and a long description.

The brief message text supports the %n notation within the message text to cause field values to be substituted into a message. For example, the message text "Unable to assign To Do Entry (%1). It is already assigned to another User (%2)" will have the values of 2 fields merged into it before it is displayed to the user (%1 is the To Do Entry ID, %2 is the current assigned user).

> **Note**
>
> **You cannot change the text provided with the product.** If the message is "owned" by the product, you cannot change the product's message or detailed description. If you want your users to see a different message or detailed description other than that supplied by the product, enter your desired verbiage in the corresponding override fields (and flush the cache).

- Defining System Messages

Defining System Messages

This portal is used to search for message categories and messages that appear throughout the system. An implementation may introduce messages used in custom processes or may choose to override the text for messages delivered by the product.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a message. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding System Messages for more information.

- Defining Message Categories
- Defining Messages

Defining Message Categories

This portal is used to view and maintain a message category and its messages.

> **⚠ Caution**
>
> Message category 80000 or greater must be used to define new messages introduced for a specific implementation of the system. Changes to other Message Text will be overwritten when you next upgrade. If you want to make a change to a Message, drill down on the message and populate the Override Message Text. Note that even for message categories 80000 and higher, message numbers lower than 1000 are reserved for common base product messages.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a specific message. Once a message category has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding System Messages for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Message Category.** This zone provides information about the selected message category.
- **Messages.** This zone lists and maintains messages for the selected category.

Defining Messages

This portal is used to view and maintain a specific message. An implementation may introduce messages used in custom processes or may choose to override the text for messages delivered by the product.

> **⚠ Caution**
>
> Message category 80000 or greater must be used to define new messages introduced for a specific implementation of the system. Changes to other Message Text will be overwritten when you next upgrade. If you want to make a change to a Message, drill down on the message and populate the Override Message Text. Note that even for message categories 80000 and higher, message numbers lower than 1000 are reserved for common base product messages.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a specific message. Once a message has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding System Messages for more information.

The following zones may appear as part of the portal's **Main** tab page

Copyright © 2026, Oracle and/or its affiliates.

- **Message.** This zone provides information about the selected message.

Understanding Unified Search

Depending on your product, the toolbar search box may support a product specific search for business entities in addition to the standard search for menu items. For example, the search may support a query of business entities by their name, address, contact details etc. The unified search feature is a simplified version of your product's main search, that allows the user to lookup up records using free form search text or hint codes. For more complex queries, the user may navigate to the corresponding advanced search portal.

The configuration of a unified search option includes the following items:

- A **search service script** that implements the data query and defines the supported **filters**. The application service associated with this script is used to secure the unified search option.
- **Search filter rules** and **hints** for the supported filters. These rules are used to parse the text entered by the user to determine whether it qualifies as a valid filter value.
- The **advanced search** portal to use for more complex queries.

The unified search box interacts with a unified search option implementation as follows:

- As the user types free form text in the search box, the application uses the **search filter rules** and **hints** to parse the entered text and translate it to corresponding query filters.
- The **search service script** is then called to query for entities that match these filters and present the results.
- Selecting a result record navigates to the corresponding portal.
- Selecting the **Advanced Search** link in the search box navigates to the general query portal associated with the unified search option.

If a unified search option is not enabled for your product or the user does not have security access to it then the search box may only be used for searching the menu.

The following sections describe various aspects related to the configuration of a unified search option.

The Search Script Controls Everything

A unified search is implemented as a single service script that controls the following:

- **Filters.** The script determines which filters are supported and how they are applied to return results efficiently. Each filter name must be a valid **Field** record.
- **Results.** The script returns a list of business entities that matches the query filters provided by the user, along with the navigation information needed to drill in to a selected record in the list.
- **Application security.** Only users that have access to the script's application service may use the unified search associated with the search script.
- **Override Advance Search.** The script may override the advanced search portal configured for the search option.

When the search box component initializes, the script is called once in **Prepare Search Details** mode to return the list of supported filter names. In this mode the script does not perform any query.

Copyright © 2026, Oracle and/or its affiliates.

As the user types in text in the search box, the search box component analyzes the entered text to derive query filters and calls the search script in **Perform Search** mode to return matching results.

The search box component is designed to support any type of unified search implementation and as such, the interaction between the user interface component and the search script is generic. Filters are defined as name / value pairs and results for each matched entity includes details like primary, secondary and tertiary description lines, along with a line that describes the matching text for entry.

> **Note**
>
> If your organization needs to implement a custom unified search script, it is strongly recommended that you follow the base product's implementation of such scripts as a reference.

Filter Rules and Hints

A unified search is designed to support a predefined set of filters for querying data. For example, searching customers by their address, name, identifiers; or searching locations by address or name etc. Each filter is uniquely identified by a **Field** name as declared by the search service script.

As the user types in free text in the search box, the application applies parsing rules to determine whether the text may be used as the value of one or more filters supported by the unified search. For example, if the user types in the text "ave", parsing rules may determine that it can be used to filter results by a person's name or address but not their email address.

Each parsing rule, defined in the **Search Filter Rule** extendable lookup, associates a filter's field name with a regular expression. Multiple filter rules defined for the same filter name are processed in execution order sequence and it is enough for one rule to return a match for the search text to be used as the value for the specific filter.

Alternatively, the user may use an explicit hint code to identify the filter they would like to use followed by the filter value. For example, an address filter may be associated with the hint "ad:" allowing the user to enter "ad: main" to look up records by address.

Each hint, defined in the **Search Filter Hint** extendable lookup, associates a filter's field name with a hint code. When applicable, a hint may be associated with a value expression rule that is used to qualify the entered value as eligible for use as a filter value. For example, requiring the value to be of a minimum length before a search is performed. If the entered value does not match the hint's value expression the corresponding search is not performed.

Note that filter rules and hints are defined for a specific filter name and not a specific unified search option. The same parsing rules may be shared across searches using the same filter names. For example the parsing rule to identify an "email" filter or the "ad:" hint for an address filter are the same regardless of how these are used by a search query. These rules are automatically associated with any unified search option that uses the same filter names.

> **Note**
>
> You may customize the parsing rules and hints associated with a filter by inactivating the base product's extendable lookup records and introducing custom rules that match your business requirements.

Copyright © 2026, Oracle and/or its affiliates.

Which Unified Search Option is Used?

Each unified search option is defined in the **Unified Search** extendable lookup, referencing the option's search service script and its corresponding advanced search portal. The unified search option is only enabled to users that have access to the search script's application service. When more than one unified search option is configured and enabled for a user, the application chooses the option with the highest sequence as the default option, allowing users to switch to other options they have access to.

Customizing Unified Search

You may customize the parsing rules and hints associated with base product search filters by inactivating the base product's extendable lookup records and introducing custom rules that match your business requirements.

There is no need to duplicate a product hint to override its Value Expression rule. You may do so by populating the Override Value Expression element on the hint definition record.

If you need to support a search for different business entities and filters, you need to implement a custom unified search. The configuration of a unified search option involves the following high level steps:

- Design and implement a search script that supports a predefined set of filters and returns the topmost set of matching results while taking into account performance considerations.
- Design hints and filter rules as needed for each of the supported filters. Review existing hints and filter rules associated with same filter names when applicable.
- Register each hint, if any in the **Search Filter Hint** extendable lookup.
- Register each filter rule, if any in the **Search Filter Rule** extendable lookup.
- Register the search script in the **Unified Search** extendable lookup.

Trees

Many edge products and implementations make extensive use of UI tree structures to display hierarchical relationships between records in the system. The framework provides an objects to define a tree structure and the rules on how to populate the nodes in the tree, and processes to interpret the configuration and dynamically render the tree display.

The topics in this section provide information about what is supplied in the framework. Your product may supply UI components with ready built trees for common use cases. Refer to your product documentation for more information.

Common Attributes of Trees

Your organization may have many portals and zones showing data in a hierarchical tree. Most trees in the system follow a similar pattern:

- Trees have a number of levels. There may be different types of tree nodes present in any given level; for instance a list of algorithm types for a script, followed by a list of script steps. Icons may be displayed to further identify the node’s object type.
- The most common node types reference an entity within the system such as a schema, script, data area and so on. These nodes usually display the information string from the entity's FK reference and provide the ability to navigate to the entity. They may also allow the node context to be broadcast to other zones in the portal.

Copyright © 2026, Oracle and/or its affiliates.

- In addition to the ability to navigate to the associated entity, many node types provide a context menu with further navigation paths.
- Some nodes are simply informational and are used to group a set of child nodes below a type or category. For example, the business object Options tree groups the option value nodes below summary nodes that describe the option type.

Each tree defines the nodes that make up the tree and the parent/child relationship between those nodes. A tree node defines which of the common attributes are applicable to the node and the algorithms responsible for retrieving the node data and, optionally, determining the node information to be displayed.

Populate Node Algorithm

A tree node displays a combination of the attributes listed above, typically for an object in context. For a root node, the object will normally be derived from a global or portal context value, such as the current script. For each lower level node, the context is derived from the context fields of its parent row; for instance, algorithms for a specific algorithm type.

The key responsibility of the populate node algorithm is to receive the tree code, node name and the parent context value (or values), determine if there is data that should be shown for this node, and return the a list of the context fields and values applicable to each row in the node. The following highlights additional optional information that may be returned by the algorithm:

- Entity information. If the node is related to a specific entity, the maintenance object code and primary key values may be returned. These are used for determining default information for the node, including navigation details.
- A display icon value. This is used to override the icon defined on the tree node and/or the default icon for the entity as defined on the maintenance object or business object.
- An information string. This may be used to override the default information string for an entity or to display stand-alone text such as a summary or description of the child nodes to follow.
- Override broadcast details if needed for a specific node. For example, set specific broadcast field names and values when they differ from the related entity's primary keys or indicate that broadcasting is not applicable for a node instance even though broadcasting is enabled by the node's definition.

> **Note**
>
> The populate node algorithm is responsible for returning all the context values required to populate its child nodes. For an informational or summary node, this may require including the node’s parent context fields in its own context. For example, in the script tree, the summary node describing the number of steps is followed by a list of steps for the script. The summary node populate node algorithm needs to include the script code passed in as parent context in its output context list, in order for the steps to be correctly retrieved.

Information Override Algorithm

The information override algorithm has two main purposes:

- Providing alternative information text for a tree node when the information provided isn’t suitable for your implementation.

Copyright © 2026, Oracle and/or its affiliates.

- Retrieving contextual insights related to the tree node entity, to be included in the node's information text. Only badge insight types are currently supported.
- Override broadcast details if needed for a specific node. For example, set specific broadcast field names and values when they differ from the related entity's primary keys or indicate that broadcasting is not applicable for a node instance even though broadcasting is enabled by the node's definition.

The override information algorithm receives the tree code, node name, maintenance object and prime key values for the node. The following highlights the information that may be returned by the algorithm:

- A display icon value.
- An information string. This will be used to override the default information string in the node's formatted string.
- Full node text. This will be used as the full text for the display, overriding the formatting defined on the node.
- A list of badge insight types and their context.

> **Note**
>
> If the information override algorithm returns insight types for the entity, it is also responsible for returning all the context values required for those insight types. Refer to Contextual Insights for more information.

Preparing a Node for Display

The following section highlights key steps in fetching a node's data and formatting it for display.

Determining Node Content

The following steps are performed to retrieve the components making up the tree node display:

- The Populate Node algorithm is called to determine all instance nodes for the tree node along with their related details.
- The Information Override algorithm is called for each node instance to allow for optional override of presentation related details.

The node display contents are determined as follows:

- The icon is determined from the following sources, in order: tree node definition, override information algorithm value, populate node algorithm value, maintenance object / business object default value.
- Broadcast usage is determined by the setting on the tree node definition.
  - This setting can be overriden by either the Populate Node algorithm or the Override Information algoirthm (which takes precedence) to indicate that broadcasting is not applicable for a specific node instance.
  - By default the broadcast fields for a node are the primary key names of its related entity. Broadcast field names can be overriden by either the Populate Node algorithm or an Override Information algorithm where the latter takes precedence.
- The information string is determined from the following sources, in order: override information algorithm value, populate node value, maintenance object / business object information algorithm. If the string is derived from the MO/BO algorithm, the short information text is used when available else the standard information is used.

Copyright © 2026, Oracle and/or its affiliates.

- Context menu usage is determined by the settings on the tree node definition.
- The system supports a different style for header rows, including a larger font. Header style usage is determined by the setting on the tree definition and is only applicable if the node is a root node with only one instance.

Node Text Format

The text for nodes referencing a system entity commonly includes some combination of a label, the entity key and an information string. The node configuration provides a label field and a substitution format string field which may be used to define the node text format. The format string defines which of the components should be included in the text, the order of those components and any surrounding text and HTML markups. For example, a format string of '`%label` - `%key` - `%info`' will result in a node text like the example below:



If the format string is blank, the default text will include '`%info`' only. If an override information algorithm is configured that returns the full node text, the text is presented without further formatting.

Auto Expansion

If the tree node definition indicates that the node should be expanded, the system will automatically fetch and display the data for its child node. This processing continues until there are no further child nodes marked for expansion.

There are cases where a tree node should be conditionally expanded based on business rules. For example, prevent the initial expansion when the number of child nodes exceeds a certain threshold. You may design the "Populate Node" algorithm to indicate whether a tree node should be initially expanded or not. When populated, the indication overrides the setting on the tree node definition.

Base Delivered Trees

Trees are system data and your product may already deliver several trees out of the box. You may choose to customize a base delivered tree in the following ways:

- You may add your own nodes to an existing tree definition.
- You may override the information string for a node by supplying your own **Information Override** algorithm. If a base override algorithm is configured on the node, your custom algorithm should be plugged in at a higher sequence number, to take precedence over the base algorithm.

Rendering a Tree

To render a tree on the UI, you would include the ou-tree web component in a UI map that references the tree definition.

> **Note**
>
> Refer to Include a Web Component for more information.

- Defining Trees

Copyright © 2026, Oracle and/or its affiliates.

Defining Trees

This portal is used to view and maintain information about a tree definition. Trees are commonly used to display hierarchical relationships between objects in the system. Refer to Trees for background information about defining trees.

You can access the portal from **Admin**, and then **System**, and then **Tree**. You are brought to a query portal with options for searching for a specific tree definition. Once a tree has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones will appear as part of the portal's **Main** tab page

- **Tree.** This zone provides information about the selected tree.
- **Tree Nodes.** This zone lists the nodes for this tree in hierarchical order. Use the zone actions to add or maintain a node in the tree.
- **Tree Node.** This zone displays the details for a tree node. It is only visible if a tree node name has been broadcast from the Tree Nodes zone.

Contextual Insights

Contextual Insights provide a powerful way to surface focused intelligence and alerts to users in the context of the core screens for any applications built on Oracle Utilities Application Framework. There are multiple options for the visual patterns when surfacing Contextual Insights and the actions that can be performed by users are easily configurable as well. The subsequent topics provide detail about the Insight Type and Insight Group objects that are used to configure the information that may then be displayed in appropriate screens.

- Insight Types
- Insight Groups

Insight Types

The insight type object contains all the detailed configuration for displaying one or more alerts, or important information for a record in context. The following sections provide more information about all the configuration managed by the insight type.

Insight Algorithm

Each insight type is used to display some information, typically related to an object in context. It is the responsibility of the insight type algorithm to receive the context value (or values), determine if there is information that should be shown for this context, return the text to display.

The following highlights additional optional information that may be returned by the algorithm:

- A secondary message. This is only applicable to some visual structures where more than one line of information may be displayed.
- A severity value. This is used to override the visual display configuration if different emphasis should be used for different severity values. Refer to Override Configuration for more information.
- Entity information. If the insight is related to a specific entity, the maintenance object code and primary key values may be returned. This is useful if the insight type configures a BPA or user action algorithms that may perform logic for this entity.

Copyright © 2026, Oracle and/or its affiliates.

- Navigation and User Action information. This is only applicable to visual structures that support user actions. The algorithm must return the list of all the actions that should be made visible on the insight. If the action includes navigation, the algorithm should return the navigation option and navigation context key and value pairs. Additionally, there is configuration for each action that must be defined. Refer to User Actions for more information.

Each algorithm can return multiple insights, if applicable. For example, if the insight is about To Do entries related to the object being displayed, you could design your insight algorithm to return a single insight summarizing all the To Do related information. Or you could choose to return separate insights based on To Do information so that you provide more granular information and customize actions or navigation based on the different conditions. For example, if there are To Do Entries that the current user may perform actions on and others that current user cannot update, the algorithm could return two separate insights: one with actions and one without.

> **Note**
>
> **Preview mode**. The insight type maintenance page includes an Insight Preview section that shows the output from the insight algorithm when invoked with an insight mode of "Preview". Every algorithm should include logic that returns appropriate text when called in this mode so that something is displayed in that section.

Visual Structures

Each insight type defines a visual structure which dictates what kind of information may be shown. The system has some fixed layout expectations for each visual structure type. The insight type configuration is used to provide the source for the possible layout.

> **Note**
>
> All the insights use color options as part of their configuration. Refer to Color Option for more information.

Badge Insight

The simplest visual structure is the badge. It is used to display a single line of quick text with a solid background (for visibility) that is meant to alert the user to important summary information or attributes.

Inline Insight

The inline insight defines an icon and a single message. This is meant to be more descriptive than a badge. For this type of insight, a default light background is used so the recommendation is to select a color option for both the text and the icon that has only a foreground color defined or has a light background color.



List Insight

The list insight are intended to exist in a list of insights for the object. It defines an icon and can support two lines of text. In addition, this type of insight supports buttons or user actions. Refer to User Actions for more information. Note that for a list insight, if there is one valid action or 2

Copyright © 2026, Oracle and/or its affiliates.

valid actions, the action buttons are shown. If there are three or more valid actions, one action is shown and then an 'Other Actions' button is visible. Clicking this shows the other available actions.

Note that the background color for the insight is used as the color for the full insight snippet. The text and the icon are shown on this background. The icon's background color is ignored.

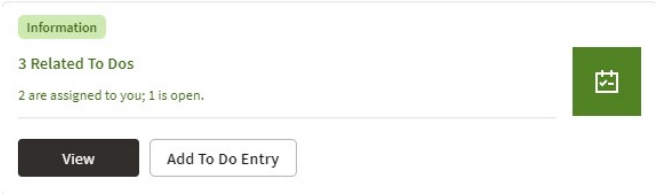


Card

The card insight is the most information rich, with up to two lines of text, user actions, a large icon to call attention and header text. The header text is chosen from a dropdown list of valid options. The product provides out of the box header text of "Information", "Warning" and "Error". Implementations may add additional values to this using the lookup field HEADER_CAT_FLG.

Please note the following about the treatment of the various components of the card insight:

- The header text is rendered like a badge insight.
- For the message text displayed in the card, the background color is ignored so a color option with no background color or a light background color should be chosen.
- For the icon, the background color is used to create a square area for display and the icon is rendered in the foreground color.



Override Configuration

Each insight supports default visual setting for the various components of the insight. This includes the color option for the applicable components along with icons and header text, where applicable. In each case, override visual settings may be defined for a given severity number.

For example imagine the algorithm for a badge insight that returns a message with the count of related To Do entries uses a different color option if there are more than 10 related records. To do this, the algorithm can return a severity number and then the configuration user can define the override color option for this severity number.



User Actions

The list and card insights support the definition of user actions that may be presented as buttons when displaying the insight. The actions may do one or more of the following:

- Perform algorithm logic. One or more user action algorithms may be associated with an action button. This should be used to perform server logic that does not require any user interaction. For example, you can provide an action to update the status of a record related to the insight. These algorithms are configured on the Insight Type.

Copyright © 2026, Oracle and/or its affiliates.

- Navigate to another page. For this type of action, the insight algorithm must provide the navigation context and the navigation option to launch when clicking the button. It's possible to configure navigation and configure user action algorithms. In that case the algorithms are run first and then the navigation occurs.
- Launch a BPA. This is used if there is user interaction needed to perform additional logic and / or if there is conditional navigation such that the logic is needed before navigating. It's possible to configure a BPA and configure user action algorithms. In that case the algorithms are run first and then the BPA is launched. If the insight algorithm also provides navigation for this action, it is ignored in favor of the BPA. The BPA is configured on the Insight Type.

The list of valid actions to display at run time must be returned by the insight algorithm. This allows for the algorithm to have control over which actions to display, in case there are some actions that are conditional. In addition, each possible button requires configuration related to what text should display and the style of the button.

Button Text

The button text to display is chosen from a button text lookup. The lookup name is `INSIGHT_ACT_FLD_FLG`. Implementations may add additional values to this lookup to include new button text. In addition, if you would like to change button text on any product delivered button, you can override the lookup value description.

Action Class

Each action button defines an action class. Any action configured with the action class value of **Positive** value is shown first and is styled differently for emphasis. Other action class values are for information purposes.

Securing Actions

If the action navigates or launches a BPA, the system automatically checks the underlying security for the target page or the BPA and only shows the action if the user has security rights. Other types of actions, such as ones that should launch an algorithm, do not have security configuration that can be automatically detected by the system. For these types of actions, you can define an explicit application service and access mode on the Buttons collection. It's also possible for the algorithm to check security before returning applicable buttons.

Base Delivered Insight Types

Insight types are system data and your product may already deliver several insight types out of the box. You may choose to include one or more insight types in an Insight Group.

Rendering an Insight

To render an insight for an insight type, you would include the ou-insights web component in a UI map that references this insight type. This is how the Insight Preview section of the insight type maintenance page renders the preview.

> **Note**
>
> Refer to Include a Web Component for more information.

- Defining Insight Types

Copyright © 2026, Oracle and/or its affiliates.

Defining Insight Types

The insight type entity is used to define the configuration of a specific type of alert to display extra information related to one or more records in context. Refer to Insight Types for a detailed description of the supported functionality.

To view or maintain insight types, open **Admin**, and then **General**, and then **Insight Type.**

This is a standard All-in-One portal.

The information captured on the insight type depends on the record's business object. Refer to the embedded help text for more information.

Related Insight Groups

If there are any insight groups that reference this insight type, the **Related Insight Groups** zone displays the linked insight groups.

Insight Groups

An insight group may be used to include several insight types together onto a single map. When defining an insight group, you configure the visual structures that the insight group should include. Only insight types for the configured visual structure are available.

Insight Class

Each insight group must reference a unique insight class. The insight class is defined using a lookup. The lookup field is `INSIGHT_CLASS_FLG`. Your product may already deliver base values for insight class. To display insights in a user interface your UI map would include the ou-insights web component. This component includes syntax to reference an insight class and the logic will find the insight group for that class and render all the insight types linked to that group.

Because the insight class is used as the reference in the web component rather than the insight group, it allows your product to supply base logic to show insight group information in a base delivered user interface. The base logic defines an insight class in base and references it on a base UI map. Your implementation can then define the insight group for this insight class and choose the insight types that should be rendered in that map. Refer to your specific product documentation to find out if there are any base delivered insight classes.

If your implementation identifies a new use case where one ore more insight types should be displayed, once the insight types are configured, do the following:

- Define a new Insight Class lookup value.
- Define an Insight Group that references this insight class and configure the appropriate insight types.
- Include the appropriate web component in the map for your user interface where you wish to display this information.

Multiple Insight Groups for an Insight Class

Typically there should exist a single insight group for an insight class. In cases where the list of insight types for the class needs to vary based on context, a separate insight group needs to be defined for each list.

Copyright © 2026, Oracle and/or its affiliates.

When more than one insight group is associated with an insight class **Action Method** rules may be applied to determine which group to use based on the context provided to insight processing. Refer to Action Method for an overview of action method functionality.

- Defining Insight Groups

Defining Insight Groups

The insight group entity allows implementations to define multiple related insight types that should be displayed together in a given user interface map or zone. Refer to Insight Groups for an overview of insight group configuration and functionality.

To view or maintain insight groups, open **Admin**, and then **General**, and then **Insight Group.**

This is a standard All-in-One portal.

The information captured on the insight group depends on the record's business object. Refer to the embedded help text for more information.

Related Insight Types

The **Related Insight Types** zone displays all the insight types that are currently linked to the group. The Edit link in the zone header allows you to add or remove insight types. Only insight types that reference one of the **Valid Visual Structures** may be selected.

Knowledge Articles

When applicable, a knowledge article icon may be configured to appear in a specific location on a page. Clicking the icon launches a drawer presenting an article on the topic and allows the user to further search and review other articles.

> **Note**
>
> The product supports only knowledge articles defined in Oracle Fusion.

The association between a user interface location and a related knowledge article is registered in the **Knowledge Management Articles** extendable lookup. The user interface location references an extendable lookup knowledge code and the article ID is referenced on the extendable lookup record. The knowledge article icon appears on the page only when the corresponding extendable lookup entry is active and references an article ID.

Some product pages are released with embedded references to knowledge codes. By default, these knowledge codes are not associated with any article. To enable these locations for knowledge functionality update their respective extendable lookup entries with references to your articles.

You may also introduce knowledge functionality to custom pages as follows:

- Add a knowledge code entry to the extendable lookup along with a reference to an article ID.
- Reference the knowledge code on your custom page using the Knowledge Article web component.

Access to knowledge articles requires an integration with Oracle Fusion. Once the integration is established, follow these instructions to complete the configuration:

- Update the context details for the **Get Fusion Article** Message Sender.

Copyright © 2026, Oracle and/or its affiliates.

Table Get Fusion Article - Message Sender (F1-KMGET)

Context Type	Context Value
HTTP Method	GET
HTTP Timeout	60
HTTP Transfer Method	SendReceive
HTTP URL 1	@EXT_PUB@<hostname>:<portnumber>/api/knowledge-management/v2/articles/answers/${pathParms}?${queryParms}
OAuth Grant Type	jwt_user_assertion
OAuth Access Token URL OAuth Client Credentials Key Ring Oauth Signature Key Ring OAuth REST API Scope	Refer to OAuth Configuration for more information.

- Update the context details for the **Search Fusion Articles** Message Sender.

Table Search Fusion Articles - Message Sender (F1-KMSCH)

Context Type	Context Value
HTTP Method	GET
HTTP Timeout	60
HTTP Transfer Method	SendReceive
HTTP URL 1	@EXT_PUB@<hostname>:<portnumber>/api/knowledge-management/v2/search?${queryParms}
OAuth Grant Type	jwt_user_assertion
OAuth Access Token URL OAuth Client Credentials Key Ring Oauth Signature Key Ring OAuth REST API Scope	Refer to OAuth Configuration for more information.

Color Configuration

The topics in this section describe some options for configuring colors to be used in displaying information in the user interface.

- Defining Colors
- Color Options

Defining Colors

The system provides the extendable lookup Color (F1-Color) to define valid colors to be used in the system. The base colors provided are ones related to the standard color palette used in Oracle products.

Copyright © 2026, Oracle and/or its affiliates.

At this time, zones and UI maps that define specific colors do not support any syntax to reference a color extendable lookup. In the meantime, colors applied at the zone or map level can still reference HTML colors or reference hex codes directly.

Color Options

The color option entity is used to define a combination of colors to be used in user interface components. The system provides a standard set of color options used for contextual insights.

To view or maintain color options, open **Admin**, and then **System**, and then **Color Option.**

This is a standard All-in-One portal.

The information captured on the color option depends on the record's business object. Refer to the embedded help text for more information.

Defining Display Icons

Icons are used to assist users in identifying different types of objects or instructions. A limited number of control tables allow administrative users to select an icon when they are configuring the system. Select **Admin**, and then **System**, and then **Display Icon** to maintain the population of icons available for selection. You are brought to a query portal with options for searching for a specific display icon. Once a display icon has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The latest standard is to use Scalable Vector Graphic (SVG) icons. For these types of icons, the Icon Type is **Variable (SVG)**. The product also supports icons in the .gif or .png format. For these types of icons, the Icon Type defines a pixel size.

URL describes where the icon is located (folder) and the icon name (file name).

Note that there is a standard folder structure for delivered assets such as images. The structure is `{DefaultWebApp}/{owner}/assets/images|svgs/`

- {Owner} - this is the owner of the icon. The `/cm` folder is used for custom icons.
- svgs - all images that are of type svg are in this folder.
- images - all files that have a file suffix of jpg, gif, png or other types of image formats are in this folder.

When using the above folder structure, then the URL only needs to contain the icon name (file name). The correct full URL is determined automatically at runtime. Images can optionally be placed in other folders under the DefaultWebApp. However, this is not recommended but if used, the URL must contain the full path to the icon (e.g. /cm/assets/newImages/myIcon.gif)

Custom Look and Feel Options

The default look and feel of the application can be customized via feature configuration. The base product is provided with a **Custom Look And Feel** Feature Configuration type. If you define a feature configuration of this type, you can adjust the look and feel in the following ways.

- You may adjust the look and feel of the application using cascading style sheets. Refer to User Interface Style Customization for more information.
- You may adjust configuration related to the display of the UI Map Help icon.

Copyright © 2026, Oracle and/or its affiliates.

- You may choose to change the UI View Settings for all users in the system. To do this, define an entry for the **UI View** option type and enter the desired value. Note that individual users may override this setting on their user preferences.
- Redwood Enablement
- User Interface Style Customization
- UI Map Help

Redwood Enablement

The Redwood User Experience is opt-in functionality. To enable this in your implementation, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type `Custom Look and Feel`. If one does not exist, create a feature configuration of this type. Select the `User Interface Style` option and set the value to 'Oracle Redwood'.

User Interface Style Customization

The base product allows for the conditional inclusion of custom style sheets into the system style set. Custom styles may override any style provided by the base product. The style sheet may also include new styles for use in customer zone definitions. Use the **Style Sheet** option on the **Custom Look And Feel** Feature Configuration to define your custom style sheet. The option supports either referencing a *.css file or referencing a Managed Content record (with the content type **CSS Definitions**.)

> **Note**
>
> The Managed Content option is recommended as this makes the Custom CSS available to be easily transferred to other environments if desired using Content Migration Assistant (CMA).

> **Note**
>
> Some styles cannot change if they are part of the HTML code.

> **Caution**
>
> Implementers must ensure that the customized user interface is stable and scalable. Changing font, alignment padding, border size, and other user interface parameters may cause presentation problems, like scrollbars appearing or disappearing, cursors not working as expected, and unanticipated look and feel alterations of some layouts.

The structure of the CSS extension follows the standard CSS file format regardless of using the File option or the Managed Content option. This means that the "<style>" tag is not needed and the file will contain a list of CSS selectors with their declarations (properties and values).

To find the necessary CSS Selector, navigate the application to an appropriate screen, then "right click" on the element you wish to re-style and select **inspect** from the menu (normally at the bottom). This opens a view of the HTML document. From this view you will be able to see any "id" and "class" attributes that are available on the element. These are generally the most suitable attributes for CSS selectors. In order to make the styling more specific, it is also

Copyright © 2026, Oracle and/or its affiliates.

appropriate to scan up the HTML DOM tree to view parent "id" and "class" attributes. Making the CSS selector as precisely targeted as possible will reduce the possibility of custom stylings affecting elements outside your desired area.

All standard CSS is supported. However, when looking at the most up-to-date CSS standards, be aware that not all browsers support all options without some specific browser settings.

UI Map Help

A tool tip can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map zone or UI Map. Refer to the tips context sensitive zone associated with the UI Map page for more information. The **Custom Look And Feel** Feature Configuration provides options to control the following:

- Whether UI Map Help functionality is turned on or off. By default it is turned on.
- Override the default help image with a custom image
- The location of the help image, either before or after the element.

> **NOT_SUPPORTED**
>
> Refer to the feature configuration for a detailed description of each option.

Database Options

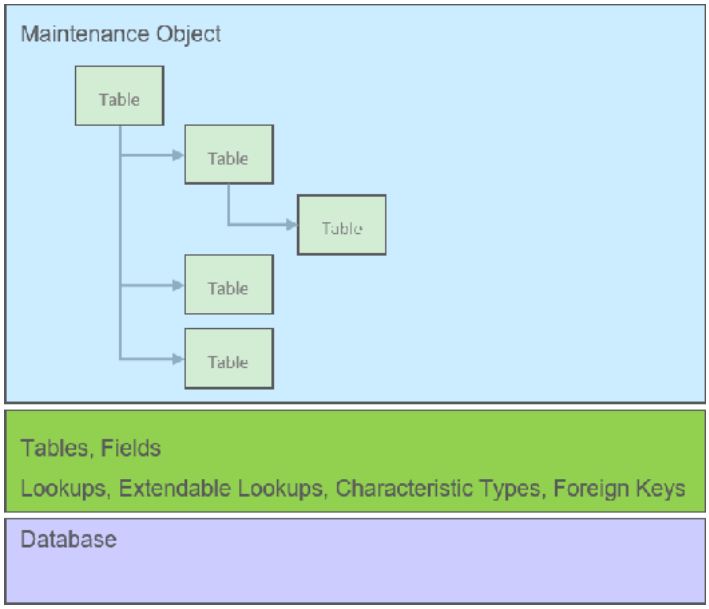
This section describes a variety of database tools that are supplied with the your product.

- Understanding The Data Model
- Defining Tables
- Defining Fields
- Defining Maintenance Objects
- Maintenance Object Configuration
- Defining Lookups
- Defining Extendable Lookups
- Defining Characteristic Types
- Defining Foreign Key References
- Audit Trails
- Bundling
- Revision Control
- Information Lifecycle Management

Understanding The Data Model

The application uses meta-data to describe its data model, from plain description of its tables and fields to more complex hierarchies of maintenance objects.

Copyright © 2026, Oracle and/or its affiliates.



The following points briefly outline key aspects of application's data model as illustrated in this diagram:

- Table and Field records describe all the tables maintained by the application.
- Valid values for certain fields are defined using corresponding Lookup and Extendable Lookup records.
- Characteristic types may be defined to support an extension of a maintenance object via its characteristics table.
- Maintenance Objects define the physical structure of an entity. Business objects further refine maintenance objects to describe business entities. Refer to The Big Picture of Business Objects for more information.
- Foreign Key References define ways for searching and navigating to the maintenance portal of an entity.

The following sections further discusses concepts and guidelines related to the data model.

- Fields
- Tables
- Maintenance Objects
- Defining Valid Values
- Characteristic Types
- Foreign Key References

Fields

A Field definition describes the metadata needed by the application to support the capture and display of a piece of information. Fields may correspond to physical columns on database tables or represent any type of element used for calculation or presentation purposes.

The definition includes the following key details:

- Whether the field refers to a physical column on a database table or used as a work field.
- The type of data the field holds.

Copyright © 2026, Oracle and/or its affiliates.

- The label to use when the field is presented.
- The help text to display when the user clicks on the help icon seeking additional information about this field.
- and more

The following sections further discusses concepts and guidelines related to defining fields.

Physical vs. Work Fields

A Field definition record exists for every physical column on a database table that is used by the application. Refer to Understanding Tables for more information.

A Field that does not represent a physical column on a database table is considered as work field and needs to be marked as such. Work fields are typically used as labels as well as describe internal elements used in business rules.

Data Type Information

The type of data captured by a field is defined using the following attributes:

- **Data Type** indicates the basic type of value the field holds. Valid values are **Character, Character Large Object, Date, DateTime, Number, Time, Varchar2** and **XML Type**.
- **Extended Data Type** is used to further define the type of data for certain data types. Valid values are **Currency Source, Day of Month, Duration, Money, Month of Year, Flag, Switch** and **URI**.
- **Precision** defines the length of the field. In the case of variable length fields, it is the maximum length possible. For number fields that include decimal values, the precision includes the decimal values.
- **Scale** is only applicable for number fields. It indicates the number of decimal places supported by the field.
- **Sign** is only applicable for numbers. It indicates if the data may contain positive or negative numbers.

Data Type Inheritance

A Field can be defined to inherit its data type attributes from the definition of another field. The field to inherit the attributes from is referenced as the **Base Field**. When using a base field reference, the data type attributes are defined once, on the base field, and all the fields inheriting their data type information from it are kept consistent with that definition as it may change over time. Any change of a field’s data type attributes is automatically reflected on all the fields that reference it as their base field.

Data type information may either be specified directly on the field itself or by referencing a base field. The system does not support a way to partially override the base field’s data type information.

It is recommended to leverage data type inheritance whenever possible, especially when the reason for creating a new field is to provide an alternate label for the original field in a certain context. In that situation the new field should reference the original field as its base field and provide the new labels and help text as needed.

Labels and Help Text

For fields that appear on a user interface, the description acts as a label to be used for presentation purposes. As per the product's multi-language support, a field is associated with a label for each of the languages enabled in the system.

Copyright © 2026, Oracle and/or its affiliates.

A field's help text may be used to provide the user with additional information on how the field is used. When populated for a field that is visible on a user interface that supports display of embedded help, a help icon appears for the field, presenting the help text to the user when clicked.

Overriding Labels and Help Text

The label and help text provided for a product owned field may be overridden by your organization using the corresponding override fields.

The field's label and help text can be overridden for a specific table by populating the corresponding override information as part of the table / field definition. However, this override is not used in portal based user interfaces. It is only applicable if the field is displayed on fixed page user interfaces. For fixed pages, if the override label at the table / field level takes precedence over the override at the field level.

Alternate Labels

There are contexts where a separate label may be needed for the "date" and "time" parts of a date/time field. Such an example is when the field is used by the operational analytics product. For example, for a "Completion Date/Time" field you may provide the context label "Creation Date" for the **Date** context and "Creation Time" for the **Time** context. These context labels are optional.

In the same way, date fields, time fields and date/time fields may be associated with a **Date and Time Base** context label, which is a label without a reference to the word "Date", "Time" or "Date / Time". For example, for a "Completion Date/Time" field, the date and time base label may be "Completion". This label that can be used in combination with date or time attributes such as day, month, year, hours, minutes etc.

Overrides labels are available for these additional context labels.

Tables

A Table definition describes the metadata needed by the application to work with and access a database table or view.

The definition includes the following key details:

- The table's description in the application.
- The list of fields defined on the table and any foreign key constraints they establish with other tables.
- The table's prime key and how it is generated.
- Whether the table holds administrative, master or transactional data.
- Whether the table is released with product owned system data rows or not.
- The table's role in the conversion process.
- How should table content be upgraded.
- and more

The following sections further discusses concepts and guidelines related to defining tables.

System Tables

Tables released with rows that are owned by the base product are referred to as system tables. These tables include the **Owner** (OWNER_FLG) field to capture the owner of each row.

Copyright © 2026, Oracle and/or its affiliates.

Refer to System Data for more information about naming conventions and standards around row record ownership.

The **Upgrade** setting on the table definition record controls what happens to the rows in a table when the system is upgraded to a new release. The most common setting allows customer owned rows to be merged with rows owned by the base product. Other settings may keep the content of the table as is, i.e. rows are not touched by the upgrade, or fully refresh the content of the table with base product rows.

Some fields on certain system owned tables are explicitly marked to allow customization. This means implementation are allowed to change these specific fields for a base owned record. Changes to the field value of one of these types of fields by an implementation are maintained when upgrading to a new version of the product.

Designated Table Roles

Some tables are designed to support the following application roles:

- **Key Tables**
  - A key table holds a copy of a table's prime key only and is only applicable to tables with system generated keys.
  - Tables with a system generated keys reference their Key Table as part of their Table definition record.
  - Records in the key table remain in the system even when the related entity is archived, preventing the same key from being allocated to a new record. Retaining the key allows the system to keep the integrity of any references that may still exist to the archived entity.
- **Language Tables**
  - To support multi-language implementations, all labels and descriptions that are presented to a user are kept in a designated language table where a row exists for each enabled language.
  - The product is released with the English content provided in the various language tables. An additional row is added to these tables for each enabled language along with the translated content for that language. Refer to Support For Different Languages for more information.
  - Tables associated with language sensitive fields reference their corresponding Language Table as part of their Table definition record.
- **Characteristics Tables**
  - Many entities in the system support one or more Characteristics tables, which are used to capture additional fields for the object that are not already supported by the entity's provided attributes.
  - A Characteristics table is designed to capture characteristics applicable to its specific entity only. The entity associated with a Characteristics table is referenced on the Table's definition record.
  - Refer to Setting Up Characteristic Types & Values for more information.
- **Entity Log Tables**
  - Many entities in the system support the ability to capture lifecycle and other important business events on a designated log table.
  - A log table captures the event details in the form of an application message and stores any parameters for the message in a child log parameters table. All log and log

Copyright © 2026, Oracle and/or its affiliates.

parameters tables follow a predefined structure allowing them to be consistently and efficiently implemented.

  - Typically log tables do not participate in the conversion process.

Table Constraints

Foreign key relationships between tables are not enforced at the database level but rather at the application level, providing the application more flexibility around data manipulation and referential integrity.

The following types of constraints may be defined on the Table's definition record:

- **Primary Key** represents the field or set of fields that represent the unique identifier of a record stored in a table.
- **Logical Key** represents an alternate unique identifier of a record based on a different set of fields than the Primary key.
- **Foreign Key** represents a field or set of fields that specifies identifying and non-identifying relationships to other tables in the application. A foreign key constraint references the primary key constraint of another table.
- **Conditional Foreign Key** represents rare relationships between tables where a single field (or set of fields) may reference multiple primary key constraints of other tables within the application as a foreign key.

The system leverages a table's foreign key constraints information to performs referential integrity validation when rows in this table are deleted. These validations can be disabled by checking off the **Enable Referential Integrity** field on the Table definition record.

Table Indexes

The indexes delivered by the base product for a given table are visible in the index metadata. In addition, table index information may be defined as part of the cloud services support for generation of custom tables in certain use cases. Refer to the section Generating Custom Tables below for more information.

Primary Key Type

A table's primary key may be of the following types:

- **User-defined** means the user specifies the key when a record is added. This type of key is typically used for configuration and administrative tables.
- **System-generated** means a program generates a random numeric key for the record when it is added.
  - This type of table is typically associated with a key table. Refer to the **Designated Table Roles** section for more information.
  - System generated keys may be fully random or partially inherited from a parent table's key for performance reasons. Refer to the **System Generated Keys** for more information.
- **Sequential** means a sequence number is incremented whenever a record is added to the table. The next number in the sequence determines the key value.
- **Database Native Sequential Key** means a unique number is assigned using a database sequence object.
- **Other** means some other business rule allocates the table's prime-key or the prime key is made of multiple parts.

Copyright © 2026, Oracle and/or its affiliates.

System Generated Keys

In general, randomly generated numeric system keys are used to attempt to evenly distribute records across a full range of possible IDs. Batch programs that use multiple threads will typically divide the threads using ID ranges and evenly distributed keys will help spread out the work across these threads.

If a table using a system generated key is a child of another table, the child table's key may **inherit** a portion of the random number from its parent table's key. Key inheritance for child records further extends the performance benefit. When considering partitioning, the recommendation for DBAs is to range partition data based on the primary key so that different batch threads operate on different partitions which reduces contention for hot blocks. Ideally the number of batch threads will be an exact multiple of the number of partitions. Batch programs that insert child data (for example batch Billing creation) also benefit from this design especially when the child tables are partitioned in the same way. The parent is often the driver of the batch process. If this is multi-threaded, then each thread is processing a set of parent records in a given ID range and all child records are being inserted into the same ID range.

While most entities with system generated keys either use fully random keys or partially inherited keys from a single parent, entities may be designed to implement more specific key generation rules.

When key inheritance is used, the table definition of the child table defines the following information:

- The number of most significant digits from the parent record's primary key to use as the **prefix** for the child record's key value.
- The **parent key field name** to inherit the key from. This information is provided for display and documentation purposes only and is only provided when the inheritance rule implemented by the entity may be documented this way. Typically the child key value is inherited from a single parent key but in some cases the inheritance rule may be more complex. For example, different types of records in the same child table may inherit their key from different parent key fields. In this case, multiple parent field names may be listed to describe the various inheritance options. When the rule is too complex, this information may not be provided on the table definition record.

Conversion Options

The conversion process manages the process by which master and transaction legacy data, loaded into a staging schema, is validated and assigned new system generated keys before it is copied to the production schema. Refer to Understanding The Conversion Process for more information.

The tool classifies tables into the following roles:

- Administrative tables. These only exist in the production schema. Instead of a table, a view to the production table exists in the staging schema. These views are necessary to enforce foreign key references when data is validated in the staging schema
- Convertible tables that retain their prime keys. Data conversion is supported for these tables and their prime key values remain unchanged.
- Convertible tables that require a new system generated key to be assigned. Data conversion is supported for these tables and their prime key is reassigned by the system during conversion.
- Not convertible tables. Assigning new system generated keys for a record and replacing all foreign key references it might have with the new key is a complicated process that may not be supported for all tables.

Copyright © 2026, Oracle and/or its affiliates.

. The **Data Conversion Role** setting on the table definition record controls if / how the table is used by conversion processing.

Date/Time Configuration

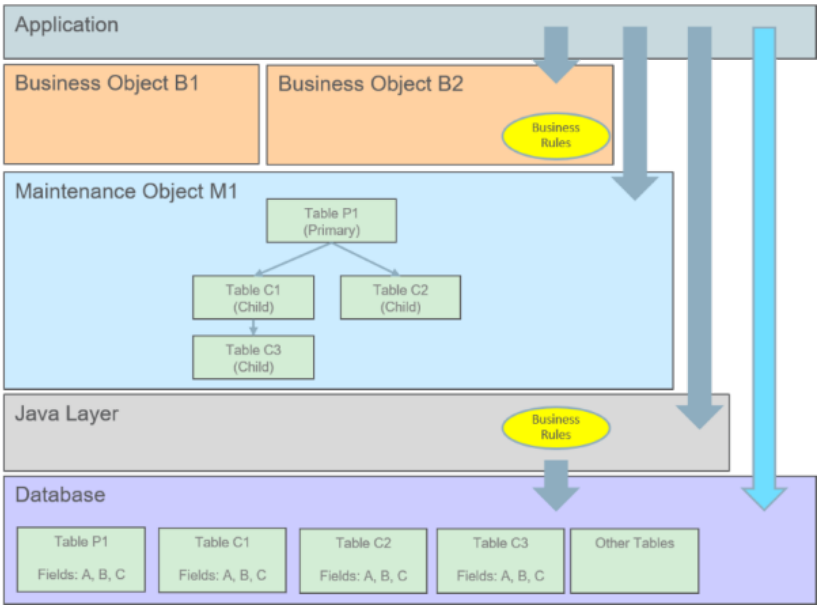
Fields capturing date/time information may be designed to store data in either local legal time or local standard time. Local legal time is captured as entered and displayed as captured in the database. It also assumes the date/time information refers to the time zone defined on the installation options. Refer to Standard vs. Legal for more information.

Generating Custom Tables

For cloud customers and for certain use cases, the product supports defining table metadata for custom tables and generating the related DDLs for that metadata in the database. For these custom tables, there is a Generation Status visible on the Table metadata that identifies where the Table is in the lifecycle for this supported feature. In addition, the index tab on the table portal is used to define custom indexes that are included in the DDL as part of this feature. If you are a cloud customer and would like more information, refer to the Cloud Services Implementation Guide.

Maintenance Objects

A maintenance object defines the configuration of a given “entity” in the system. It includes a primary table and any additional child tables that together capture the physical data for the entity. The application reads, adds, updates and deletes an entity as a single object by invoking its maintenance object service.



The following points briefly outline key aspects of application’s interaction with maintenance objects as illustrated in this diagram:

- Database tables are maintained solely via the application’s Java layer. Direct updates to a table via its Java class is not common and typically reserved to special processes.
- A maintenance object orchestrates the update to an entity’s primary table and its child tables as a single object.
- A business object defines a logical representation of an entity, including all or a subset of the maintenance object’s tables and fields.

Copyright © 2026, Oracle and/or its affiliates.

- The more common way to maintain an entity is via its maintenance object, either directly or via one of its business objects. Most entities support the use of business objects and as such their maintenance is typically made via their business objects.
- Direct access to database tables is supported for query purposes only.

The following sections further discusses concepts and guidelines related to defining tables.

Maintaining An Entire Object

A maintenance object includes a single primary table and one or more child tables where each child table can further have nested child tables. The table structure of a maintenance object represents the scope of data that should be maintained together as a single object:

- When an object is added, data in added to each corresponding table.
- When an object is updated, data in updated in each corresponding table.
- When an object is read, data in retrieved from all the tables.
- When an object is deleted, data in deleted from all the tables.

Algorithms

Several algorithm plug-in spots are also defined on the maintenance object, allowing for business rules that govern all records for this maintenance object.

Options

The maintenance object includes options that define important information related to the maintenance object that may be accessed for logic throughout the system.

Maintenance Objects vs. Business Objects

A maintenance object defines the physical structure of an entity as stored in the database. Different business objects may be defined on top of a maintenance object to represent different types of such entity. A business object includes only the relevant information applicable to the specific type of entity as well as enforces business rules applicable only to that type.

When a maintenance object supports business object configuration, the system uses the **Determine Business Object** maintenance object algorithm to determine the identifying business object that controls a given entity's data structure and rules.

Refer to The Big Picture of Business Objects for more information.

Defining Valid Values

The product provides several options for defining valid values for a column on a table:

- Lookup
- Extendable Lookup
- Control Table

The following provides more information about the functionality of each of the options available for defining valid values for a column.

Lookup

The simplest mechanism for defining valid values for a column on a table is via the Lookup table. This is sometimes referred to as a "simple" lookup to distinguish it from an extendable lookup (described below). Using the lookup table, you can define valid values and their

Copyright © 2026, Oracle and/or its affiliates.

descriptions. When choosing a valid value that is defined by a lookup, a dropdown UI metaphor is used.

The following highlights functionality related to lookups:

- Lookups are associated with a Field. The field is defined as a character data type with an extended data type of **Flag**. The field's label serves as the description for the prompt to select the valid value.
- The lookup code is limited to four characters and must be all uppercase. If there is any functionality where a valid value in the application must match valid values in an external system, the lookup table may not be the appropriate choice.
- The lookup table does not support additional attributes to be defined for each value. This option is only appropriate when a simple code and description pair is needed.
- The product may also use Lookups to define valid values for functionality unrelated to a column on a table. For example, an algorithm plug-in spot may define an input parameter that supports one or more valid values. The plug-in spot may define the valid values using a lookup, allowing for a simple way to validate the value supplied when invoking the algorithm and to document the valid values.

> **NOT_SUPPORTED**
>
> For more information, refer to Defining Lookup Options.

Extendable Lookup

The extendable lookup provides a way of defining valid values for a column with additional capabilities that are not supported using the Lookup table. When choosing a valid value that is defined by an extendable lookup, a dropdown UI metaphor is used.

The following highlights functionality related to extendable lookups:

- Each Extendable Lookups is defined using a business object.
- A field should be defined for the extendable lookup code. The field defines the label for the lookup code and defines the size of the lookup code. The size is determined based on the business use case. In addition, there are standard fields included in all extendable lookups, including a description, detailed description and an override description (so that implementations can override the description of base delivered values).
- The extendable lookup may define additional information for each value if warranted by the business requirement. See Additional Attributes for technical information about additional attributes.

> **NOT_SUPPORTED**
>
> For more information, refer to Defining Extendable Lookups.

Control Table

There may be scenarios where a list of valid values warrants a standalone maintenance object, which is considered an administrative or control table object. When choosing a valid value that is defined by a control, either a dropdown UI metaphor or a search metaphor is used, depending on how it has been designed.

The following points highlight some reasons why this option may be chosen:

Copyright © 2026, Oracle and/or its affiliates.

- The records require a lifecycle such that BO status is warranted.
- The additional attributes are sophisticated enough that they warrant their own column definition rather than relying on using CLOB or flattened characteristic. For example, if a list of information needs to be captured with several attributes in the list and the information in the list needs to be searchable.

In this situation, if a product has provided a control table for this type of functionality, it will be documented fully in the appropriate functional area. If an implementation determines that a custom control table is warranted, all the standard functionality for a maintenance object is required: database tables, maintenance object metadata, appropriate Java maintenance classes, portals, zones, etc. Refer to the Software Development Kit for more information. No further information is provided in this section for this option.

Characteristic Types

Many maintenance objects in the system support characteristics tables, which are used to capture additional fields for the object that are not already supported by the object's provided attributes. Each characteristic is associated with a characteristic type, which defines attributes of the field you wish to capture.

While each characteristic is stored in a separate row in the characteristic table, the user interface metaphor used to capture and display them differ based on the type of page that is used to maintain the object.

- For portal based pages, the business object drives the display and maintenance of the object. The display and maintenance of each characteristic is determined by the business object's user interface design. For these types of pages, it is recommended that characteristics are defined as flattened elements allowing the user interface to display the characteristic as if it is another field. Refer to The Flattening Nodes and Attributes for more information about flattening characteristics on a business object schema.
- There are some fixed pages in the system that do not support customization of the user interface. For these objects, the characteristics are displayed / maintained as a generic list.

The topics in this section describe various aspects of a characteristic type definition and usage.

Characteristic Value Types

Every characteristic referenced on an object references a characteristic type. The characteristic type controls the validity of the information entered by a user when they enter the characteristic's values. For example, if you have a characteristic type on user called "skill", the information you setup on this characteristic type controls the valid values that may be specified by a user when defining another user's skill.

When you setup a characteristic type, you must classify it as one of the following categories:

- **Predefined Value.** When you setup a characteristic of this type, you define the individual valid values that may be entered by a user. A good example of such a characteristic type would be one on User to define one or more predefined skills for that user. The valid values for this characteristic type would be defined in a discreet list.
- **Adhoc Value.** Characteristics of this type do not have their valid values defined in a discreet list because the possible values are infinite. Good examples of such a characteristic type would be ones used to define a user's birth date or their mother's maiden name. Optionally, you can plug-in an algorithm on such a characteristic type to validate the value entered by the user. For example, you can plug-in an algorithm on a characteristic type to ensure the value entered is a date.

Copyright © 2026, Oracle and/or its affiliates.

- **Foreign Key Value.** Characteristics of this type have their valid values defined in another table. For example perhaps you want to link a user to a table where User is not already a foreign key. Valid values for this type of characteristic would be defined on the user table. Before you can create a characteristic of this type, information about the table that contains the valid values must be defined on the foreign key reference table.

  > **Note**
  >
  > When you are searching for a foreign key characteristic value, there are some limitations of when a search is available depending on the search metaphore. Refer to Search Options for more information.

- **File Location.** Characteristics of this type contain a URL. The URL can point to a file or any web site. Characteristics of this type might be useful to hold references to documentation / images associated with a given entity. For example, the image of a letter sent to you by one of your customers could be referenced as a file location characteristic on a customer contact entry. When such a characteristic is defined on an entity, a button can be used to open the URL in a separate browser window.

  File location characteristic values must be entered in a "non-relative" format. For example, if you want to define a characteristic value of *www.msn.com*, enter the characteristic value as `http://www.msn.com`. If you omit the `http://` prefix, the system will suffix the characteristic value to the current URL in your browser and attempt to navigate to this location when the launch button is pressed. This may or may not be the desired result.

  > **Note**
  >
  > Due to browser security restrictions, opening URLs using the file protocol ("file://") from pages retrieved using http does not work. If the file protocol is used, the browser either does not return properly or an error is thrown (e.g., "Access Denied", which usually results from cross site scripting features added for security reasons). This issue has no known workaround. To comply with browser security standards, the recommendation is to move the target files to an FTP or HTTP server location to avoid protocols that are subject to browser security restrictions.
  >
  > Also note that the functionality described in the topics for Referencing URIs do not apply to this value given that the browser is responsible for connecting to the URI and does not go via server logic.

  For references to a file, the recommendation is to use the Attachment functionality to link a file to an object rather than a characteristic type of File Location. Refer to Attachment Overview for more information. The documentation related to file location remains for upgrade purposes.

Searching By Characteristic Values

For certain entities in the system that have characteristics, you may search for a record linked to a given characteristic value. The search may be done in one of the following ways:

- Some base searches provide an option to search for an object by entering Characteristic Type and Characteristic Value.
- Your implementation may define a customized search for an entity by a characteristic value for a specific characteristic type using a query data explorer.

Copyright © 2026, Oracle and/or its affiliates.

- Your implementation may require a business service to find a record via a given characteristic value. For example, maybe an upload of user information attempts to find the user via an Employee ID, defined as a characteristic.

Not all entities that support characteristics support searching by characteristics. To confirm if a given characteristic table supports searching, verify that it includes the field SRCH_CHAR_VAL.

> ⚠ **Caution**
>
> For ad hoc characteristics, only the first 50 bytes are searchable. For foreign key characteristics, the search value is populated by concatenating the values of each foreign key column to a maximum of 50 bytes.

For the base searches that provide a generic option to search by characteristic type and value, you can restrict the characteristic types that can be used to search for an entity. For example, imagine you use a characteristic to define a "jurisdiction" associated with a To Do for reporting purposes. If your company operates within a very small number of jurisdictions, you wouldn't want to allow searching for a To Do by jurisdiction, as a large number of To Do entries would be returned.

A flag on the characteristic type allows an administrator to indicate if searching by this characteristic type is **allowed** or **not allowed**.

Data Typing an Ad hoc Characteristic Value

An **Adhoc Value** characteristic type may further restrict the data type of the captured value by referencing a Field. When populated, the system validates that the captured characteristic value complies with the data type definition of the referenced field.

Valid Characteristic Entities

Characteristics are stored in designated tables, each associated with a unique characteristic entity value referenced on the table definition record. A characteristic type record lists all the valid characteristic entities it is allowed to be captured on.

Customizing Base Owned Characteristic Types

Implementations may customize the following aspects of a base owned characteristic type:

- Extend the list of valid values for a base owned **Predefined Value** characteristic type to include additional custom values. Such extension is only supported if the characteristic type is defined with the **Custom** option turned on.
- Provide a custom algorithm to validate an **Adhoc Value** characteristic type.
- Indicate whether characteristics of this type may be used in searches.
- Extend the list of entities associated with the characteristic type.

Foreign Key References

A Foreign Key Reference defines the necessary information needed to reference an entity in a certain table.

You need to set up this control table if you need to validate a foreign key value against a corresponding table. For example, if a schema element is associated with an FK Reference the system validates the element's value against the corresponding table. Refer to Configuration Tools to learn more about schema-based objects. Another example is characteristics whose

Copyright © 2026, Oracle and/or its affiliates.

valid values are defined in another table (i.e., you use "foreign key reference" characteristic types). Refer to Characteristic Types for a description of characteristics of this type.

A FK Reference is used not just for validation purposes. It also used to display the standard information description of the reference entity as well as provide navigation information to its maintenance transaction. Info descriptions appear throughout the UI, for example, whenever an account is displayed on a page, a description of the account appears. The product provides base product FK references for many of its entities as they are used for validation and display of elements in both fixed page user interfaces as well as portal based user interfaces.

An implementation may also see the need to define a foreign key reference. The following points describe what you should know before you can setup a foreign key reference for a table.

- The physical name of the table. Typically this is the primary table of a maintenance object.
- The program used by default to construct the referenced entity's info description.
- The transaction used to maintain the referenced entity. This is where the user navigates to when using the "go to" button or hyperlink associated with the entity.
- The name of the search page or zone used to look for a valid entity.

The topics in this section describe various aspects of a foreign key reference definition and usage.

Information Description Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the standard information associated with a specific referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the Determine BO maintenance object algorithm system event for more information. If a business object has been determined, the system lets the business object's Information plug-in, if any, format the description.
- If a business object has not been determined or the business object has no such plug-in, the system lets the maintenance object's information plug-in, if any, format the description.
- If the maintenance object has no such plug-in, the system uses the info program specified on the FK Reference to format the information.

> **Note**
>
> **Technical note.** The class that returns the information displayed adjacent to the referenced entity is generated specifically for use as an info routine. Please speak to your support group if you need to generate such a class.

> **Note**
>
> **Generic routine.** The system provides a generic information routine that returns the description of control table objects from its associated language table. By "control table" we mean a table with an associated language table that contains a **DESCR** field. Refer to Defining Table Options for more information on tables and fields. The java class is `com.splwg.base.domain.common.foreignKeyReference.DescriptionRetriever`.

Copyright © 2026, Oracle and/or its affiliates.

Navigation Information Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the actual transaction to navigate to for a given referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the Determine BO maintenance object algorithm system event for more information. If a business object has been determined, use the maintenance portal defined as its **Portal Navigation Option** business object option.
- If a business object has not been determined or the business object defines no such option, the system uses the transaction specified on the FK Reference.

Search Options

The product provides two main metaphors for implementing a user interface. For input fields that are foreign keys, search options are dependent on the metaphor used by the page in question.

- A portal based user interface is a more flexible user interface where an implementation has more options for customizing the look and feel. The base product uses UI maps or automatic UI rendering to display input fields. Elements that are foreign keys may display a search icon if the FK reference defines a Search Zone.

  > **Note**
  >
  > **Defining search zones directly.** It's possible for elements on a UI map to define a specific search zone directly in the HTML, rather than using the search zone defined on an FK reference. Refer to the UI map tips for more information on implementing searches using zones.

- A fixed maintenance page user interface is a page supplied by the base product where only minor enhancements, if any, can be introduced by implementations. The foreign key reference may be used in one of two ways.
  - The based product may use an FK reference to define a base element on one of these pages. If a search is available for such elements, the FK reference's Search Navigation Key is used to implement the search.
  - Entities that support characteristics typically include a generic characteristic collection UI metaphor on these types of pages. In this metaphor, a foreign key characteristic displays a search icon if the FK Reference has configured a Search Navigation Key.

> **Note**
>
> Not every FK reference provided with the product is configured with a search option. This may be the case if the base delivered pages use a dropdown for this foreign key rather than a search. In addition, base provided FK references that do provide a search may not be configured with both search options. It means that if linking this type of object as a characteristic, the search may not be available if the appropriate search is not configured.

Copyright © 2026, Oracle and/or its affiliates.

Defining Tables

This portal is used to view and maintain information about an application table.

You can access the portal from the**Admin Menu**, and then **Database**, and then **Table**. You are brought to a query portal with options for searching for a specific table. Once a table has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Tables for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Table.** This zone provides information about the selected table.
- **Fields.** This zone lists all fields of the selected table and provides various actions to maintain this information.

The following zones may appear as part of the portal's **Relationship** tab page

- **Entity Relationship Diagram.** This zone provides a visual representation of the table's relationship to other tables in the application's data model.

The following zones may appear as part of the portal's **Constraints** tab page

- **Table Constraints.** This zone lists the various key constraints defined for the selected table and provides various actions to maintain this information.
- **Referred By Constraints.** This zone lists all tables referring to the selected table via foreign key constraints.

Refer to Tables for more information.

The following zones may appear as part of the portal's **Indexes** tab page

- **Table Indexes.** This zone lists the indexes defined for the selected table and provides various actions to maintain this information.

Refer to Tables for more information.

Defining Fields

This portal is used to view and maintain information about a field. For product owned fields, you may use this portal to override the label of a field on the various pages in the system.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Field**. You are brought to a query portal with options for searching for a specific field. Once a field has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Fields for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Field.** This zone provides information about the selected field.

Copyright © 2026, Oracle and/or its affiliates.

- **Tables Using Field.** This zone lists tables referencing the field. It is not visible for work fields.
- **Lookup Values.** This zone lists and maintains valid values for the selected lookup field. The zone is only visible when the selected field is a lookup field.

The following zone may appear as part of the portal's **References** tab page

- **Field References.** This zone list various configuration entities referencing the current field.

Defining Maintenance Objects

A maintenance object defines the configuration of a given “entity” in the system. It includes the definition of the tables that together capture the physical data for the entity. In addition, the maintenance object includes options that define important information related to the maintenance object that may be accessed for logic throughout the system. Several algorithm plug-in spots are also defined on the maintenance object, allowing for business rules that govern all records for this maintenance object.

Most maintenance objects are provided with the base package. An implementation can introduce custom maintenance objects when needed. Most fields may not be changed if owned by the base package.

Many maintenance objects in the system support the use of business objects to further define configuration and business rules for a given record. Refer to Maintenance Objects for more information.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Maintenance Object**. You are brought to a query portal with options for searching for a specific maintenance object. Once a maintenance object has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Maintenance Object.** This zone provides information about the selected maintenance object.
- **Tables Information.** This zone displays information about the tables included in the maintenance object.

The following zone may appear as part of the portal's **Relationship** tab page

- **Entity Relationship Diagram.** This zone describes the relatiomship of this entity to other entities in the data model.

The following zone may appear as part of the portal's **Options** tab page

- **Options.** This zone is used to maintain various configuration options associated with the current maintenance object.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You can add new option types.** Your implementation may want to add additional maintenance option types. For example, your implementation may have plug-in driven logic that would benefit from a new type of option. To do that, add your new values to the customizable lookup field `MAINT_OBJ_OPT_FLG`.

The following zone may appear as part of the portal's **Algorithms** tab page

- **Algorithms.** This zone is used to maintain the algorithms associated with the current maintenance object. Refer to the table that follows for a description of all possible system events. If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to Plug-in Scripts for more information.

The following zone may appear as part of the portal's **Business Objects** tab page

- **Business Objects.** This zone displays business objects associated with the current maintenance object.

The following zone may appear as part of the portal's **References** tab page

- **Maintenance Object References.** This zone list various configuration entities referencing the current maintenance object.

Actions

The page supports the following special actions:

- **Create an Algorithm.** Use this action, on the Algorithm tab to create a script based algorithm related to this maintenance object. You are prompted for information regarding the plug-in spot and the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the maintenance object. You may then navigate to the script to adjust it as needed.
- **Inactivate / Activate Base Algorithms.** You can inactivate algorithms on Maintenance Objects. Your implementation may want to use a maintenance object provided by the base product, but may want to inactivate one or more algorithms provided by the base maintenance object. To do that, use the "Inactivate" action on the appropriate algorithm entry on the Algorithm tab. Use use the "Activate" action to mark an inactive algorithm as active.
  Note that these actions add and remove the "Inactive Algorithm" maintenance option as needed.
- Maintenance Object Algorithms

Maintenance Object Algorithms

The following table describes all the system events supported for maintenance objects.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Audit	Optional	Algorithms of this type are called to notify of any changes to the maintenance object's set of tables. These algorithms are invoked just before the commit at the end of a logical transaction. The system keeps track of what records are added or changed in the course of a transaction and all MO audit algorithms are executed in order of when each record was first added or updated.
Determine BO	Optional	Algorithm of this type is used to determine the Business Object associated with an instance of the maintenance object. It is necessary to plug in such an algorithm on a Maintenance Object to enable the business object rules functionality. The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.
ILM Eligibility	Optional	Algorithms of this type are used for maintenance objects that are enabled for object erasure for Information Lifecycle Management. They are used to review records that have reached the maximum retention days and evaluate if they are ready to be archived. The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Information	Optional	We use the term "Maintenance Object Information" to describe the basic information that appears throughout the system to describe an instance of the maintenance object. The data that appears in this information description is constructed using this algorithm. The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.
Manage Erasure Schedule	Optional	Algorithms of this type are used for maintenance objects that are enabled for object erasure, which is a measure to protect Data Privacy. They are triggered when certain system events occur for those objects. The algorithms are used to determine if the record needs to be scheduled for erasure and, if so, create or maintain an entry for the record in the Object Erasure Schedule.
Revision Control	Optional	An algorithm of this type is used to enforce revision control rules when an object is added, changed, or deleted. The maintenance object service calls the plug-in once before the object is processed and once more after applying all business object rules. This allows revision rules to take place in proper revision timings.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Transition	Optional	The system calls algorithms of this type upon each successful state transition of a business object as well as when it is first created. These are typically used to record the transition on the maintenance object's log. Note that most base maintenance objects are already shipped with an automatic logging of state transitions. In this case you may use these algorithms to override the base logging functionality with your own. Refer to State Transitions are Audited for more information.
Transition Error	Optional	The system calls this type of algorithm when a state transition fails and the business object should be saved in its latest successful state. The algorithm is responsible for logging the transition error somewhere, typically on the maintenance object's log. Notice that in this case, the caller does NOT get an error back but rather the call ends successfully and the exception is recorded somewhere, as per the plug-in logic. The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.

Maintenance Object Configuration

The Maintenance Object Configuration portal provides several zones to help implementations manage configuration across several maintenance objects. It is a standard All-in-One portal.

Navigate using**Admin**, and then **Implementation Tools**, and then **Maintenance Object Configuration.**

The following zones appear on the portal.

- **Object Erasure Configuation.** This zone displays a list of maintenance objects that support object erasure. Press the edit icon to open a window in which the configuration values for that maintenance object can be changed.

Copyright © 2026, Oracle and/or its affiliates.

- **Generalized Initial Export Batch Controls.** This zone displays a list of maintenance objects that are eligible for generalize data export, but where appropriate batch controls have not been created. Note that base delivered maintenance objects are delivered with the appropriate batch controls. This zone may not show any data unless you have custom maintenance objects that should be configured for Generalized Data Export.
- **Conversion Batch Controls.** This zone displays a list of maintenance objects whose primary table is eligible for conversion, but where appropriate batch controls have not been created. Note that base delivered maintenance objects are delivered with the appropriate batch controls. This zone may not show any data unless you have custom maintenance objects that should be configured for conversion.

Defining Lookups

Lookup fields may be used to define valid values for a column in a table or for other types of values like parameters to an algorithm. This portal is used to view and maintain information about a lookup field and its values. Note that all lookup information is also maintained as part of the corresponding Field portal.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Lookup**. You are brought to a query portal with options for searching for a specific lookup field. Once a field has been selected you are brought to the maintenance portal to view and maintain the selected record.

NOT_SUPPORTED

Refer to Defining Valid Values for some background information.

Note

When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Lookup.** This zone provides information about the selected lookup field.
- **Lookup Values.** This zone lists and maintains valid values for the selected lookup field.

The base product provides many different lookup fields and their values as part of the product. The following points highlight some functionality related to base-package lookups.

- Fields that are owned by the product will typically provide base lookup values. Implementations are not permitted to remove base delivered lookup values. Implementations may be able to add custom values to base owned lookups. This is controlled with the Custom switch on lookup.
  - When the custom switch is unchecked, it means that there is functionality controlled by the base values and an implementation may not extend or customize this functionality. An example of this type of lookup is the Data Type field on the Field table. The system supports a distinct list of data types and an implementation may not add additional values.
  - When the custom switch is checked, it means that there is base functionality supplied for the base values but that an implementation can extend the functionality by supplying their own values. An example of this type of lookup is the Access Mode on

Copyright © 2026, Oracle and/or its affiliates.

Application Service. The product provides many values for the access mode lookup, representing various actions a user may perform. Implementations may add their own values to this lookup. Documentation should indicate when functionality may be extended and should highlight the lookup value that can be extended.

> ⚠ **Caution**
>
> Important! If you introduce new lookup values, you must prefix the lookup value code with X or Y. If you do not do this, there is a possibility that a future release of the application could introduce a new lookup value with the name you allocated.

- There may be some scenarios where the product supplies a base field and base lookup field with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the lookup values. Typically this type of lookup is for information or categorization purposes. The configuration guide for the functional area associated with the lookup should include a configuration step regarding defining values for this type of lookup.
- The description of base delivered values may be overridden by an implementation.

An implementation may also identify the need for defining a new lookup field with its values.

Defining Extendable Lookups

Extendable lookups are a way of defining valid values that are more sophisticated than simple lookups.

> **NOT_SUPPORTED**
>
> Refer to Defining Valid Values for some background information.

The base product provides extendable lookups as part of the product. The following points highlight some functionality related to base-package extendable lookups.

- The base product may supply base extendable lookup values. Implementations are not permitted to remove base delivered extendable lookup values. It is also possible that implementations may be able to add custom values to base owned lookups. If an implementation is not permitted to add lookup values to the base extendable lookup, the extendable lookup business object will include validation to prevent this. There is no equivalent of the Custom switch that is on the lookup field.
- There may be some scenarios where the product supplies a base extendable lookup with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the extendable lookup values. The configuration guide for the functional area associated with the extendable lookup should include a configuration step regarding defining values for this type of extendable lookup.
- The description of base delivered values may be overridden by an implementation.

Open this page using **Admin**, and then **General**, and then **Extendable Lookup.**

You are brought to the **Extendable Lookup Query** where you need to search for the extendable lookup object (in other words, its business object).

Copyright © 2026, Oracle and/or its affiliates.

Once you have found the appropriate extendable lookup, select the value and you are brought to a standard All-in-One portal that lists the existing lookup values for the extendable lookup. The standard actions for an All-in-One portal are available here.

- Extendable Lookup Advanced Topics

Extendable Lookup Advanced Topics

This section provides some addition technical information about extendable lookup attributes

Defining Additional Attributes

The product provides a few different ways to define additional values for an extendable lookup. Some of the methods are only relevant for base delivered lookup values as they may impact whether or not an implementation can update the values.

The following table highlights the options available and some summary information about what the option provides.

Option	Brief Description	Extendable Lookup Value Searchable by this Attribute?	Base Delivered Value Modifiable?
Element mapped to BO_DATA_AREA	The element is mapped to a CLOB field that allows for base delivered values to be modified.	No	Yes
Element mapped to BASE_BO_DATA_AREA	The element is mapped to a CLOB field that does not allow for base delivered values to be modified.	No	No
Flattened characteristic	The element is defined using the flattened characteristic mechanism.	Yes	No

The following points highlight information from the table above:

- The decision of defining an additional attribute using a CLOB mapping or a flattened characteristic will depend on whether the functionality expects that the lookup value is known when the attribute is needed (in which case a CLOB mapping is appropriate) or if the functionality expects to determine the lookup value based on the attribute (in which case, a flattened characteristic is appropriate).
- When the base product defines an extendable lookup with additional attributes and intends to provide base extendable lookup values, it needs to determine whether or not implementations may update the additional attribute or not.
  - If no and the value is mapped to a CLOB, it will map the value to the BASE_BO_DATA_AREA column. This means that implementations will receive an owner mismatch error when attempting to change the value. In addition, upgrading to a new release will replace the value with the base value.
  - If yes and the value is mapped to a CLOB, it will map the value to the BO_DATA_AREA column. This means that implementations will be able to change the value for a base owned record. In addition, upgrading to a new release will not make any changes to the value.

Copyright © 2026, Oracle and/or its affiliates.

- For values mapped to a characteristic, the product does not support an implementation changing the value of a base delivered record. If the product would like to support an implementation overriding this type of value, the business object will need to be designed with a corresponding "override" element (also a flattened characteristic), similar to how the product supplies an Override Description field to support an implementation overriding the base product delivered description for a base value. This element will not be delivered with any value and will allow an implementation to populate that value.

  > **Note**
  >
  > Note that in this situation, the product functionality that uses this value must cater for the override value.

- All of this detail is only relevant for base provided extendable lookup values. If an implementation adds custom values for a base supplied extendable lookup, all the additional attributes may be populated as appropriate.
- If an implementation defines a custom extendable lookup business object and wants to define an additional attribute using a CLOB, it doesn't matter which CLOB column is used. Both BO_DATA_AREA and BASE_BO_DATA_AREA provide the same functionality for custom business objects.

Capturing a Password

If an extendable lookup includes configuration of a password for some functionality, the system supports automatic encryption of the password value if the schema maps the password to a characteristic using the characteristic type `F1-PWD`.

Defining Characteristic Types

This portal is used to view and maintain characteristic types. An implementation may define their own characteristic types or override some of the settings for base product provided characteristic types.

You can access the portal from the **Admin Menu**, and then **General**, and then **Characteristic Type**. You are brought to a query portal with options for searching for a specific record. Once a characteristic type record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Characteristic Types for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Characteristic Type.** This zone provides information about the selected characteristic type.
- **Characteristic Values.** This zone is visible for pre-defined characteristic types and shows the existing values (if applicable) with action buttons. It also allows you to add new values.

Copyright © 2026, Oracle and/or its affiliates.

Note that you may only add values to a product owned characteristic type if the Custom switch is checked.

Defining Foreign Key References

The topics in this section describe the transaction that allows you to define foreign key references.

Refer to Foreign Key References for more information.

You can access the portal from the **Admin Menu**, and then **Database**, and then **FK Reference**. You are brought to a query portal with options for searching for a specific foreign key reference. Once a foreign key reference has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new foreign key reference, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Foreign Key Reference.** This zone provides information about the selected foreign key reference.
  - Navigation to the referenced entity may be dynamic. The page associated with the specified Navigation Option is used when a dynamic option is not available. Refer to Navigation Information Is Dynamically Derived for more information.
  - The information description of the referenced entity may be composed dynamically depending on the specified Info Program Name. Refer to Information Description Is Dynamically Derived for more information.
  - Certain restrictions apply when displaying the context menu for a referenced entity.

    > ⓘ **Note**
    >
    > Context Menu Name is not applicable to user interface elements displaying a generic collection using a foreign key characteristic type. It is only applicable for pages utilizing the foreign key compound element type for fixed page user interface and for data displayed in a portal based user interface where the foreign key reference is defined as an attribute for an element. Report parameters that reference foreign key characteristics are an example of a user interface where a context menu is not displayed even if the foreign key reference defines one.

  - Certain restrictions apply to search features on fixed page user interfaces.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Search Type and Search Tooltip attributes are only applicable to user interface elements utilizing the foreign key compound element type on fixed page user interfaces. Report parameters that reference foreign key characteristics are an example of a user interface where this information is not used even if the foreign key reference defines them.

The following zones may appear as part of the portal's **References** tab page.

- **Foreign Key References.** This zone list various configuration entities referencing the current foreign key reference.

Actions

The page supports the following special actions:

- **View the source.** If the foreign key information program is shipped with the base package, you can use the adjacent button to display the source code of this program in the Javadocs viewer.

Audit Trails

The topics in this section describe one way of auditing changes in the system. Note that this technique has limitations and may not be the best option for all situations.

- This functionality is configured at the table and field level. Many base maintenance objects use an XML or CLOB field to capture one or more elements using XML format, configured using a BO schema. This auditing technique is not able to capture changes to individual elements. It can only capture overall changes to the single field. The BO Audit plug-in is a better option for auditing changes to individual elements with the CLOB / XML field.
- The base table provided for supporting audits limits the field size of the before and after values to 254 bytes.
- Auditing is captured for each field. In some cases it is prefered to capture a before and after image for several fields at once. For example, if an address is changed, it's more user friendly to capture the before and after for the full address rather than the individual address components. The BO Audit plug-in allows for capturing a single audit record for multiple elements rather than granular changes for each element.

The subsequent topics highlight how to enable auditing for fields, and describe the auditing queries that you can use to view audit records.

- Captured Information
- How Auditing Works
- The Audit Trail File
- How To Enable Auditing
- Audit Queries

Captured Information

When auditing is enabled for a field, the following information is recorded when the field is changed, added and/or deleted (depending on the actions that you are auditing for that field):

- User ID

Copyright © 2026, Oracle and/or its affiliates.

- Date and time
- Table name
- Row's prime key value
- Field name
- Before image (blank when a row is added)
- After image (blank when a row is deleted)
- Row action (add, change, delete)

How Auditing Works

You enable auditing on a table in the table's meta-data by specifying the name of the table in which to insert the audit information (the audit table) and the name of the program responsible for inserting the data (the audit trail insert program). Then you define the fields you want to audit by turning on each field's audit switch in the table's field meta-data. You can audit fields for delete, insert and update actions.

Once auditing is enabled for fields in a table, the respective row maintenance program for the table assembles the list of changed fields and calls the audit trail insert program. If any of the changed fields are marked for audit, the audit program inserts audit rows into the audit table.

> **Note**
>
> **Customizing Audit Information.** You may want to maintain audit information other than what is described in Captured Information or you may want to maintain it in a different format. For example, you may want to maintain audit information for an entire row instead of a field. If so, your implementation team can use the base audit program and base audit tables as examples when creating your own audit trail insert program and audit table structures.

The Audit Trail File

Audit log records are inserted in the audit tables you define. The base product contains a single such table (called `CI_AUDIT`). However, the base audit insert program is designed to allow you to use multiple audit tables.

If you want to segregate audit information into multiple tables, you must create these tables. Use the following guidelines when creating new audit tables (that use the base delivered audit insert program):

- The new audit tables must look identical to the base table (`CI_AUDIT`).
- The new tables must be prefixed with **CM** (e.g., `CM_AUDIT_A`, `CM_AUDIT_B`, etc.). An appropriate java table name must be defined and an *_impl.java program for that java table name must be defined.
- The name of the new table must be referenced on the various tables whose changes should be logged in the new table.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> It's important to note if you use your own tables (as opposed to using the base package table called `CI_AUDIT`), the SQL used to insert and access audit trail records in the base delivered audit program is dynamic. Otherwise, if the base package's table is used, the SQL is static.

How To Enable Auditing

Enabling audits is a two-step process:

- First, you must turn on auditing for a table by specifying an audit table and an audit trail insert program.
- Second, you must specify the fields and actions to be audited for the table.

The following topics describe this process.

- Turn On Auditing For a Table
- Specify The Fields and Actions To Be Audited

Turn On Auditing For a Table

In order to tell the system which fields to audit, you must know the name of the table on which the field is located. You must specify the audit table and the audit trail insert program for a table in the table's meta-data.

> **Note**
>
> Most of the system's table names are fairly intuitive. For example, the user table is called SC_USER, the navigation option table is called CI_NAV_OPT, etc. If you cannot find the table using the search facility on the Table portal, please contact customer support.

To enable auditing for a table:

- Navigate to the Table portal and find the table associated with the field(s) for which you want to capture audit information.
- Specify the name of the **Audit Table**.

> **Note**
>
> **Specifying the Audit Table.** You can use the audit table that comes supplied with the base package (`CI_AUDIT`) to audit multiple tables and fields. All the audit logs are combined in a single table (`CI_AUDIT`). However, you can also have a separate audit table for each audited table. Refer to The Audit Trail File for more information.

- Specify the name of the **Audit Program** . The product provides two programs. `com.splwg.base.domain.common.audit.DefaultTableAuditor` is a standard table audit program. In addition

Copyright © 2026, Oracle and/or its affiliates.

`com.splwg.base.domain.common.audit.ModifiedTableAuditor`is supplied as an alternative. This program does not audit inserts or deletes of empty string data.

> **⚠ Caution**
>
> By default, none of a table's fields are marked for audit. Even though you have enabled auditing for a table, you must still specify the fields and actions on those fields to be audited (see below).

Specify The Fields and Actions To Be Audited

The system only audits actions (insert, update and delete) made to fields that you want audited.

To specify the fields and actions to be audited:

- Navigate to the Table portal for a table on which you have enabled auditing.
- For each field you want to audit, specify the actions you want to audit by turning on the **Audit Delete**, **Audit Insert** and **Audit Update** switches as appropriate.

> **ⓘ Note**
>
> You can also turn on the audit switches using the Field zone at the bottom of the Table portal.

> **⚠ Caution**
>
> Audit Program Caching! The audit program from the table meta-data is read into a program cache on the application server whenever the date changes or when the server starts. If you implement new auditing on a table, your audit trail does not become effective until this program cache is reloaded. In other words, new audits on tables where the audit program was not previously specified do not become effective until the next day (or the next restart of the application server). However, if you change the fields to be audited for a table where the audit program is already in the cache, your changes are effective immediately.

Audit Queries

There are two queries that can be used to access the audit information.

- Audit Query by User
- Audit Query by Table / Field / Key

Audit Query by User

This transaction is used to view changes made by a user that are stored on a given Audit Trail File.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> The system only audits changes that you've told it to audit. Refer to The Big Picture Of Audit Trails for more information.

Navigate to this page by selecting **Admin**, and then **Database**, and then **Audit Query By User**.

Description of Page

To use this transaction:

- Enter the **User ID** of the user whose changes you wish to view.
- Enter the name of the table on which the audit trail information is stored in **Audit Table**. Refer to The Audit Trail File for more information about this field.

> **ⓘ Note**
>
> **Default Note**. If only one audit table is used to store audit trail information, that table is defaulted.

- Specify a date and time range in **Created between** to restrict the records that result from the query.

> **ⓘ Note**
>
> **Default Note**. The current date is defaulted.

- Click the search button to display all changes recorded on a specific audit table associated with a given user.

Information on this query is initially displayed in reverse chronological order.

The following information is displayed in the grid:

- **Row Creation Date** is the date and time that the change was made.
- **Audited Table Name** contains the name of the table whose contents were changed.
- **Primary Key** is the prime key of the row in the **Audited Table** whose contents where changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.
- **Field Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Field Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

Audit Query by Table / Field / Key

This transaction is used to view audited changes made to a given table.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> The system only audits changes that you've told it to audit. Refer to The Big Picture Of Audit Trails for more information.

This transaction can be used in several different ways:

- You can view all audited changes to a table. To do this, enter the **Audited Table Name** and leave the other input fields blank.
- You can view all audited changes to a given row in a table (e.g., all changes made to a given user). To do this, enter the **Audited Table Name** and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).
- You can view all audited changes to a given field in a table (e.g., all changes made to all customers' rates). To do this, enter the **Audited Table Name** and the **Audited Field Name**.
- You can view all audited changes to a given field on a specific row. To do this, enter the **Audited Table Name**, the **Audited Field Name**, and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).

Navigate to this page by selecting **Admin**, and then **Database**, and then **Audit Query By Table/Field/Key**.

Description of Page

To use this transaction:

- Enter the name of the table whose changes you wish to view in **Audited Table Name**.
- If you wish to restrict the audit trail to changes made to a specific field, enter the **Audited Field Name**.
- If you wish to restrict the audit trail to changes made to a given row, enter the row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**). These fields are dynamic based on the **Audited Table Name**.
- Specify a date and time range in **Created between** to restrict the records that result from the query.

> **ⓘ Note**
>
> The current date is defaulted.

- Click the search button to display all changes made to this data.

Information on this query is initially displayed in reverse chronological order by field.

The following information is displayed in the grid:

- **Create Date/Time** is the date / time that the change was made.
- **User Name** is the name of the person who changed the information.
- **Primary Key** is the prime key of the row in the **Audited Table** whose contents where changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.

Copyright © 2026, Oracle and/or its affiliates.

- **Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

Bundling

The topics in this section describe the bundling features in the application.

- About Bundling
- Configuring Maintenance Objects for Bundling
- Working with Bundles

About Bundling

Bundling is the process of grouping entities for export or import from one environment to another.

For example, you might export a set of business objects and service scripts from a development environment and import them into a QA environment for testing. The group of entities is referred to as a bundle. You create export bundles in the source environment; you create import bundles in the target environment.

Working with bundles involves the following tasks:

- Configuring entities for bundling if they are not preconfigured
- Creating an export bundle, which contains a list of entities to be exported from the source environment
- Creating an import bundle to import those entities to the target environment
- Applying the import bundle, which adds or updates the bundled entities to the target environment

- Sequencing of Objects in a Bundle
- Recursive Key References
- Owner Flags on Bundled Entities

Sequencing of Objects in a Bundle

Bundle entities are added or updated to the target environment in the sequence defined in the bundle

Typically, the sequence of entities does not matter. However, sequence is important in the following situations:

- Entities that are referenced as foreign keys should be at the top of the sequence, before the entities that reference them. Specify zones last, as they typically contain numerous foreign key references.
- When importing a business object, specify the business object first, then its plug-in scripts, then the algorithms that reference the scripts, and then the algorithm types that reference the algorithms.
- When importing a portal and its zones, specify the portal first and then its zones.

Copyright © 2026, Oracle and/or its affiliates.

- When importing a multi-query zone, specify the referenced zones first and then the multi-query zone.
- Always specify algorithms types before algorithms.

You can specify the sequence when you define the export bundle or when you import the bundle to the target environment.

Recursive Key References

Recursive foreign keys result when one object has a foreign key reference to another object that in turn has a foreign key reference to the first object.

For example, a zone has foreign keys to its portals, which have foreign keys to their zones. If the objects you want to bundle have recursive relationships, you must create a 'bundling add' business object that has only the minimal number of elements needed to add the entity. A bundling add business object for a zone contains only the zone code and description, with no references to its portals. In the same way, a bundling add business object for a portal defines only its code and description.

When you apply the bundle, the system initially adds the maintenance object based on the elements defined in the bundling add business object. Before committing the bundle, the system updates the maintenance object with the complete set of elements based on its physical business object.

Note that use of the bundling add BO also benefits records that have optional foreign keys or foreign keys that are part of a child table. That way the person creating the bundle does not have to worry about the sequence of the records. For example, an FK Reference may optionally reference a zone for searching. If a new FK Reference and its search zone are bundled together to copy to another region, the bundling add BO for FK Reference (which doesn’t include the Zone) ensures that the FK reference could be added before the zone without getting any validation errors.

Owner Flags on Bundled Entities

The owner flag of the entities in an import bundle must match the owner flag of the target environment.

If you need to import objects that your source environment does not own, you must replace the owner flag in the import bundle with the owner flag of the target environment.

Configuring Maintenance Objects for Bundling

Most base package meta-data objects are pre-configured to support bundling. All other objects must be manually configured.

If a base package maintenance object is pre-configured for bundling, the **Eligible For Bundling** option will be set to "Y" on the Options tab for the maintenance object.

To configure other objects for bundling, review the configuration tasks below and complete all those that apply:

Configuration Task	Scope of Task
Make maintenance objects eligible for bundling	All objects to be included in the bundle.
Add a foreign key reference	All objects to be included in the bundle.

Copyright © 2026, Oracle and/or its affiliates.

Configuration Task	Scope of Task
Create a physical business object	All objects to be included in the bundle.
Create a bundling add business object	Only needed if there are objects with recursive or optional foreign key references.
Add the Current Bundle zone	Perform this step for any object where you want the Current Bundle zone to appear on the sidebar when on the maintenance page for this maintenance object. This is not required by the bundling process.
Create a custom Entity Search zone and add it to the Bundle Export portal	Perform this step if you would like the object to be listed in the Entity Search zone in the Bundle Export portal. This is not required by the bundling process.

Making Maintenance Objects Eligible for Bundling

The "Eligible For Bundling" maintenance object option must be set to "Y" for all bundled objects. Go to the Maintenance Object page and search for the maintenance object. On the Options tab, add a new option with the type **Eligible For Bundling**. Set the value to "Y" and click **Save**. This configuration must be in both the source environment and the target environment.

Adding a Foreign Key Reference

Foreign key references are used for many purposes and most maintenance objects should have them anyway. Check the Maintenance Object options to see if there is an entry for the **Foreign Key Reference** option. If one is not configured, navigate to FK Reference and set up a foreign key reference for the primary table of the maintenance object. Configure that new FK reference on the maintenance object using an option.

Creating a Physical Business Object

If the maintenance object does not already have a Physical BO configured, one must be created to support using it in a bundle. Refer to Creating a Physical Business Object for more information.

Creating a Bundling Add Business Object

If the objects to be bundled have recursive foreign key references, optional foreign key references or child tables that include foreign key references, create a bundling add business object to avoid problems with referential integrity.

1. Navigate to Business Object and add a business object for the maintenance object. Do not fill in any of the UI oriented configuration.
2. The business object is created with a schema that looks like the physical structure of the maintenance object.
3. Remove all child tables and all elements that are not required.
4. Save the business object.
5. Navigate to Maintenance Object and search for the maintenance object you want to bundle.
6. On the **Option** tab, add a new option with the type **Bundling Add BO**. The value is the name of the bundling add business object you just created.

Copyright © 2026, Oracle and/or its affiliates.

Adding the Current Bundle Zone

If you want the Current Bundle zone to appear on the sidebar when the a user is on the maintenance page for the maintenance object, you must add the Current Bundle zone as a context-sensitive zone for the maintenance page's navigation key.

1. Navigate to Context Sensitive Zone and search for the navigation key for the maintenance page of this object.
2. Add the Current Bundle zone F1-BNDLCTXT, to that navigation key.

Adding a Customized Entity Search Query Zone to the Bundle Export Portal

If you want the maintenance object to be searchable in the Bundle Export portal in the Entity Search zone, you must first create an entity-specific query zone to search for the maintenance object. Then you must create a customized entity search zone that references this new query zone. Finally, you must add the customized entity search zone to the Bundle Export portal.

Note that not all base product entities that are eligible for bundling have a search configured for this zone. There is a limit to the number of searches that may be listed here. The base product has provided searches for only the most common objects that are bundled.

Create an entity-specific query zone to search for the maintenance object:

1. Navigate to Zone and search for one of the base query zones, such as the Algorithm Search zone F1-BNALGS.
2. Click the **Duplicate** button in the record actions section.
3. Enter a name and description for the new zone.
4. Click **Save**.
5. Customize the details of the parameters for your maintenance object.

Create a customized entity search zone. This step only needs to be done once. If you already have a customized search zone in the Bundle Export portal, skip this step.

1. Navigate to Zone and search for the F1-BNDLENTQ Entity Search zone.
2. Duplicate this zone.
3. Remove any references to base query zones.

Add the new entity-specific query zone to the customized entity search zone:

1. Locate the customized entity search zone for your Bundle Export portal. This is the zone created in the previous step.
2. Add the name of the query zone you created in the first step to the list of Query Zones.

Add the customized entity search zone to the Bundle Export portal. This step needs to be done only once.

1. Navigate to Portal and search for the Bundle Export portal, F1BNDLEM.
2. In the zone list, add the entity search zone you created above. (Add the new zone after the base entity search zone).
3. Save the portal.

Copyright © 2026, Oracle and/or its affiliates.

Working with Bundles

Use the Bundle Export portal to create an export bundle. The export bundle contains a list of entities to be exported from the source environment. When you are ready to import the objects, use the Bundle Import portal to import the objects to the target environment.

- Creating Export Bundles
- Creating and Applying Import Bundles
- Editing Export Bundles
- Editing Import Bundles

Creating Export Bundles

An export bundle contains a list of entities that can be imported into another environment.

To create an export bundle:

1. Log on to the source environment from which objects will be exported.
2. Select **Admin**, and then **Implementation Tools**, and then **Bundle Export**, and then **Add**.
3. Complete the fields in the Main section to define the bundle's basic properties.

> **Note**
>
> You can use the Entities section to add bundle entities now, or save the bundle and then add entities as described in step 5.

4. Click **Save** to exit the Edit dialog. The export bundle status is set to Pending.
5. While an export bundle is in Pending state, use any of the following methods to add entities to the bundle:
   a. Use the **Entity Search** zone on the Bundle Export portal to search for entities and add them to the bundle. If an entity is already in the bundle, you can remove it.
   b. To import entities from a .CSV file, click **Edit** on the Bundle Export portal, and then click **CSV File to Upload.** Specify the file name and location of the .CSV file containing the list of entities. Click **Submit** to upload the file, and then click **Save** to save the changes.
   c. Use the **Current Bundle** zone in the sidebar of the entity you want to add. (All entities that are configured to support bundling display a Current Bundle zone). This zone displays a list of all pending export bundles to which you can add the entity.
   d. When you check an entity into revision control, specify the export bundle on the **Revision Info** dialog.
6. When you have added all entities, click **Bundle** in the Bundle Actions zone on the Bundle Export portal. The export bundle state is set to Bundled and the Bundle Details zone displays the XML representation of every entity in the bundle.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The owner flags of the entities in the bundle must match the owner flag of the bundle itself. If the owner flags do not match, the system displays a warning message. Click **OK** to continue or **Cancel** to abort the bundle. If you click OK, you will need to resolve the owner flag discrepancy before you import the bundle to the target environment.

7. Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file). You can now create an import bundle and apply it to the target environment.

> **Note**
>
> If you need to make additional changes to the bundle, you must change the bundle state by selecting the **Back to Pending** button in the **Bundle Actions** zone.

Creating and Applying Import Bundles

Import bundles define a group of entities to be added or updated in the target environment.

Before you create an import bundle, you must have already created an export bundle, added entities, and set the bundle's state to Bundled.

To create an import bundle and apply it to the target environment:

1. If you have not already copied the XML from the export bundle, do so now:
   a. Select **Admin**, and then **Implementation Tools**, and then **Bundle Export** and search for the bundle.
   b. Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file).
2. Log on to the target environment.
3. Select **Admin**, and then **Implementation Tools**, and then **Bundle Import**, and then **Add**.
4. In the **Bundle Actions** zone, click **Edit XML**.
5. Paste the contents of the clipboard (or text file if you created one) into the **Bundle Detail** zone.
6. Make any necessary changes to the XML and click **Save**. The status of the import bundle is set to Pending.

> **Note**
>
> Use caution when editing the XML to avoid validation errors.

7. To remove entities from the import bundle or change their sequence, click **Edit**. Enter your changes and click **Save** to exit the Edit dialog.
8. When you are ready to apply the bundle, click **Apply**. The import bundle state is set to Applied and the entities are added or updated in the target environment.

Copyright © 2026, Oracle and/or its affiliates.

Editing Export Bundles

You can add or remove entities from an export bundle when it is in Pending state. You can also change the sequence of entities.

To edit to an export bundle that has already been bundled, you must change the bundle state by selecting the **Back to Pending** button on the Bundle Export portal.

To edit a pending export bundle:

1. Open the bundle in edit mode.
2. Click **Edit** on the Export Bundle portal.
3. Make any necessary changes on the edit dialog and then click **Save**.

Editing Import Bundles

You can remove entities from an import when it is in Pending state. You can also change the sequence of entities and edit the generated XML.

To edit a pending import bundle:

1. Open the bundle in edit mode.
2. To edit the XML snapshot, click **Edit XML**. Edit the XML code as needed, then click **Save**.

> **Note**
>
> Use caution when editing the XML to avoid validation errors.

3. To remove entities or change their sequence, click **Edit**. Make any necessary changes and click **Save**.

Revision Control

The topics in this section describe the revision control features in the application.

- About Revision Control
- Working with the Revision Control Zones
- Revision Control Portal

About Revision Control

Revision control is a tool provided for the development phase of a project to allow a user to check out an object that is being worked on. In addition, it captures the version of the maintenance object when users check in an update, maintaining a history of the changes to the object.

If revision control is enabled for an object you must check out the object to change it. While the object is checked out no one else can work on it. You can revert all changes made since checking out an object, reinstate an older version of an object, recover a deleted object, and force a check in of an object if someone else has it checked out. You may also compare two versions of the object to more easily review changes.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Revision control does not keep your work separate from the environment. Because the metadata for maintenance objects is in the central database, any changes you make to an object while it is checked out will be visible to others and may impact their work.

Many of the maintenance objects used as configuration tools are already configured for revision control, but it is turned off by default. For example, business objects, algorithms, data areas, UI maps, and scripts are pre-configured for revision control.

- Turning On Revision Control
- Configuring Maintenance Objects for Revision Control

Turning On Revision Control

Revision control is turned off by default for maintenance objects that are configured for revision control.

To turn on revision control:

1. Add the base package **Checked Out** zone to the Sidebar portal.
   a. Navigate to Portal.
   b. Search for the sidebar portal F1SDBTOOL.
   c. In the zone list for the portal, add the zone F1-USRCHKOUT.
2. Set up application security.

   For users to have access to revision control, they must belong to a user group that has access to the application service F1-OBJREVBOAS.
3. Add the revision control algorithm to the maintenance object that you want to have revision control.

   This step must be repeated for each maintenance object that you want to have revision control.
   a. Go to the Maintenance Object page and search for the maintenance object that you want to have revision control.
   b. On the **Algorithms** tab of the maintenance object, add the revision control algorithm F1-REVCTL.

Configuring Maintenance Objects for Revision Control

Most configuration tool maintenance objects are pre-configured for revision control. You can configure other maintenance objects for revision control, as well.

To configure other objects for revision control:

1. Create a physical business object for the maintenance object, if it doesn't already have one. Refer to Creating a Physical Business Object for more information.
2. Add a foreign key reference to the maintenance object (if it doesn't already have one) so that the standard information string is displayed. Foreign key references are used for many purposes and most maintenance objects should have them anyway. Check the Maintenance Object options to see if there is an entry for the **Foreign Key Reference** option. If one is not configured, navigate to FK Reference and set up a foreign key

Copyright © 2026, Oracle and/or its affiliates.

reference for the primary table of the maintenance object. Configure that new FK reference on the maintenance object using an option.

3. Add the **Revision Control** zone to the maintenance object.
    a. Navigate to Context Sensitive Zone and search for the navigation key for the maintenance object.
    b. Add the Revision Control zone, F1-OBJREVCTL, to that navigation key.
4. Add the revision control algorithm to the maintenance object.
    a. Go to the Maintenance Object page and search for the maintenance object that you want to have revision control.
    b. On the **Algorithms** tab of the maintenance object, add the revision control algorithm F1-REVCTL.

Working with the Revision Control Zones

You use two zones in the sidebar to work with revision controlled objects when revision control is turned on.

The **Revision Control** zone gives you several options for managing the revision of the currently displayed object. This zone also shows when the object was last revised and by whom. This information is linked to the **Revision History** portal which lists all of the versions of the object.

Using the Revision Control zone you can:

- Check out an object in order to change it.
- Check in an object so others will be able to work on it.
- Revert the object back to where it was at the last checkout.
- Force a check in of an object that is checked out by someone else. You need special access rights to force a check in.
- Compare the latest version of the object to the previous checked in version.
- Delete an object.

See the subsequent sections for more information about each action.

The **Checked Out** zone lists all of the objects that you currently have checked out. Clicking on an object listed in this zone will take you to the page for that object. The zone is collapsed if you have no objects checked out.

See Revision Control Portal for more information about Check In, Force Check In, and Check Out one or more records simultaneously.

Checking Out an Object

If an entity's maintenance object is configured for revision control, you must check out the entity in order to change it. Go to the object you want to work on and click **Check Out** in the Revision Control sidebar zone.

The following points highlight behavior related to this functionality:

- When you create a record for a maintenance object that is managed with revision control, the system automatically associates the new record with a revision control in Checked Out status.

Copyright © 2026, Oracle and/or its affiliates.

- When you update an object governed by revision control, if you forget to check it out first, a dialog box informs you that the object is under revision control. You can select **OK** to check out the object and save your changes, or **Cancel** to stop the update.
- If another user has the object checked out, you will not be able to make changes.

> **Note**
>
> The system also supports Mass Check Out from the Revision Control search portal.

Checking In an Object

Once you are finished with changes, you should check in the object. This captures the latest revision and allows others to check it out.

While viewing the object, click **Check In** in the Revision Control sidebar zone. You are prompted for an **External Reference** and **Detailed Description**. It is recommended to put meaningful information in both of these fields to allow for future users that review changes to this object have a good understanding of the reason for the change.

The **Keep Checked Out** option is useful if you are checking in a version of the object that is stable but allows you to keep it checked out for further changes. The latest revision is one available for you to restore to later if you want to back out newer changes.

If you are using Bundling to manage copying objects to another environment, use the Add to Bundle option to automatically include this object in a current open bundle.

> **Note**
>
> The system also supports Mass Check In from the Revision Control search portal.

Reverting Changes

Reverting changes will undo any changes you made since you checked out an object.

While viewing the object, click **Revert** in the Revision Control sidebar zone. Once your change is reverted, the object is available to be checked out by another user.

> **Note**
>
> To restore the version of the object to a different revision than the most recent checked in revision, drill into the object's revision history. Refer to Revision Control Portal for more information.

> **Note**
>
> Any change to the object increments the version number. If you make changes and then revert your changes, internally the system updates the object to match the previously checked in revision. The version number, however, is incremented. It is not restored to the version number of the previous revision.

Copyright © 2026, Oracle and/or its affiliates.

Comparing Versions

When viewing an object that is governed by revision control, a link on the Revision Control sidebar zone allows you to **Compare to Previous Version**. The comparison works as follows:

- When viewing an object that is currently checked out, clicking the link compares the current view of the object in the database with the snapshot view of the object from its most recent checked in revision.
- When viewing an object that is checked in, when clicking the link, the behavior differs based on whether or not the current view of the object matches the most recent revision.
  - If they match, you are shown the comparison between the most recent check in to the check in before that.
  - If they do not match, it means that some process changed the object outside of revisions. In that case, you are shown the difference between the current view of the object and the most recent revision.

> **Note**
>
> The revision history view of an object in the Revision Control Portal allows you to select any two revisions to compare.

Forcing a Check In or Restore

If an object is checked out by another user and that person is not available to check it in or revert it, you may force a check in or force a restore (which is equivalent to a Revert action).

While viewing the object, click **Force Check In** in the Revision Control sidebar zone to keep the latest updates made by the checked out user. Click **Force Restore** in the Revision Control sidebar zone to discard the changes made by the checked out user and revert the object to the previously checked in revision.

> **Note**
>
> A user can only perform the Force Check In or Force Restore actions if they have security for the **Force Check In** and **Restore** access modes, respectively for the Object Revision BO application service (**F1-OBJREVBOAS**).

> **Note**
>
> The system also supports Mass Force Check In from the Revision Control search portal.

Deleting an Object

If revision control is turned on for an object, you must use the **Revision Control** zone to delete it. The object must be checked in before it can be deleted.

While viewing the checked in object, click **Delete** in the Revision Control sidebar zone. You are prompted for an **External Reference** and **Detailed Description**. Provide this information and then click OK. The system creates a revision record before the object is deleted so that the deleted object can be restored.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the object is maintained on a legacy UI page, you remain on the page with a cleared input form. If the object is maintained on a portal page, after deleting an object, you remain on the maintenance portal page, which should now be blank, which is unusual. Unlike deleting an object using the record actions on the record's maintenance page, deleting from the sidebar zone lacks the metadata to bring the user to a better portal after deletion, such as the related query portal for the deleted object.

Revision Control Portal

The **Revision History** portal lists the revision control history for a given object.

You can navigate to the **Revision History** portal from either a link in the **Revision Control** sidebar zone from that object's maintenance page or by going to **Revision Control** portal through the **Admin Menu** and using the search function to find the object whose history you want to view.

Use the broadcast icon to view the details of a given entry in the revision history.

To view the differences between two revision entries, click any two entries and then use the **Compare** button to see the differences.

You can restore a previous version of the object by clicking **Restore** in the row for the version that you want to restore. You would also use this option if an object was deleted and you want to reinstate it to a specific revision from before it was deleted.

See Working with Revision Control Zones for more information about tasks that can be performed through the Revision Control sidebar zone.

The following sections highlight other actions that are supported using the Revision Control Search portal besides searching for a specific revision. Navigate to this portal via **Admin Menu**, and then **Implementation Tools**, and then **Revision Control.**

Mass Check In

To perform a mass check in of several objects that you have checked out, choose **Check In** from the Search By dropdown on Revision Control Search.

Use the filters to narrow down the list of objects that you have checked out, if desired.

Select the desired entries and click **Check In**.

Mass Force Check In

If there are entities that are currently checked out by other users and they are not able to check them in, you can do a mass Force Check In of those objects. Choose **Force Check In** from the Search By dropdown.

Use the filters, including the Checked Out By User filter, to narrow down the list of objects to force check in on.

Select the desired entries and click **Force Check In**.

Mass Check Out

If you plan to work on updates to several objects, you can opt to proactively check out the objects en masse. Choose **Check Out** from the Search By dropdown.

Copyright © 2026, Oracle and/or its affiliates.

Use the filters to narrow down the list of objects to check out. Note that only objects that are currently checked in are available to check out.

Select the desired entries and click **Check Out**.

Information Lifecycle Management

Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle.

Data lifecycle typically refers to the fact that the most recent data is active in the system. As time progresses, the same data becomes old and unused. Older data becomes overhead to the application not only in terms of storage, but also in terms of performance. This older data's impact can be reduced by using advanced compression techniques, and can be put into slower and cheaper storage media. Depending on how often it's accessed, it can be removed from the system to make an overall savings of cost and performance. The target tables for ILM are transactional tables that have the potential to grow and become voluminous over time.

- The Approach to Implementing Information Lifecycle Management
- Enabling ILM for Supported Maintenance Objects
- Ongoing ILM Tasks
- Archived Foreign Keys

The Approach to Implementing Information Lifecycle Management

This section describes the product approach to implementing ILM for its maintenance objects (MOs).

> **Note**
>
> The term archiving is used to cover any of the possible steps an implementation may take in their data management strategy, including compression, moving to cheaper storage, and removing the data altogether.

Age is the starting point of the ILM product implementation for some of its high volume data. In general "old" records are considered eligible to be archived. In the product solution, maintenance objects (MOs) that are enabled for ILM have an ILM Date on the primary table and the date is typically set to the record's creation date. (An MO may have special business rules for setting this date, in which case, a different date may be used to set the initial ILM Date). For implementations that want to use ILM to manage the records in the MO, the ILM date is used for defining partitions for the primary table.

There are cases where a record's age is not the only factor in determining whether or not it is eligible to be archived. There may be some MOs where an old record is still 'in progress' or 'active' and should not be archived. There may be other MOs where certain records should never be archived. To evaluate archive eligibility using information other than the ILM Date, the ILM enabled MOs include an ILM Archive switch that is used to explicitly mark records that have been evaluated and should be archived. This allows DBAs to monitor partitions based on age and the value of this switch to evaluate data that may be ready to be archived.

Evaluating records to determine their archive eligibility should still occur on "old" records. The expectation is that a large percentage of the old records will be eligible for archiving. The small

Copyright © 2026, Oracle and/or its affiliates.

number that may be ineligible could be updated with a more recent ILM date. This may cause the records to move into a different partition and can delay any further evaluation of those records until more time has passed.

For each MO enabled for ILM, the product provides a batch process to review "old" records and an ILM eligibility algorithm that contains business logic to evaluate the record and mark it eligible for archiving or not. The following sections provide more information about the batch process and algorithm functionality.

- Batch Processes
- Eligibility Algorithm

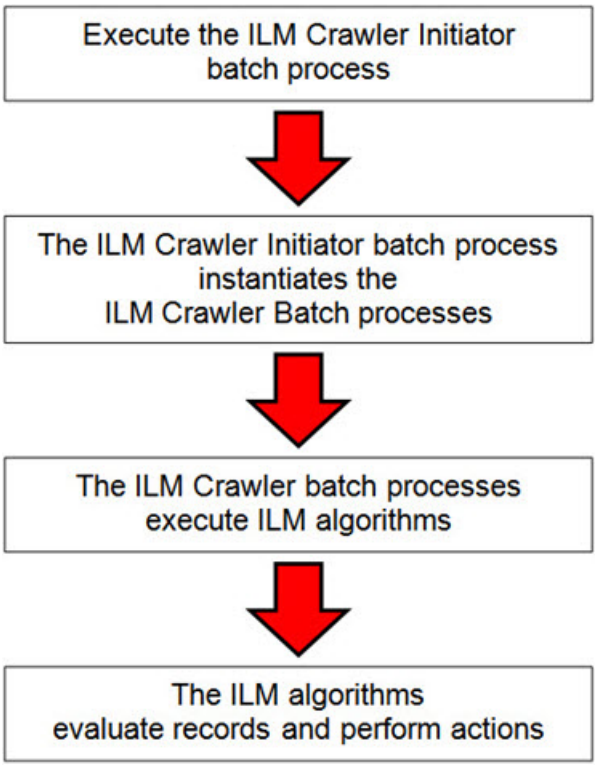
Batch Processes

There are two main types of batch processes that manage data for ILM in the application: ILM Crawler Initiator and individual ILM Crawlers (one for each MO that is configured for ILM).

- **ILM Crawler Initiator**: (`F1-ILMIN`) - The ILM Crawler Initiator is a *driver* batch process that starts the individual ILM Crawler batch control as defined by the MO's options.

  Restartable: In case of server failure, the ILM Crawler Initiator process can be restarted, which will also restart the ILM Crawler processes.

- **ILM Crawler**: Each maintenance object that is configured for ILM defines an ILM Crawler. These are *child* batch processes that can be started either by the ILM Crawler Initiator or by a standalone batch submission.



The ILM Crawler batch process selects records whose retention period has elapsed and invokes the MO's ILM eligibility algorithm to determine if the record is ready to be archived or not. The ILM eligibility algorithm is responsible for setting the record's ILM archive switch to 'Y' and updating the ILM date, if necessary.

Copyright © 2026, Oracle and/or its affiliates.

The retention period defines the period that records are considered active. It spans the system date and cutoff date (calculated as system date - retention days).

The retention days of an MO is derived as follows:

- If the ILM Retention Days MO option is defined, that is used.
- Otherwise, the Default Retention Days from the ILM Master Configuration record is used.

An error is issued if no retention period is found.

The crawler calculates the cutoff date and selects all records whose ILM archive switch is 'N' and whose ILM date is prior to the cutoff date. Each record returned is subject to ILM eligibility.

If the Override Cutoff Date parameter is supplied, it will be used instead of the calculated cutoff date. An error is issued if the override cutoff date is later than the calculated cutoff date. This parameter is useful if an object has many years of historic data eligible for archiving. Setting this parameter allows for widening the retention period and therefore limiting the process to a shorter period for initial processing.

> **Note**
>
> ILM Crawler batch processes are designed not to interfere with current online or batch processing. Because of this, these batch processes can run throughout the day.

> **Note**
>
> Before passing the cut-off date to the algorithm, the ILM crawler ensures that the number of days calculated (System Date – override cut-off date) is more than the retention period specified in the MO option or the Master Configuration. If the number of days calculated **is less than** the retention period specified on the MO option or the Master Configuration, then it throws an error.

Eligibility Algorithm

Algorithms are triggered by the ILM batch crawler for the maintenance object. The key responsibility of the ILM algorithm is to determine whether a record can be marked as ready to be archived or not. If a record is determined to be ready for archive, the algorithm should set the ILM Archive switch to Y. If not, the algorithm leaves the switch set to N and may decide to update the ILM Date to something more recent (like the System Date) to ensure that the record does not get evaluated again until the future.

This algorithm is plugged into the Maintenance Object — Algorithm collection.

Oracle Utilities Application Framework provides the algorithm **ILM Eligibility Based on Status** (`F1-ILMELIG`) to support the ILM batch crawler. Refer to the algorithm type description for details about how this algorithm works. If a maintenance object has special business rules that are evaluated to determine the eligibility for ILM, a custom algorithm can be created and applied by the implementation team.

Enabling ILM for Supported Maintenance Objects

In order to enable ILM for one or more maintenance objects, several steps are needed in both the configuration and in the database. This section describes some configuration enabled by

Copyright © 2026, Oracle and/or its affiliates.

default and some steps that must be taken in order to fully implement ILM. While some configuration is enabled by default, it will not be used unless ILM is fully configured.

Managing a maintenance object's data for ILM involves the following aspects:

- Determining which records are eligible for archiving and marking them as such. Depending on the entity, this step is implemented using one of this methods:
  - **Configurable.** This option relies on a configurable eligibility algorithm that is processed in batch periodically.
  - **Non-configurable batch.** This option uses a specific batch process that embeds non-configurable eligibility rules as part of its logic for performance reasons.
  - **Set at creation time.** Records in very high volume tables that rarely change are marked as eligible for archival at creation time for performance reasons. For example, measurement data, some snapshot tables, etc.
- Partitioning the entity's tables and archiving old partitions as needed.

Each maintenance object that follows the configurable option has the following provided out of the box:

- **Special Table Columns**: Maintenance objects that support ILM include two specific columns: ILM Archive Switch (`ILM_ARCH_SW`) and ILM Date (`ILM_DT`).
- **Crawler Batch Process**: A "crawler" batch process is provided for each maintenance object that supports ILM and it is plugged into the MO as an option. Refer to Batch Processes for more information.
- **ILM Eligibility Algorithm**: Each maintenance object that is configured for ILM defines an eligibility algorithm that executes the logic to set the ILM Archive switch appropriately. This is plugged in to the MO algorithm collection.

If an implementation decides to implement ILM, there are steps that need to be followed, which are highlighted below.

Create the Master Configuration Record

The first step when enabling ILM is to create the **ILM Configuration** master configuration record.

The master configuration for ILM Configuration defines global parameters for all ILM eligible maintenance objects. For example, the Default Retention Period. In addition, your product may implement additional configuration. Refer to the embedded help for specific details about the information supported for your product's ILM configuration.

In addition, the user interface for this master configuration record displays summary information about all the maintenance objects that are configured to use ILM.

Confirm the Maintenance Objects to Enable

In viewing the list of maintenance objects that support ILM in the ILM master configuration page, your implementation may choose to enable ILM for only a subset of the supported maintenance objects. For example, some of the maintenance objects may not be relevant for your implementation. Or perhaps, the functionality provided by the maintenance object is used, but your implementation does not expect a high volume of data.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Customizing the list of maintenance objects managed by ILM is not supported in a cloud installation.

For each maintenance object that your implementation has confirmed for ILM, the following steps should be taken:

- Determine if the maintenance object should have a different number of default retention days than the system wide value defined on the master configuration. If so, use the MO option **ILM Retention Period in Days** to enter an override option for this maintenance object.
- When applicable, review the functionality of the ILM Eligibility algorithm provided by the product for this maintenance object. Each algorithm may support additional configuration based on business needs. If your organization has special business rules that aren't satisfied by the algorithm provided by the product, a custom algorithm may be provided to override the base algorithm.

For each maintenance object that follows the configurable option and that your implementation does not want to enable for ILM, inactivate the eligibility algorithm. This will ensure that the ILM Crawler Initiator background process does not submit the crawler batch job for the maintenance object in question.

- Go to the Maintenance Object - Algorithm tab for each maintenance object and inactivate the **ILM Eligibility** algorithm.

Database Administrator Tasks

In order to implement ILM for each of the maintenance objects determined above, your database administrator must perform several steps in the database for the tables related to each MO. The following points are a summary of those steps. More detail can be found in the Information Lifecycle Management section of your product's *Database Administration Guide*.

- **Initializing ILM Date**: Your existing tables that are enabled for ILM may not have the ILM Date and ILM Archive switch initialized on all existing records. When choosing to enable ILM, a first step is to initialize this data based on recommendations provided in the *Database Administration Guide*.
- **Referential Integrity**: The recommended partitioning strategy for child tables in a maintenance object is referential partitioning. In order to implement this, database referential integrity features must be enabled.
- **Partitioning**: This provides a way in which the data can segregate into multiple table partitions and will help in better management of the lifecycle of the data.

Schedule the ILM Crawler Initiator

The final step of enabling the system for ILM is to schedule the ILM crawler initiator `F1-ILMIN` regularly based on your implementation's need. It is recommended to only schedule this batch process once all the required database activities are complete.

Ongoing ILM Tasks

For an environment where ILM is enabled, besides the periodic execution of the ILM crawler batch processes to review and mark records, your database administrator has ongoing tasks.

Copyright © 2026, Oracle and/or its affiliates.

The DBA reviews and maintains partitions and identifies partitions that may warrant some type of archiving step. The Information Lifecycle Management section of your product's *Database Administration Guide* provides more information for your DBA.

Archived Foreign Keys

If your DBA chooses to archive a partition, there may be records in the system that refer to one of the archived records as a foreign key.

When a user attempts to view a record using a hyperlink or drill down mechanism, if the implementation of the navigation uses FK Reference functionality, the system will first check if the record exists. If not, it will display a message to the user indicating that the record has been archived.

Configuration Tools

This section describes tools to facilitate detailed business configuration. The configuration tools allow you to extend both the front-end user interface as well as create and define specialized back-end services.

- Business Objects
- Data Areas
- Business Services
- User Interface (UI) Options
- Process Flows
- Action Method
- Algorithms
- Script Options
- Maintaining Managed Content
- Defining Entity Tags
- Advanced Schema Topics
- Business Event Log
- Miscellaneous Topics
- Extensions Dashboard

Business Objects

A maintenance object defines the physical tables that are used to capture all the possible details for an entity in the system. A business object is tool provided to further define business rules for a maintenance object.

This section provides an overview of business objects and describes how to maintain them.

- Understanding Business Objects
- Defining Business Objects
- Advanced BO Tips and Techniques

Copyright © 2026, Oracle and/or its affiliates.

- Defining Status Reasons

Understanding Business Objects

The topics in this section describe background topics relevant to business objects.

- What Is A Business Object?
- Business Object Inheritance
- Each Business Object Can Have A Different Lifecycle
- BO Algorithm Execution Summary
- Granting Access To Business Objects

What Is A Business Object?

A business object (BO) is a powerful tool used throughout the system. For many maintenance objects, a BO is a key attribute of the record used to define the data it captures, its user interface behavior and its business rules. Some business objects support the definition of a lifecycle, capturing different states that a record may go through, allowing for different business rules to be executed along the way. This type of business object is considered the "identifying" or "governing" business object. We will see later that other types of BOs exist that are different from the "identifying" business object.

The use of business objects allows for extensibility and customization of product delivered maintenance objects. There are many options to adjust the behavior of base delivered business objects. In addition, implementations may introduce their own business objects if the base product delivered objects do not meet their business needs.

> **Note**
>
> Not all maintenance objects in the product support business objects as a "identifying" or "governing" tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

- A Business Object Has a Schema
- A Business Object May Define Business Rules
- A Business Object Defines its User Interface
- A Business Object Defines Options
- Invoking A Business Object
- Determine the Identifying BO
- Base Business Objects

A Business Object Has a Schema

A business object has elements. The elements are a logical view of fields and columns in one of the maintenance object's tables. The structure of a business object is defined using an XML schema. The main purpose of the schema is to identify all the elements from the maintenance object that are included in the business object and map them to the corresponding maintenance object fields. Every element in the BO schema must be stored somewhere in the maintenance object. The BO may not define elements that are derived.

Copyright © 2026, Oracle and/or its affiliates.

When defining elements for the primary table or the language table (for an administrative object) there is no need to define the name of the physical table in the schema. The system infers this information. The following is a snippet of a schema:

```
<schema>
    <migrationPlan mapField="MIGR_PLAN_CD" suppress="true" isPrimeKey="true"/>
    <bo mapField="BUS_OBJ_CD" fkRef="F1-BUSOB"/>
    <customizationOwner mapField="OWNER_FLG" suppress="input"/>
    <version mapField="VERSION" suppress="true"/>
    <description mapField="DESCR"/>
    <longDescription mapField="DESCRLONG"/>
  â€¦
```

Many maintenance objects have child table collections (e.g., a collection of names for a person, or a collection of persons on an account). Depending on the requirements, the business object may define the full collection such that the user will maintain the information in a grid. However, the schema also supports "flattening" records in a child table so that they can be treated as if they were singular elements. The following are examples of each:

Example of a child table. This is a snippet of the Instructions collection on the migration plan business object. You can see that the list attribute defines the child table and all elements within it map to the appropriate column in that table.

```
   <migrationPlanInstruction type="list" mapChild="F1_MIGR_PLAN_INSTR">
         <migrationPlan mapField="MIGR_PLAN_CD" suppress="true"/>
         <sequence mapField="PLAN_INSTR_SEQ" suppress="true"/>
         <instructionSequence mapField="INSTR_SEQ"/>
         <instructionType mapField="INSTR_TYPE_FLG"/>
         <parentInstructionSequence mapField="PARENT_INSTR_SEQ"/>
         <businessObject mapField="BUS_OBJ_CD" fkRef="F1-BOMO"/>
...
```

Example of a simple "flattened" field. The business object for Status Reason includes an element called Usage, which maps to a pre-defined characteristic of type **F1-SRUSG**. The "row" defines which child table is being flattened and the attributes of the row in that child that uniquely identify it.

```
    <usage mdField="STATUS_RSN_USAGE" mapField="CHAR_VAL">
        <row mapChild="F1_BUS_OBJ_STATUS_RSN_CHAR">
            <CHAR_TYPE_CD is="F1-SRUSG"/>
            <SEQ_NUM is="1"/>
        </row>
    </usage>
```

Example of a "flattened row". This business object for Account includes a single row for the Person collection where only the "financially responsible, main" customer is defined. The "accountPerson" attribute defines one field from that row (the Person Id) and includes the "˜flattening" criteria in the "row" information. In addition, a second field from that same row ("accountRelType") is defined. Instead of having to repeat the flattening criteria, the "rowRef" attribute identifies the element that includes the flattening.

```
    <accountPerson mapField="PER_ID">
        <row mapChild="CI_ACCT_PER">
            <MAIN_CUST_SW is="true"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <FIN_RESP_SW default="true"/>
          </row>
     </accountPerson>
     <accountRelType mapField="ACCT_REL_TYPE_CD" rowRef="accountPerson"
dataType="string"/>
```

Example of a "flattened list". The business object for Tax Bill Type includes an list of valid algorithms for "bill completion". The Sequence and the Algorithm are presented in a list. The list element identifies the child table and the "˜rowFilter" identifies the information about the list that is common.

```
       <taxBillCompletion type="list" mapChild="C1_TAX_BILL_TYPE_ALG">
         <rowFilter suppress="true" private="true">
             <TAX_BILL_TYPE_SEVT_FLG is="C1BC"/>
         </rowFilter>
         <sequence mapField="SEQ_NUM"/>
         <algorithm mapField="ALG_CD" fkRef="F1-ALG"/>
     </taxBillCompletion>
```

In addition, many maintenance objects support an XML structure field within the entity. These fields may be of data type CLOB or XML. One or more business object elements may be mapped to the MO's XML structure field. These elements may be defined in different logical places in the business object schema based on what makes sense for the business rules. When updating the MO, the system builds a type of XML document that includes all the elements mapped to the XML structure and stores it in one column. The following is an example of elements mapped to an XML column:

```
        <filePath mdField="F1_FILE_PATH" mapXML="MST_CONFIG_DATA"
required="true"/>
        <characterEncoding mdField="F1_CHAR_ENCODING"
mapXML="MST_CONFIG_DATA"/>
```

> **Note**
>
> If the MO's XML structure field is of the data type XML, the database will allow searching for records based on that data, assuming appropriate indexes are defined. If the MO's XML structure field is of the data type CLOB, indexing or joining to elements in this column via an SQL statement is not typically supported. Note that most MOs are currently using the CLOB data type for the XML structure column, if provided.

Some business objects may have child tables that allow data to be stored in an XML structure field. The schema language supports defining elements from those fields in your schema as well.

Besides including information about the physical "mapping" of the element to its appropriate table / field location in the maintenance object, the schema supports additional syntax to provide the ability to define basic validation and data manipulation rules, including:

- Identifying the primary key of the record or the primary key of the a row in a list.
- Identifying which elements are required when adding or changing a record.
- Default values when no data is supplied on an Add.

Copyright © 2026, Oracle and/or its affiliates.

- For elements that are lookup values, the lookup may be specified to validate that the value of the element is a valid lookup value.
- For elements that are foreign keys to another table, the FK Reference may be specified to validate the data.

The system will check the validity of the data based on the schema definition obviating the need for any special algorithm to check this validation.

In addition, the schema language may include some attributes that are used to auto-render the view of the record on the user interface, such as the `suppress` attribute. Refer to BO Defines its User Interface for more information.

> **Note**
>
> Refer to Schema Syntax for the complete list of the XML nodes and attributes available to you when you construct a schema.

A business object's schema may include a subset of the fields and tables defined in the maintenance object. There are two reasons for this:

- The fields or tables may not be applicable to the type of record the business object is governing. For example, a field that is specific to gas may not be included on a Device business object that is specific to electric meters.
- The information is not maintained through the business object, but rather maintained separately. For example, many BO based maintenance objects include a Log table. The records in the log table are typically not included the BO because they are viewed and maintained separately from the business object.

A Business Object May Define Business Rules

A business object may define business rules that govern the behavior of entities of this type. The following table describes supported system events that can be associated with such rules.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Audit	Optional	Algorithms of this type may be used to audit certain changes made to business object instances. The system hands over to the algorithms a summary of all the elements that were changed throughout a specific call to update an object. For each element its original value before the change as well as its new value are provided. It is the responsibility of such algorithms to log the changes if and where appropriate, for example as a log entry in the maintenance object's log table or an entry in the business event log. These plug-ins are executed after all the post-processing rules are executed. The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information. By default all elements of the business object are subject to auditing. You can however mark certain elements to be excluded from the auditing process using the **noAudit** schema attribute. Marking an element as not auditable will prevent it from ever appearing as a changed element in the business object's audit plug-in spot. In addition, if the only elements that changed in a BO are ones marked to not audit, the audit algorithm is not even called. Refer to Schema Syntax for more information on this attribute.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Information	Optional	We use the term "Business Object Information" to describe the basic information that appears throughout the system to describe an entity defined by the business object. The data that appears in this information description is constructed using this algorithm. The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number found on the business object closest to the current business object in the inheritance hierarchy. Refer to Business Object inheritance for more information.
`Post-Processing`	Optional	Algorithms of this type may be used to perform additional business logic after a business object instance has been processed. For example, such an algorithm may be used to create a To Do Entry or add a log record as part of the business object logical transaction. These plug-ins are executed after all the validation rules are executed. The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
`Pre-Processing`	Optional	Algorithms of this type further populates a request to maintain a business object instance right before it is processed. Such algorithm may be used to "massage" a business object's elements prior to any maintenance object processing. For example, although simple element-level defaulting is supported by schema attributes. you may use this type of algorithm to default element values that are more sophisticated. The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.
Validation	Optional	Algorithms of this type may be used to validate a business object instance when added, updated or deleted. Simple element-level validation is supported by schema attributes. Note that element-level validation is executed before any maintenance object processing. For more sophisticated rules you create validation algorithms and associate them with your business object. BO validation algorithms are only executed after "core validation" held in the MO is passed. The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.

The system applies business object rules (schema based and algorithms) whenever a business object instance is added, changed or deleted. This is only possible when the call is made via the maintenance object service. For example, when made via business object interaction ("invoke BO"), the MO's maintenance page or inbound web services that reference the BO. In addition the system must be able to determine the identifying business object associated with the actual object being processed. If the business object cannot be determined for a maintenance object instance business object rules are not applied.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Pre-Processing is special.** The pre-processing algorithm plug-in spot is unique in that it only applies during a BO interaction. It is executed prior to any maintenance object processing. It means that when performing add, change or delete via the maintenance object service, the pre-processing plug-in is not executed.

> **Caution**
>
> Direct entity updates bypass business rules! As mentioned above, it is the maintenance object service layer that applies business object rules. Processes that directly update entities not via the maintenance object service bypass any business object rules you may have configured.

> **NOT_SUPPORTED**
>
> Refer to BO Algorithm Execution Order for a summary of when these algorithms are executed with respect to lifecycle algorithms.

The plug-in spots described above are available for all business objects and they are executed by the system when processing adds or updates to the business object. It is possible for a specific maintenance object to define a special plug-in spot for business objects of that MO. When this happens, the maintenance object identifies the special algorithm entity lookup value as an MO option: **Valid BO System Event**, causing the BO Algorithm collection to include that system event in its list.

A Business Object Defines its User Interface

One of the responsibilities of an identifying business object is to define its user interface rules for viewing and maintenance of its record. The standard implementation for maintaining a business object is that a maintenance portal is used to display a record. This portal includes a "map" zone that displays the information about the business object. To add or make changes to a record, the user clicks a button that launches a maintenance BPA script which displays a maintenance "map".

The display and maintenance "maps" are driven by the business object.

- **Schema based UI.** This is the preferred option. The BO's schema definition and UI Hints define user interface behavior that results in generated HTML. The schema language includes some basic display attributes such as **label** and **suppress**. UI hints provide many additional tags and elements that allow dynamic generation of formatted UI Maps. For more complex behavior in the user interface, for example where javascript is needed, UI map fragments may be defined within the schema via UI hints. In this way only complex UI behavior warrants small snippets of javascript and HTML. However the rendering of standard fields can be dynamically rendered. UI map fragments also allow for derived fields to be included in the user interface.
- **Explicit UI map.** The BO may rarely define a full UI map where all the information is displayed based on the map's HTML. Note that for a child BO, the maps may be inherited by a parent BO (or any BO "up the chain"). This technique is less flexible for describing a business object and as such is mainly supported for backward compatibility.

Copyright © 2026, Oracle and/or its affiliates.

A business object schema may include data areas for segments of its schema definition to allow for reusable components. In this case the data area would also include schema attributes and UI hints for the elements that it is including.

> **Note**
>
> Refer to UI Hint Syntax for detailed information about the supported syntax.

As mentioned in Business Object Inheritance, schemas are not inherited on a child business object. As such, when using UI hints for automatic UI rendering, the child BO must define the full schema with all the definitions. A good business object hierarchy will be designed for reuse meaning that the child BO will include the parent BO schema or alternatively, the BO schemas will include reusable data areas.

By default, the input form used to add or update an entity is presented in the large drawer panel. If the small or medium drawer panel fits better to display this form, you can configure the **Maintenance Drawer Size** BO option accordingly.

A Business Object Defines Options

A business object may be associated with configuration options that control specific features. For example, the scripts used to maintain and display a business object are defined as options. Refer to the detailed description of each option type for more information on the functionality it controls.

> **Note**
>
> **You can add new options types**. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field `BUS_OBJ_OPT_FLG`. If you add a new option type for a business option, you must update its maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Option Type** maintenance object option.

Some options may be defined for a specific lifecycle status. For example, marking a status specific base algorithm as inactive is defined as an option on that staus definition record.

> **Note**
>
> **You can add new options types**. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field `BO_OPT_FLG`. If you add a new option type for a status, you must update the business object's maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Status Option Type** maintenance object option.

Copyright © 2026, Oracle and/or its affiliates.

Invoking A Business Object

We have talked about defining a business object. This section describes how business objects are used throughout the system to view, add and update records.

- Various parameters for zones that are used to display data in the system include support for retrieving data by referencing a business object. The zone code will "invoke" the BO, meaning that the record will be retrieved using the referenced BO.
- The system's scripting language includes a step type to "invoke BO". This allows for BPA scripts, service scripts and plug-in scripts to retrieve information and add or update records using BO interaction.
- Inbound web services may reference a business object in its operations collection. This allows external systems to add or update records in our product via web service interaction.

Often when configuring a zone or writing a script, the BO to use in the "invoke BO" statement should be the identifying BO of the record. As such, often the script will include steps prior to the "invoke BO" step to "determine the identifying BO of the record" and once the identifying BO is found, the script step will invoke that BO. Note that zones and inbound web services reference a BO directly. In each case, if the BO to use should be dynamic, then the zone / inbound web service should reference a service script that can perform the steps to identify the BO and then invoke that BO.

It should be noted however that the BO used in an "invoke BO" statement (or referenced in an inbound web service) **does not have to match** the identifying BO for the record. Here are some examples of where this may be true:

- A script may only require a subset of elements for a record and not the entire record. In this case, it is better for performance purposes to define a special BO (sometimes called a "lite" BO or a "mini" BO) that only defines the needed elements. When the system retrieves the data, it will only access the tables that are included in the BO's schema definition. In addition, if there are no elements that map to an XML structure field, the system will skip any parsing of that column. Similarly, if a script is **updating** a subset of elements on a record, it may be beneficial to use a "mini" BO to do the updates.

  > **Note**
  >
  > Please note the following with respect to using a mini BO. This BO is only used for its schema. This type of BO would not define algorithms or a lifecycle. Because the BO is special, it often should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to Determine the Identifying BO for more information.

- The maintenance object to be added or updated in a script may not support business objects as "identifying BOs". For example, Batch Control maintenance object does not have an identifying BO. However, scripts may still wish to retrieve data (or make updates) to these types of records. An easy way to achieve that goal is to define a business object and use "invoke BO" to access the data.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Not all maintenance objects support being maintained through a business object interaction. This is true in a small number of older objects where the underlying maintenance service includes additional functionality besides simply updating the database tables. These maintenance objects are identified via the MO option **BO Maintenance**, set to `N`.

- Some functionality may be trying to add or update records for a maintenance object in a 'physical' manner and do not want or need to use the object's identifying BO. Or the MO may not have an identifying BO. For example, revision control takes a snapshot of a record for audit purpose and to be able to restore a previous version. In this case, the system wants to capture a full "physical" view of the record. To do this, a special "physical" BO may be created that includes all (or most of) the columns and the child tables.

> **Note**
>
> Like the mini BO, the physical BO would not define algorithms or a lifecycle and should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to Determine the Identifying BO for more information.

> **Note**
>
> To reiterate, the BO referenced in the "invoke BO" statement or referenced in an inbound web service does not have to match the identifying BO and does not have to be configured to "allow new instances".

Determine the Identifying BO

As mentioned in other topics, the identifying BO is the business object that governs the business rules for a record. This is the business object that the record will be validated against when any additions or changes are made to the record as long as updates are made via the maintenance service. This includes using "invoke BO" for add or update, using inbound web service interaction and for access to the maintenance page service (via an old style fixed page or via a business service).

How does the system determine the identifying BO? An algorithm plugged into the maintenance object (the **Determine BO** plug-in spot) is responsible for this. If the maintenance object is not configured with an algorithm for this plug-in spot, or no BO is found by the algorithm, no BO business rules are applied.

Most maintenance objects in the system capture the record's identifying BO directly on the record. However, it is possible to define the identifying BO somewhere else. For example, there may be some maintenance objects that are master or transaction objects with an associated "type" object where the identifying BO is defined on its "type" object. Note that the standard Determine BO algorithm plugged into most maintenance objects (**F1-STD-DTMBO - Determine Standard Business Object**) checks for these two conditions.

There may also be cases where a single identifying BO is used for all BOs for a given MO. This may be an option used for some older maintenance object created prior to the business object functionality when implementations wish to introduce custom business rules that are

Copyright © 2026, Oracle and/or its affiliates.

common for all records of that MO. The product provides a base algorithm type (**F1–MOBO - Determine Specific Business Object**) that captures the BO as a parameter.

Base Business Objects

For each maintenance object (MO) that supports an "identifying" business object, the type of business object provided by the product depends on the functionality and expected use by implementations. The following are some common patterns.

- There are MOs where the product provides base BOs that implementations may use if applicable for their business rules. In addition, it is expected that implementation will define custom BOs to support their business needs. Good examples of this type of MO are any of the various "rule" MOs. For example, calculation rule in Oracle Utilities Customer Care and Billing or the usage rule in Oracle Utilities Meter Data Management. The product provides business objects for common rules but each implementation could have special rules that they need to implement and will need to create custom business objects.
- There are MOs where the product provides base BOs that supply common behavior for an object. Implementations may find that supplied the business objects match their business requirements and use the BOs as is. It is expected, however that for many implementations, their business rules will require additional elements to be captured or have special rules to apply. In this case the base business objects may be extended. This scenario may apply to 'master' data objects in various products such as the Device or Meter.
- There are MOs where the product may deliver a base BO that is not expected to satisfy most implementations because different jurisdictions or different implementations will typically have their own rules. In this case the base delivered BO can be used as a template or starting point for custom defined BOs. Some examples of this are Rebate Claim in Oracle Utilities Customer Care and Billing.
- There are MOs where the expectation is that every implementation will have different requirements for the type of data to capture and the product will not supply base BOs that can be used as the "identifying" BO. However, it may supply a "parent" BO that defines the lifecycle and many of the business rules that it expects all records to follow. In these scenarios, the implementations will create "child" BOs that will serve as the "identifying" BOs and refer to the base "parent" BO for many of its rules through inheritance. An example of this is the Activity in Oracle Utilities Mobile Workforce Management.
- There are some scenarios where the base product provides business objects and the expectation is that implementations will use the business objects as delivered with little or no customization. This is a case where the system used business objects to implement product functionality, not because there is an expectation that the implementers will extend the functionality, but because the business object model is the favored development tool even for the product. The objects delivered for Content Migration Assistant are an example.

> **Note**
>
> Not all maintenance objects in the product support business objects as a "identifying" or "governing" tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

For all maintenance objects, the base product may provide additional BOs that are not meant to be "identifying" BOs, but instead are provided to support functionality to interact with the MO using the BO as a tool as described in Invoking a BO.

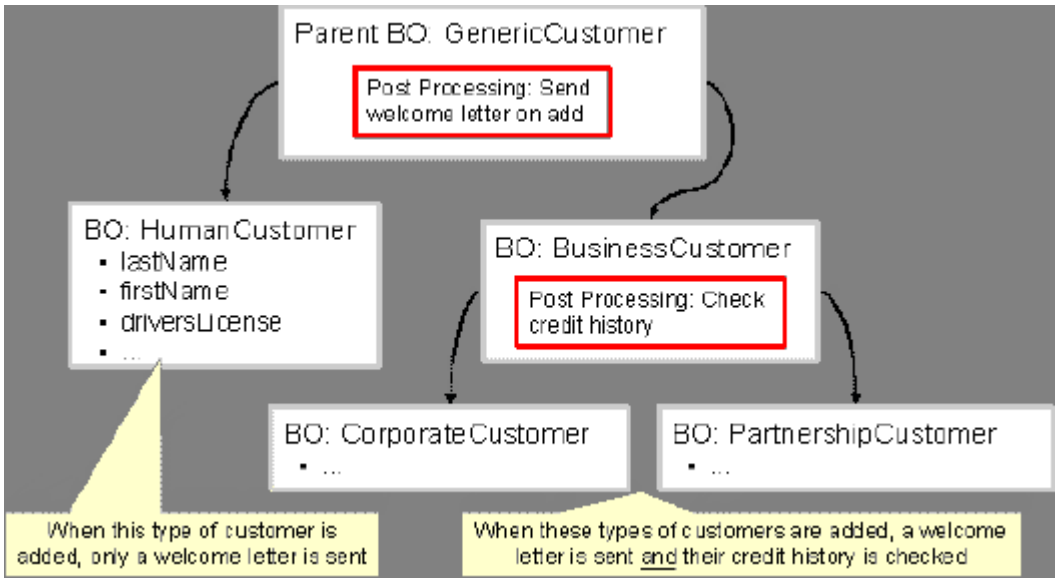
Copyright © 2026, Oracle and/or its affiliates.

- One or more "mini" or "lite" BOs may be supplied for a maintenance object. This may be found when the product has functionality to retrieve a subset of elements for the maintenance object via scripting or via a user interface.
- A "physical" BO may be supplied. This a BO that typically includes all tables and all fields of the maintenance object in there "physical" form. In other words, there is no "flattening" of child tables and any XML structure fields are defined as a single field. Physical BOs are used in system processing where the full record needs to be captured as is. Some functionality that uses a physical BO includes bundling, revision control and the pre-compare algorithm for CMA to adjust data prior to comparing. If there are any maintenance objects that do not have a base supplied physical BO, refer to Creating a Physical Business Object for steps on how to provide one.
- A "bundling add" BO may be supplied. Refer to Recursive Key References for more information as to why this type of BO may be supplied.

Business Object Inheritance

A business object may inherit business rules from another business object by referencing the latter as its parent. A child business object can also have children, and so on. A parent's rules automatically apply to all of its children (no compilation - it's immediate). A child business object can always introduce rules of its own but never remove or bypass an inherited rule.

The following is an illustration of multiple levels of business object inheritance.



Notice how the "Business Customer" business object extends its parent rules to also enforce a credit history check on all types of customers associated with its child business objects.

Most types of business object system events allows for multiple algorithms to be executed. For example, you can have multiple **Validation** algorithms defined for a business object. For these, the system executes all algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

Other types of system events allows for a single algorithm to be executed. For example, you can only have one **Information** algorithm to format the standard description of a business object instance. For these, the system executes the one at the level nearest to the business object currently being processed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The parent and its children must reference the same maintenance object.

> **Note**
>
> **Data structures are not inherited.** While you can declare schemas on parent business objects, their children will not inherit them. A good practice is to design child business object schemas to **include** their parent business object's schema.

> **Note**
>
> **User interface maps are inherited.** When determining if the business object has a UI map to use for UI rendering, the system looks for display and maintenance maps linked to the BO as options. If the identifying BO does not have maps defined, the system follows “up the chain” of inheritance until it finds a map to use. This allows for a child BO to be used to extend business rules of a parent BO but inherit its user interface behavior. Refer to Business Object Defines its User Interface for more information.

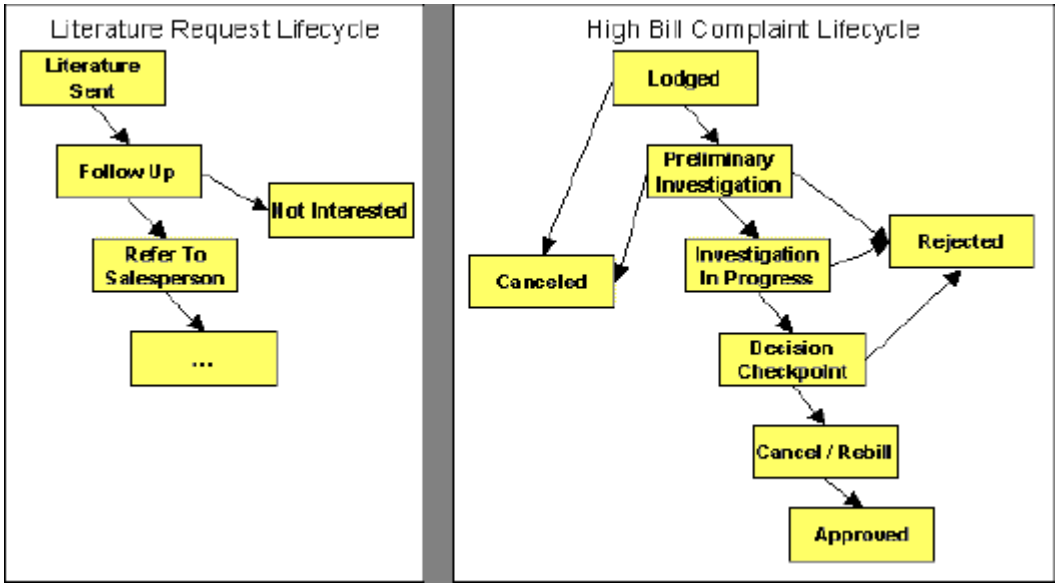
> **Note**
>
> **Use Inheritance Wisely.** While it is intellectually attractive to abstract behavior into parent BOs to avoid redundant logic and simplify maintenance, before doing this weigh the reuse benefits against the cost in transparency, as it is not easy to maintain a complex hierarchy of business objects.

Each Business Object Can Have A Different Lifecycle

Many maintenance objects have a status column that holds the business entity's current state within its lifecycle. Rules that govern lifecycle state transition (what is its initial state, when can it transition to another state, and so on.) and the behavior associated with each state are referred to as lifecycle rules. Older Maintenance Objects, such as To Do Entry, have predefined lifecycles whose rules are governed by the base-package and cannot be changed. The lifecycle of newer Maintenance Objects exists in business object meta-data and as such considered softly defined. This allows you to have completely different lifecycle rules for business objects belonging to the same maintenance object.

Here are examples of two business objects with different lifecycles thast belong to the same maintenance object.

Copyright © 2026, Oracle and/or its affiliates.



The boxes in the above diagram show the potential valid states a business entity of the above business object can be in. The lines between the boxes indicate the state transition rules. These rules govern the states it can move to while in a given state. For example, the above diagram indicates a high bill complaint that's in the **Lodged** state can be either **Canceled** or moved into the **Preliminary Investigation** state.

When you set up a business object, you define both its valid states and the state transition rules, including the label for the Action to display on a button for users that may initiate a transition. For example, the action label to transition to the 'Approved' state could be configured as 'Approve'.

> ⓘ **Note**
>
> The status description and the action label of base delivered values may be overridden by your implementation.

> ⓘ **Note**
>
> A Maintenance Object supports soft lifecycle rules if it is defined with a **Status Field** Maintenance Object option and that field is not a lookup field (which denotes a hard-coded lifecycle).

The topics that follow describe important lifecycle oriented concepts.

- One Initial State and Multiple Final States
- State-Specific Business Rules
- Inheriting Lifecycle
- Auto-Transition
- Keeping An Entity In Its Last Successful State
- Monitoring Batch Processes
- Transitory States

Copyright © 2026, Oracle and/or its affiliates.

- State Transitions Are Audited
- Required Elements Before Entering A State
- Capturing a Reason for Entering a State
- Opening UI Maps Before Entering A State

One Initial State and Multiple Final States

When you set up lifecycle states, you must pick one as the initial state. The initial state is the state assigned to new entities associated with the business object. For example, the above high-bill complaint business object defines an initial state of **Lodged**, whereas the literature request one defines an initial state of **Literature Sent**.

You must also define which statuses are considered to be "final". Typically when an entity reaches a "final" state, its lifecycle is considered complete and no further processing is necessary.

> **Note**
>
> **Allowing An Entity To Be "Reopened".** You can set up your state transition rules to allow a business entity to be "reopened" (i.e., to be moved from a final state to a non-final state). Neither of the above examples allows this, but it is possible if you configure your business object accordingly.

State-Specific Business Rules

For each state in a business object's lifecycle, you can define the following types of business rules.

System Event	Optional / Required	Description
Enter	Optional	Algorithms of this type apply business rules when a business object instance enters a given state.
Exit	Optional	Algorithms of this type apply business rules when a business object instance exits a given state.
Monitor	Optional	Algorithms of this type monitor a business object instance while in a given state. Typically these are used to auto-transition it to another state.

The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object Inheritance for more information.

> **NOT_SUPPORTED**
>
> Refer to BO Algorithm Execution Order for a summary of when these lifecycle algorithms are executed with respect to BO level algorithms.

Copyright © 2026, Oracle and/or its affiliates.

- Entering A State
- Exiting A State
- Monitor Rules

Entering A State

You can define algorithms that execute before a business entity enters a given state. For example, you could develop an algorithm that requires a cancellation reason before an entity is allowed to enter the **Canceled** state.

You can also incorporate state auto-transitioning logic within this type of algorithms. Refer to auto-transition for more information.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for an enter algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

Exiting A State

You can define algorithms that execute when a business entity exists a given state. For example, you could develop an algorithm that clears out error messages when a given entity exits the **Error** state.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for an exit algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

Monitor Rules

You can define algorithms to monitor a business entity while it is in a given state. This type of logic is typically used to check if the conditions necessary to transition the entity to another state exist (and, if so, transition it). For example, transition an entity to the **Canceled** state if it's been in the **Error** state too long. Another common use is to perform ancillary work while an entity is in a given state. For example, update statistics held on the object while it's in the **Active** state.

Monitor algorithms are invoked when a business entity first enters a state and periodically after that in batch. You have the option to defer the monitoring of a specific state until a specific monitoring batch job runs. You do so by associating the state with a specific monitoring process. In this case the system will only execute the monitoring rules of this state when that specific batch process runs. This is useful when processing one type of record typically creates another type of record. You may want the processing of the second set of records to be deferred to a later time.

A monitor algorithm can carry out any business logic. In addition it can optionally tell the system to do either of the following:

- Stop monitoring and transition to another state. The system will not call any further monitoring algorithm plugged in on the state and attempt to transition the entity to the requested new state.
- Stop monitoring. Same as above except that no transition takes place. You may want to use this option to prevent transitions while some condition is true.

If none of the above is requested the system keeps executing subsequent monitoring algorithms.

Copyright © 2026, Oracle and/or its affiliates.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for a monitor algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

> **NOT_SUPPORTED**
>
> Refer to Business Object - Lifecycle for more information on how to set up state-specific algorithms.

Inheriting Lifecycle

If a business object references a parent business object, it always inherits its lifecycle from the highest-level business object in the hierarchy. In other words, only the highest-level parent business object can define the lifecycle and the valid state transitions for each state. Child business objects, in all levels, may still extend the business rules for a given state by introducing their own state-specific algorithms.

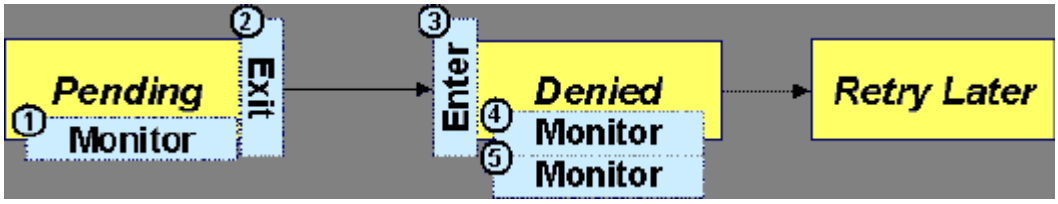
The system executes all state-specific algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

Auto-Transition

In a single transition from one state to another, the system first executes the **Exit** algorithms of the current state, transitions the entity to the new state, executes the **Enter** algorithms of the new state followed by its **Monitor** algorithms. At this point if a **Monitor** algorithm determines that the entity should be further automatically transitioned to another state the remaining monitoring algorithm defined for the current state are not executed and the system initiates yet another transition cycle.

Notice that an **Enter** algorithm can also tell the system to automatically transition the entity to another state. In this case the remaining **Enter** algorithm as well as all **Monitor** algorithms defined for the current state are not executed.

The following illustration provides an example of an auto-transition chain of events.



In this example a business entity is in a Pending state. While in that state a **Monitor** algorithm determines to auto-transition it to the Denied state. At this point the following takes place:

- No further **Monitor** algorithms of the Pending state are executed
- Pending state **Exit** algorithms are executed
- The system transitions the entity to the Denied state
- Denied state **Enter** algorithms are executed. No further auto-transition is requested.
- Denied state **Monitor** algorithms are executed. No further auto-transition is requested.

Copyright © 2026, Oracle and/or its affiliates.

Keeping An Entity In Its Last Successful State

By default, any error encountered while transitioning a business entity from one state to another rolls back all changes leaving the entity in its original state.

When applicable, the Maintenance Object can be configured to always keep an entity in its last successful state rather than rolling all the way back to the original state. This practice is often referred to as "taking save-points". In case of an error, the entity is rolled back to the last successfully entered state and the error is logged on the maintenance object's log. Notice that with this approach no error is returned to the calling process, the error is just logged.

The logic to properly log the error is in a **Transition Error** Maintenance Object plug-in. The system considers a maintenance object to practice "save-points" when such an algorithm is plugged into it.

Even if the maintenance object practices "save-points", in case of an error the system will not keep an entity in the last successfully entered state if that state is either marked as transitory or one of its **Enter** algorithms has determined that the entity should proceed to a next state. The system will roll back to the first previous state that does not match these conditions.

Monitoring Batch Processes

A monitor batch process may be used to transition a business object into its next state by executing the monitor algorithms associated with the current state of the entity. The use cases for performing the monitor logic in batch are as follows:

- The record may be waiting for something else to occur before transitioning. The monitor algorithm may be coded to determine if the condition is satisfied and initiate the transition then. For example perhaps when entering a state, a field activity is generated and the record should exit the state when the field activity is complete. The monitor algorithm can check the status of the field activity.
- Perhaps a record is added or updated manually and the next step in the BO lifecycle includes a large amount of processing such that the logic should occur in batch. In this case the BO status is configured with an explicit reference to a batch control (referred to as "deferred"), which indicates to the system that the monitor algorithms should not be performed automatically (but should be deferred to batch). Later when the batch process runs, it selects all the records to process to progress the records.

  > **Note**
  >
  > When a status includes a deferred batch control, it may also be configured to allow a user to manually transition the record to the next state rather than waiting for batch. When a user manually transitions a record that includes monitor algorithms, those algorithms are not executed.

- Perhaps a record is added or updated in batch, but a subsequent step in the overall lifecycle should be processed later. This may be accomplished by ensuring that the batch control linked to the state to process later does not match the batch control that added or updated the record.
- Monitor processes may also be used to periodically perform some logic related to the record without actually transitioning the record.

Note that only the parent business object may refer to a deferred monitor batch process. However, any business object in the "inheritance" chain may be configured with monitor algorithms, which will all be executed.

Copyright © 2026, Oracle and/or its affiliates.

The base package provides a periodic monitoring batch process for each maintenance object that supports a configurable BO lifecycle. The process periodically executes the monitoring algorithms associated with the current state of an entity, excluding states explicitly referencing a deferred monitoring batch process that is for a different batch control.

A deferred monitoring process selects records whose current state references this particular batch control as their monitor process. Deferred monitor process is only needed when an object has different use cases for monitoring the same type of records with different schedules. In this case only one periodic monitor batch should be configured. Other monitors should be configured to restrict by batch control so that there is no overlap in processing.

> **Note**
>
> **MO option configuration**. The maintenance object includes options to indicate the batch controls delivered for periodic and deferred monitor batch controls.

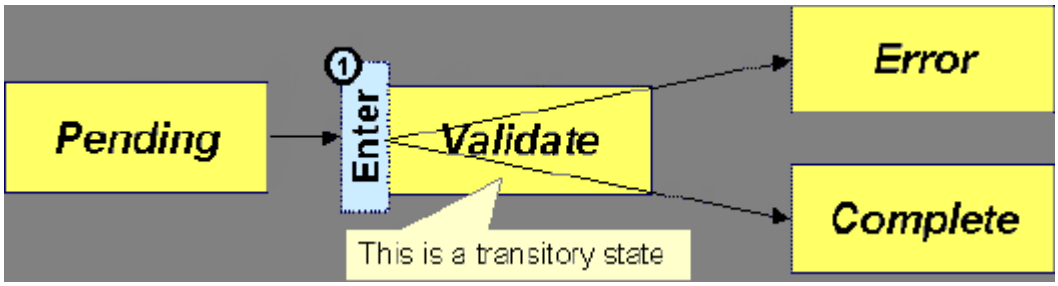
Your business rules will dictate the execution frequency of each monitoring process and the order in which they should be scheduled. Refer to Monitor Background Processes in the background process chapter for more information about the parameters supported for this type of batch process.

> **Note**
>
> **Updates to the business object.** When the monitor algorithms indicate that the business object should transition, the monitor batch processes are responsible for ensuring the business object is transitioned appropriately and that the appropriate exit, enter and monitor algorithms are executed. Please note that the business object is not updated using a call to the maintenance object service and therefore by default the business rules plugged in to the business object are not executed. However, it is possible for an Enter algorithm, Exit algorithm or Monitor algorithm to indicate that the other algorithms should be executed by the batch process. If the "force post processing" indicator is set to true, then the batch process invokes the BO Post Processing, BO Audit and MO Audit algorithms.

Transitory States

You can define a state as **Transitory** if you do not wish the business entity to ever exist in that particular state.

The following illustrates a lifecycle with a transitory Validate state.



In this example, the business entity is saved still not validated in the Pending state. At some point, the user is ready to submit the entity for validation and transitions it into a transitory state Validate whose **Enter** rules contain the validation logic. The responsibility of the transitory

Copyright © 2026, Oracle and/or its affiliates.

state's **Enter** algorithms is to decide if the entity is valid or in error and then transitions it into the appropriate final state. In this scenario, you may not ever want the business entity to exist in the Validate state.

Let's also assume that the maintenance object in this example is practicing "save-points" and requires the entity to be kept in its last successful state. If an error were to occur during the transition from **Validate** to the next state, the system would roll back the entity back to Pending, and not Validate even though the entity has successfully entered the Validate state. Refer to the Auto Transition section for more information.

State Transitions Are Audited

Most Maintenance Objects that support soft lifecycle definition also have a log to hold significant events throughout a business entity's lifecycle. For example, log entries are created to record:

- When the business entity is created (and who created it)
- When its status changes (and who changed it)
- If a transition error occurred (and the error message)
- References to other objects created throughout the entity's lifecycle. For example, if a To Do entry is created as part of processing the entity, the To Do Entry is referenced in the log.
- Manual entries added by a user (think of these as "diary" entries)

When a business entity is first created and when it transitions into a new state the system calls **Transition** algorithm(s) plugged in on the Maintenance Object to record these events. If the maintenance object supports a log these events can be captures as log entries.

> **Note**
>
> Most base package maintenance objects supporting a log may already provide state transition logging as part of their core logic. In this case you only need to provide a **Transition** plug-in if you wish to override base logging logic with your own.

Required Elements Before Entering A State

You can define additional elements that are required before a business entity can enter a given state. For example, let's assume that a Cancel Reason must be defined before an object can enter the *Canceled* state. You do this by indicating that element as a **Required Element** state-specific option on the appropriate state on the business object.

Capturing a Reason for Entering a State

Some business objects support configuring certain states to allow or require a status reason when an object enters the state. The product provides a centralized status reason table that may be used to define the valid BO status reasons for various business objects and various states. The status reasons are defined using the Status Reason portal.

The following sections provide additional information about the BO status reason functionality.

Copyright © 2026, Oracle and/or its affiliates.

Maintenance Object Must Support Status Reason

In order for a business object to use the centralized status reason table to define reasons, the maintenance object must first support the status reason. MOs that support status reason have the following characteristics:

- The primary table includes a column for Status Reason. This represents the status reason for the record's current status, if applicable.
- The log table includes a column for Status Reason. The standard logic for capturing a log record when entering a state also captures the status reason, if applicable. This allows a user to review the history of the changes in status and the status reason captured for a previous state transition, if applicable.
- The maintenance object option collection includes an option that defines the Status Reason field. This setting is a trigger for business objects of this MO to be able to configure states to allow or require status reason.

Business Object State Indicates if Reasons are Applicable

Once the MO is configured to support status reason, configuration on the business object is required to indicate the states where a reason is applicable. States may be configured to require status reasons, allow status reasons as optional or not allow status reasons. With this configuration, the framework will automatically get the list of valid reasons for a state that allows or requires them and then prompt the user for a status reason when a manual state transition occurs for that state. It also automatically triggers an error if the state requires a status reason and no reason is provided.

> **Note**
>
> The status reason configuration on the business object state is customizable. That means that for a product owned business object, an implementation may opt to change the delivered configuration.

Status reasons are defined for the parent (or "lifecycle") business object. All business objects in the hierarchy of the parent business object have the same valid reasons for their states.

The status reason code must be unique for the centralized status reason table. Business object and status are required fields, so it is not possible to share a common reason code (like "Not applicable") across multiple business objects or states. If multiple BOs / states want to support a reason "Not applicable" then each must define a unique record for it. This point should be considered when planning for your status reasons.

Override Description

The description of product owned status reasons may be overridden by your implementation.

Selectable vs. Not Selectable

When defining a status reason, you may indicate whether it's **Selectable** or **Not Selectable**. When a manual transition is performed and a user is prompted for a status reason, only the **Selectable** reasons are presented. The **Not Selectable** reasons may be defined to support transitions that occur via algorithm processing.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Selectable setting is customizable. That means that if a product provides a base owned status reason for a business object state, an implementation may opt to change whether it is selectable or not. Careful consideration should be made before changing a base delivered status reason from **Not Selectable** to **Selectable** as this may affect base provided algorithm functionality that could be relying on the setting of **Not Selectable**.

Status Reason Business Object

The status reason maintenance object, as with many maintenance objects in the product, references a business object used to define attributes and behavior related to defining status reasons. The framework provides a business object for status reason (`F1-BOStatusReason`). For the business objects that have states that require a status reason (let's call these "transactional BOs"), if there is some special logic required for defining the status reasons, it is possible to define a different status reason BO. In this situation, the override status reason BO to use for capturing status reasons should be defined as a BO option on the transactional BO using the **Status Reason Business Object** option type. If a transactional BO does not define any status reason BO option, then the `F1-BOStatusReason` is used when adding a status reason.

Defining a Usage

The base product status reason BO provides the ability to define a "usage" value. This is useful for algorithms that perform state transitions where a status reason is needed and where the algorithm is usable by more than one business object. In this case, the status reason to use cannot be provided as a parameter because each business object must define its own set of status reasons for each state. The Usage value can be used instead. Each business object can configure the status reason to use in the algorithm and set the appropriate usage value. The algorithm can reference the usage value and retrieve the correct status reason to use based on the record's transactional BO.

The status reason business object provided with the framework product (`F1-BOStatusReason`) supports capturing a usage. The valid usage values are defined in the **Status Reason Usage** characteristic type.

Alternatives for Defining Reasons

There may be business objects in the system that capture reasons that are defined somewhere besides the BO status reason table. For example, some objects may have an explicit administrative table for status reasons. Some objects may use a Lookup or an Extendable Lookup to capture reasons. Refer to the business object description for information about how valid reasons are defined, if applicable.

If a business object supports a reason that is not related to a state transition (such as a creation reason), the BO status reason would not be used. One of the alternate methods for defining a reason, described above, would be used.

Opening UI Maps Before Entering A State

You can define a UI Map to capture additional elements before a business entity can enter a given state. You do this by configuring the **State Transition UI Map** state-specific option on the appropriate state on the business object.

Copyright © 2026, Oracle and/or its affiliates.

There may be circumstances in which the status pre-processing logic can determine the value of the elements, in which case it is not necessary to invoke the map. The system provides a data area (`F1-StateTransitionCommon`) which contains a boolean element to indicate if the state transition map can be skipped. This data area can be included in your pre-processing script to allow the element to be set according to your business logic. The base maintenance script (`F1-MainProc`) references this element to determine whether to open the state transition map.

BO Algorithm Execution Summary

This table highlights the processing steps that occur when adding or changing a record that is governed by a business object.

Invoke BO	
Event	**Comments**
BO Pre-processing algorithms executed	These algorithms are only executed when Invoke BO is used. The business object in the Invoke BO is the one whose rules are executed.

MO Processing	
Event	**Comments**
Determine if status has changed.	The system keeps a note of the new status value but initially proceeds with the old value.
MO Processing.	Standard MO processing, including MO validation is executed.
Determine BO algorithm executed.	The MO level algorithm is executed to determine the identifying BO.
BO Validation algorithms executed.	
State transition rules are performed if the status has changed.	BO Status Exit algorithms for the "old" status executed. Status updated to the new value. BO Status Enter algorithms for the "new" status executed. If no error â€” MO Transition algorithms are executed.

NOT_SUPPORTED

Refer to State Transitions are Audited for more information.

Copyright © 2026, Oracle and/or its affiliates.

MO Processing	
	If error and there are "save points" the MO Transition Error algorithms are executed. **NOT_SUPPORTED** Refer to Keeping An Entity In Its Last Successful State for more information. Otherwise, the error is reported.
BO Status Monitor algorithms are executed.	If the record transitions again, the prior step (State transition rule step) is repeated for the new transition.
BO Post-processing algorithms are executed.	
BO Audit algorithms are executed.	These algorithms are only executed if the system detects a change in elements that are not marked with "no audit".

Note

To emphasize, the steps in the MO Processing table are only executed when the maintenance object service is invoked. Any add or update initiated by an "invoke BO" statement will invoke the MO service. This is also true for web service that invoke the business object. The Monitor Batch Process does not invoke the maintenance service. By default the monitor batch process only executes the monitor algorithms and the state transition rules (if the monitor algorithms indicate that a status change should occur). However, it is possible for an Enter algorithm, Exit algorithm or Monitor algorithm to indicate that the other algorithms should be executed by the batch process. If the "force post processing" indicator is set to true, then the batch process invokes the BO Post Processing, BO Audit and MO Audit algorithms.

Note

For records that do not have a status, the state transition rules and the monitor rules are not applicable.

Granting Access To Business Objects

Every business object must reference an application service. When you link a business object to an application service, you are granting all users in the group access to its instances. You can prevent users from transitioning a business object into specific states by correlating each business object status with each application service action (and then don't give the user group rights to the related action).

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

The system checks if a user has access rights each time the application is invoked to add, change, delete, read, or transition a business object. However, if a business object invokes another business object, we assume that access was controlled by the initial business object invocation and we do not check access for other business objects that it invokes. In other words, access rights are only checked for the initial business object invoked in a service call.

In order to apply business object security the system must be able to determine the business object associated with the actual object being processed. To do that the Maintenance Object itself has to have a **Determine BO** algorithm plugged in. If this algorithm is not plugged in or it cannot determine the BO on the MO, the system will not invoke any BO rules. If the business object cannot be determined for a maintenance object instance, business object security is not checked. In this case the system checks the user's access rights using standard maintenance object security.

Note

Parent business objects are ignored. If a child business object exists, a user need only have access to the child business object's application service (not to every application service in the business object hierarchy).

Defining Business Objects

This portal is used to view and maintain information about a business object.

You can access the portal from the **Admin Menu**, and then **System**, and then **Business Object**. You are brought to a query portal with options for searching for a specific business obejct. Once a business object has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Business Objects for more information.

Caution

Important! If you introduce a new business object, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Business Object.** This zone provides information about the selected business object.
- **Schema Editor.** This zone describes the business object’s schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the business object's schema using the Schema Designer tool. The purpose of the schema is to describe the business object's properties and map them to corresponding maintenance object fields.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

Tip

Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **Algorithms** tab page. Refer to A Business Object May Define Business Rules for more information.

- **Algorithms.** This zone is used to maintain the algorithms associated with the current business object.
- **Hierarchy Algorithms.** This zone displays all algorithms associated with the business object's inheritance hierarchy. The zone is visible when the current business object is part of a hierarchy and such algorithms exist.

The following zones may appear as part of the portal's **Options** tab page. Refer to A Business Object Defines Options for more information.

- **Options.** This zone is used to maintain various configuration options associated with the current business object.
- **Hierarchy Options.** This zone displays all options associated with the business object's inheritance hierarchy. The zone is visible when the current business object is part of a hierarchy and such options exist.

The following zones may appear as part of the portal's **Lifecycle** tab page. This tab is only available when the maintenance object supports lifecycle definition.

- **Lifecycle.** This zone describes and maintains the current business object's lifecycle definition. Broadcast a specific status to review its complete configuration.
- **Status Details.** This zone is used to maintain basic definition information for a selected status within the current business object's lifecycle.
- **Status Algorithms.** This zone is used to maintain algorithms for a selected status within the current business object's lifecycle.
- **Status Options.** This zone is used to maintain various configuration options for a selected status within the current business object's lifecycle.
- **Hierarchy Status Algorithms.** This zone displays all algorithms associated with the business object's inheritance hierarchy for a selected status. The zone is visible when the current business object is part of a hierarchy and such algorithms exist.
- **Hierarchy Status Options.** This zone displays all options associated with the business object's inheritance hierarchy for a selected status. The zone is visible when the current business object is part of a hierarchy and such options exist.
- **Status Reasons.** This zone is used to maintain reasons for a selected status. The zone is visible when the maintenance object supports a reason field and usage of reasons is allowed by the status definition.

The following zones may appear as part of the portal's **Lifecycle View** tab page. This tab is only available when the maintenance object supports lifecycle definition.

Copyright © 2026, Oracle and/or its affiliates.

- **Lifecycle View.** This zone provides a schematic view of the current business object's lifecycle definition. Broadcast a specific status to review its complete configuration.
- **Status Details.** This zone is used to maintain basic definition information for a selected status within the current business object's lifecycle.
- **Status Algorithms.** This zone is used to maintain algorithms for a selected status within the current business object's lifecycle.
- **Status Options.** This zone is used to maintain various configuration options for a selected status within the current business object's lifecycle.

The following zones may appear as part of the portal's **Hierarchy** tab page. This tab is only available when the current business object is part of a hierarchy.

- **Business Object Hierarchy.** This zone lists the inheritance hierarchy of business objects up to and including the current business objects. The zone is visible if the current business object is not the top level business object of the hierarchy.
- **Business Object Tree.** This zone describes the entire hierarchy of the current business object.
- **Hierarchy Algorithms.** This zone lists all algorithms defined for the current business object. The zone is visible if algorithm information exists for the current business object.
- **Hierarchy Lifecycle Algorithms.** This zone lists all algorithms associated with the current business object's inheritance hierarchy. The zone is visible if the current business object is not the top level business object of the hierarchy and algorithm information exits. Refer to Inheriting Lifecycle for more information.
- **Hierarchy Options.** This zone lists all configuration options for the current business object's lifecycle. The zone is visible if options information exists for the current business object.
- **Hierarchy Lifecycle Options.** This zone displays all options associated with the current business object's inheritance hierarchy. The zone is visible if the current business object is not the top level business object of the hierarchy and options information exits. Refer to Inheriting Lifecycle for more information.

The following zones may appear as part of the portal's **References** tab page.

- **Business Object References.** This zone list various configuration entities referencing the current business object or its schema.

Actions

The page supports the following special actions:

- **Generate User Interfact Configuration Artifacts.**When creating a new buisiness object or editing an existing one, you may request to generate the user interface configuration artifacts needed to properly maintain and display the business object on an existing portal. For example, if a display script is needed to further prepare the business object's details for presentation then request to generate such script. You can then navigate to the script from the **User Interface Configuration** section and adjust it as needed.
- **Subclass a Business Object.** Use the **Subclass** action to create a new business object that is a child of an existing business object. The action generates the new business object's schema as inherited from the existing business object. Refer to Business Object Inheritance for more information.
  Once created, you may extend and adjust the new business object's schema and rules as needed.

Copyright © 2026, Oracle and/or its affiliates.

- **Create an Algorithm.** Use this action, on the Algorithm tab or the Lifecycle tab for a specific status, to creates a script based algorithm related to this business object. You are prompted for information regarding the plug-in spot (BO or BO Status) and the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the business object. You may then navigate to the script to adjust it as needed.
- **Inactivate / Activate Base Algorithms.** You can inactivate algorithms on base Business Objects. Your implementation may want to use a business object provided by the base product, but may want to inactivate one or more algorithms provided by the base business object. To do that, use the "Inactivate" action on the appropriate algorithm entry on the Algorithm tab. Use use the "Activate" action to mark an inactive algorithm as active. Note that these actions add and remove the "Inactive Algorithm" businesss option as needed.
- **View Schema.** Click the hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

Advanced BO Tips and Techniques

The topics in this section describe some advanced tips and techniques for configuring business objects.

- Creating a Physical Business Object
- Managing To Do Entries
- Submitting a Batch Job

Creating a Physical Business Object

The physical business object's schema represents the complete physical structure of the maintenance object, and includes elements for all fields in the maintenance object's tables. There are several processes in the system that may use a maintenance object's physical BO.

To create a physical business object for the maintenance object:

1. Navigate to Business Objectand add a business object for the maintenance object. Do not fill in any of the UI oriented configuration.
2. The business object is created with a schema that looks like the physical structure of the maintenance object.
3. Save the physical business object.
4. Navigate to Maintenance Object and search for the maintenance object.
5. On the **Option** tab, add a new option with the type **Physical Business Object**. The value is the name of the physical business you just created.

Managing To Do Entries

The product provides several base algorithm types that may be used to manage To Do entries through status changes for a given record via BO lifecycle plug-ins.

Create To Do Entry

The product supplies a BO status Enter algorithm type **Generic To Do Creation** (`F1-TDCREATE`) that creates a To Do entry based on parameter configuration. Refer to the algorithm type description for more information about how it determines the To Do type or To

Copyright © 2026, Oracle and/or its affiliates.

Do role and how to populate the appropriate message text onto the To Do. This algorithm may be used in conjunction with the Retry Logic (below).

If your implementation has a business rule that requires a To Do entry to be created when entering a given BO status and the logic provided by the algorithm type meets the needs of the business rule, this algorithm type may be used. Create an algorithm for the algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as an Enter algorithm.

Retry Logic

The algorithm type **Retry for To Dos** (`F1-TODORETRY`) is supplied for a special use case. It is a BO status monitor plug-in and may be used for a state that is a type of ‘error’ or ‘waiting’ state. It relies on the To Do entry creation logic to set a Retry Frequency. The algorithm transitions to the originating state to retry the logic. The idea is that the condition that caused the record to enter the ‘error’ or ‘waiting’ state may be resolved after some period of time has passed, allowing the record to progress in its lifecycle. Refer to the algorithm type description for more information about its logic.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as a Monitor algorithm. The state should also have an algorithm for the **Generic To Do Creation** algorithm type plugged in as an Enter algorithm (or something equivalent) that sets the appropriate Retry Frequency.

To Do Completion

It is common that one or more To Do entries associate with a given record should be completed when exiting a state (if it is not already completed). The system supplies the algorithm type **Generic To Do Completion** (`F1-TODOCOMPL`) that may be used for this purpose. Note that the algorithm type functionality is not tied to any To Do creation logic. It may be used for any use case where To Do entries should be completed on exiting a state. Refer to the algorithm type description for more information about its functionality and how to prevent certain To Do entries from being automatically completed.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as an Exit algorithm.

Submitting a Batch Job

The product provides a base algorithm type that submits a batch job when entering a BO state. This functionality allows for “event driven” batch submission where the event is the lifecycle transition for a certain record.

The algorithm type is **Create Batch Job Submission Entry for Batch Control** (`F1-SCHEDJOB`). The batch control code is a parameter for the algorithm. Refer to the algorithm type description for more information about its logic.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameter with the batch control that should be submitted and plug the new algorithm into the appropriate business object status as an Enter algorithm.

Defining Status Reasons

Status Reasons are used to provide more information about why a business object transitioned to a given state. The status reason table provides a centralized place where status reasons can be defined across many different business objects and states.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Defining Reasons for Entering a State for overview information.

If a business object has one or more states that are configured to capture a status reason, you may configure the valid reasons by navigating to the status reason portal using **Admin**, and then **System**, and then **Status Reason**.

The topics in this section describe the base-package zones that appear on the Status Reason portal.

Business Objects with Status Reason List

This zone displays the business objects that have one or more status values that allow status reasons to be defined.

Click the broadcast icon to open other zones that contain more information about the business object's status reasons.

Status Reasons

The **Status Reasons** zone contains a list of the existing status reasons for the broadcasted business object.

Data Areas

The data area has no business purpose other than to provide a common schema location for re-used schema structures. It exists solely to help eliminate redundant element declaration. For example, if you have multiple schemas that share a common structure, you can set up a stand-alone data area schema for the common elements and then include it in each of the other schemas.

Be aware that a stand-alone data area can hold elements that are mapped to true fields. For example, you might have 50 different types of field activities and all might share a common set of elements to identify where and when the activity should take place. It would be wise to declare the elements that are common for all in a stand-alone data area and then include it in the 50 field activity business objects.

It's strongly recommended that you take advantage of stand-alone data areas to avoid redundant data definition!

> **Caution**
>
> **Dynamic inclusion!** When the system renders a schema, all schemas included within it are expanded real-time. This means that any change you make to a data area will take effect immediately within all schemas it is referenced within. Therefore, changing a schema that is referenced in a large number of schemas may be too large to be performed online. In this situation, the change should be performed in batch. Refer to "Update Object Schema (F1-SCMUP)" batch control for more information.

This section describes additional topics related to designing and working with data areas.

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Extension

You may use a custom data to extend a base product data area. To extend another data area, reference that data area in the **Extended Data Area** field. By extending a data area you can add additional elements to a base product data area.

Here's an example of an extended data area:

- The product releases with data area A, which contains elements a, b, and c.
- Your implementation creates data area CM-A, which contains element z, and references data area A as the extended data area.
- At run time, everywhere data area A is included it will contain elements a, b, c, and z.
- Defining Data Areas

Defining Data Areas

Use this page to define basic information about a data area.

You can access the portal from the **Admin Menu**, and then **System**, and then **Data Area**. You are brought to a query portal with options for searching for a specific data area. Once a data area has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new data area, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Data Area.** This zone provides information about the selected data area.
- **Extending Data Areas.** This zone lists all data areas extending the current data area. The zone appears if such data areas exist.
- **Schema Editor.** This zone describes the data area's schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the data area's schema using the Schema Designer tool. The purpose of the schema is to describe the data area's properties and map them to the underlying internal service fields .

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

Copyright © 2026, Oracle and/or its affiliates.

> **Tip**
>
> Click the **View Schema** hyperlink to view the data area's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **Data Area References.** This zone list various configuration entities referencing the current data area or its schema.

Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the data areas's expanded schema definition. Doing this opens the schema viewer window.

Business Services

A business service is used to expose a back-end service so that it may be invoked by a script or a zone or a map to retrieve information or perform functions, depending on the related service.

As with the business object, the business service's interface to the internal service is defined using its schema. The schema maps the business service's elements to the corresponding elements in the internal service program's XML document. Just as a business object can simplify the schema of its maintenance object by only defining elements that it needs and "flattening" entries in a child collection to be defined as a singular element, a business service schema may simplify its service XML in a similar way.

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

Inbound web services and scripts support interaction with business services. You can also invoke a business service from a Java class.

- Defining Service Programs
- Defining Business Services

Defining Service Programs

This transaction defines services available in the system. These include user interface services as well as stand-alone services that perform a specific function. A service may be referenced by a business service. Use this transaction to view existing service and introduce a new stand-alone service to be made available to a Business Service.

You can access the portal from the **Admin Menu**, and then **System**, and then **Service Program**. You are brought to a query portal with options for searching for a specific service program. Once a service program has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new service programs, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Service Program.** This zone provides information about the selected service program.
- **Service Program References.** This zone list various configuration entities referencing the current service program.

Application Service is the application service that is used to provide security for the service. If the service is related to a maintenance object, the access modes for the application service should be the standard **Add**, **Change**, **Delete** and **Inquire**. For other services, the application service should have the **Execute** access mode.

Service Type indicates whether the service is a **Java Based Service** or a **Java (Converted) Service**. Note that services generated to support a portal in the system will not have a service type populated.

The **Programs** grid shows the list of program user interface components associated with the service. For a stand-alone service, this list is typically not applicable.

Defining Business Services

Use this page to define basic information about a Business Service.

You can access the portal from the **Admin Menu**, then **System**, then **Business Service**. You are brought to a query portal with options for searching for a specific business service. Once a business service has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **⚠ Caution**
>
> Important! If you introduce a new business service, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Business Service.** This zone provides information about the selected business service.
- **Schema Editor.** This zone describes the business service's schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the business service's schema using the Schema Designer tool. The purpose of the schema is to describe the business service's properties and map them to the underlying internal service fields.

> **ⓘ Note**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

Copyright © 2026, Oracle and/or its affiliates.

> **Tip**
>
> Click the **View Schema** hyperlink to view the business service's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **Business Service References.** This zone list various configuration entities referencing the current business service or its schema.

Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the business service's expanded schema definition. Doing this opens the schema viewer window.
- **Test.** Use this action to test your business service. Enter input values and save. The service is performed and output information is presented, but the system issues a roll-back so that the changes are not committed. Note that if your service uses the save-point dispatcher to perform any actions or if it issues a call to an external system, the roll-back does not apply to those actions. If the service issues an error the corresponding message is displayed. Note that the input and output testing user interface is dynamically rendered based on UIHints embedded in the business service's schema. As such, the user interface may be as elaborate or basic as the UIHints in that schema.

> **Note**
>
> This action is secured by a designated "Test a Service" application service and the "Generic Service Executor" application service and also requires security rights for creating a script, which is the alternative way for testing any service. It is recommended to allow access to this function in non-production environments only.

- Useful Services and Business Services

Useful Services and Business Services

The following section highlights some business services and services provided by the product that may be useful for implementations to use.

Data Explorer Service

The system provides a mechanism for performing an SQL select statement for use in scripting, Java plug-ins, or via a web service call. This is done by creating a zone using one of the data explorer zone types where the SQL is defined. Then, create a business service using the Data Explorer service (**FWLZDEXP**).

> **Note**
>
> There are numerous business services delivered with the base product that reference this service that may be used as a template.

Copyright © 2026, Oracle and/or its affiliates.

The following points highlight how to create your own business service for this service. Note that typically a separate business service exists for each zone.

- Enter a **Business Service** code and a **Description**. It is recommended to define the business service code to match the zone code so that it's easier to manage which business service invokes which zone.
- Select the **Service Name FWLZDEXP**.
- On the **Schema** tab, under the ***<schema>*** node, enter mapping for the fields that are required for the Data Explorer service:
  - The **Zone** should be mapped into service field ***ZONE_CD*** . Define the zone code as the default value.
  - For every **user filter** defined on the zone, create a schema mapping into the service field ***Fx_VALUE***, where "x" is the filter number (from the zone parameters).
  - For every **hidden filter** defined on the zone, create a mapping into the service field ***Hx_VALUE***, where "x" is the filter number (from the zone parameters).
  - The search results are returned as a list by the data explorer service. Each column value is in the service field ***COL_VALUE*** with an appropriate sequence number (***SEQNO***). The results can be flattened based on sequence number allowing for a logical element name to be defined.
  - Another useful field is ***ROW_CNT***, which provides the number of rows retrieved by your search.

The following is an example of the schema for a BS that receives a business object code and returns a list of status values and their descriptions that allow status reasons to be defined.

```
<schema>
     <zone mapField="ZONE_CD" default="F1-BOSTSLST"/>
     <bo mapField="H1_VALUE" />>
     <rowCount mapField="ROW_CNT"/>>
     <results type="list" mapList="DE">>
          <status dataType="string" mapField="COL_VALUE">
               <row mapList="DE_VAL">>
                    <SEQNO is="1" />>
               </row>>
          </status>>
          <description dataType="string" mapField="COL_VALUE">
               <row mapList="DE_VAL">>
                    <SEQNO is="2" />>
               </row>>
          </description>>
      </results>>
</schema>
```

Maintenance Object Log Service

Many maintenance objects support a log table that follows a pattern of column names and behavior. The system provides a service called Generic MO Log Service (F1MOLOGP) that may be used to perform common functions related to log entries:

- Read log entries. If you pass a certain MO, primary key and log sequence number, the service will return the details of that log entry. The product provides a generic business service that may be used for this purpose — Generic MO - Retrieve Log Details (F1–

Copyright © 2026, Oracle and/or its affiliates.

ReadMOLog). Alternatively, it is possible to create a business service for a given MO where the MO code is assigned to the MO element using the default syntax. This allows business functionality specific to that maintenance object to use the specific BS.

- Add log. The service may be used to add a log entry. If a user log is added, then the comments from the user are populated in the detailed description. System generated log entries typically supply a message category / message number along with other information such as the status, a specific log type and optionally a related object reference (via a characteristic). The product provides a generic business service that may be used for this purpose — Add Generic MO Log (F1–AddMOLog). Alternatively, it is possible to create a business service for a given MO where the MO code is assigned to the MO element using the default syntax. This allows business functionality specific to that maintenance object to use the specific BS.

Base Business Services

The following table highlights some business services provided by the product that may be useful for custom logic for an implementation.

> ⚠ **Caution**
>
> This is not intended to be a complete reference of Business Services. Refer to the business service page to find all the supported business services.

Business Object Related Services

Business Service Name	Description
F1-AutoTransitionBO	Performs monitoring algorithms associated with the current state of a given business object instance (which may result in subsequent state transitioning).
F1-CompareBusinessObjectData	Compares two versions of a given business object instance.
F1-DetermineBo	Determines the business object of a given instance of a maintenance object by executing the MO's Determine BO logic.
F1-GetRequiredFieldsForBOState	Returns the required fields for a given business object status.
F1-RetrieveBOOption	Returns BO option values for a given BO and option type.
F1-RetrieveBOStatusOption	Returns BO option values for a given BO, status and option type.
F1-RetrieveBOStatusOption	Retrieves a list of BOs for a given MO that are accessible for the current user.
F1-RetrieveBoStatusDescription	Return the description of a given BO status.
F1-RetrieveBusinessObjectLabel	Return the label appropriate for a given path (e.g. element) within a BO schema.
F1-RetrieveNextStates	Return a list of next possible states based on the input of a MO and its prime key, or a BO and one of its statuses.

Email Related Services

Copyright © 2026, Oracle and/or its affiliates.

Business Service Name	Description
F1-EmailService	Sends an email message in real time.
F1-RetrieveEmailAddress	Retrieves the email addresses of users belonging to a To Do Role.
F1-RetrieveEnvironmentURL	Retrieves the current environment URL information for the installation.

Tools for Maps and Scripting

Business Service Name	Description
F1-AddWarning	This business service may be used to log a warning by providing the message details. The system gathers all the warning and displays them to the user all at once.
F1-CheckOnlineConnection	Check if Connection is Online or Batch.
F1-CheckWarning	Used to check if any warnings have been accumulated. If warnings have been accumulated, any logic that cannot be rolled back (when the warnings are shown) should not be performed.
F1-DateMath	Performs various date and time math calculations. Refer to the BS description for more details.
F1-DateTimeFormattingService	Formats a given date / time based on the user's display profile settings.
F1-ExecuteScriptInNewSession	Executes a Service Script in a new processing session/transaction.
F1-GetFieldLabel	Retrieves the label for a given field.
F1-GetForeignKeyReference	Returns foreign key reference information for a given FK Reference and primary key, including info description, navigation option, and context menu.
F1-GetFKReferenceDetails	Returns foreign key reference information for a given MO and primary key, including FK reference code, info description, navigation option, search zone and context menu.
F1-GetLookupDescription	Returns lookup description for a lookup field value given the lookup field name.
F1-GetExtLookUpVal	Returns the list of values for a given extendable lookup BO.
F1-GetMonthInYearAbbreviation	Returns a 3-character month abbreviation for an input date in system format.
F1-NumberAmountFormatter	Formats a given amount or number based on the user's display profile settings. It also may receive input to adjust the scale and optionally apply currency settings.
F1-OutmsgDispatcher	Dispatches a real-time message giving the user the option of whether to persist the message on the database, and whether to trap errors that may take place during the call. Refer to Real Time Messages for more information.

Copyright © 2026, Oracle and/or its affiliates.

Business Service Name	Description
F1-OutmsgMediator	Alternative to F1-OutmsgDispatcher and may be a better option if the sender does not require an outbound message record to be instantiated. Refer to Real Time Messages for more information.
F1-RethrowError	Issues an application error using the input message category / number / parameters.
F1-RetrieveMODescription	Retrieves the description for a maintenance object.
F1-ReturnMessage	Returns the expanded message given a message category, number, parameters, and parameter types.
F1-SavePointDispatcher	Allows for a service script to be executed where exceptions are trapped and the transaction is rolled back to a save point set before the service script execution.

User Related Services

Business Service Name	Comments
F1-CheckApplicationSecurity	Checks a user's security for a given application service / access mode
F1-CheckUserAuthorization	Determine whether a given user is authorized for access based on the input application service, security code, and authorization level.
F1-DetermineIfUserCanApproveTD	Determine if the current user can approve a given To Do.

User Interface (UI) Options

Your implementation may create input forms and output maps that closely match your customer's business practice within the following user interface areas:

- Content presented in portal zones. For example, a map that displays the details of a business object or any other informational content. Refer to Map Zones for more information.
- An input form presented via Business Process Assistant (BPA) scripts for data entry and update purposes. For example, a form that captures business object information or prompts the user for additional information related to a task they perform.
- An input form presented as a panel within a process flow.

The HTML for these forms and maps may be provided using the following options:

- **UI Map.** Compose your HTML and store it as a UI map, which is a repository for an HTML document. The UI Map HTML gives you the ability to craft the display by any method that an HTML document can support, including JavaScript and full CSS functionality.
- **UI Hints.** With this technique you do not compose the HTML. Instead, the HTML is generated at run time based on a schema that describes the data being displayed and embeds user interface rendering instructions known as "UI Hints".
- Understanding UI Maps

Copyright © 2026, Oracle and/or its affiliates.

- Understanding UI Hints
- UI Hint Syntax
- Defining UI Maps

Understanding UI Maps

The preferred method for creating input and output HTML maps is by leveraging the schema-based HTML rendering technique known as UI Hints. For more complex requirements, you may craft an HTML that closely matches your customer's business practices and capture it as a User Interface (UI) map.

The User Interface (UI) map is a repository for a single HTML document paired with an XML schema where the schema defines the data that the HTML document displays and/or modifies. The UI Map HTML gives you the ability to craft the display by any method that an html document can support, including JavaScript and full CSS functionality.

The topics in this section describe concepts and guidance relevant to UI map configuration.

Output Maps and Input Forms

UI maps are typically crafted as output tables when used in conjunction with portal zones - please refer to Map Zones for more information. When referenced within BPA scripts or process flows, UI maps are typically crafted as forms for the capture and update of data.

Figure HTML Input Form Rendered (for BPA Script)



Copyright © 2026, Oracle and/or its affiliates.

Figure HTML to Display Customer Business Object

```
<html>
<head>
<title>Output Personal Information</title>
<link rel="stylesheet" type="text/css" href="cm_templates/cmStyles.css"/>
</head>
<body>

<table width="100%" border="0" cellpadding="2" style="margin-top:15px;" >

    <tr>
        <td/>
         <td/> <!-- locate the edit button on the bottom of the third column -->
        <td rowspan="99" style="vertical-align:bottom; margin-left:5px">
            <input type="button" value="edit" onClick="oraRunScript('HumanInfoU','personId');"/>
        </td>
    </tr>

    <tr>
        <td class="outputLabel">Name:</td>
        <td><span oraField="name" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Home Phone:</td>
        <td><span oraField="homePhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Business Phone:</td>
        <td><span oraField="businessPhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Cell Phone:</td>
        <td><span oraField="cellPhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">FAX:</td>
        <td><span oraField="fax" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Social Security:</td>
        <td><span oraField="socialSecurity" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Drivers License:</td>
        <td><span oraField="driversLicense" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Email:</td>
        <td><span oraField="email" class="outputData"></span></td>
    </tr>

</table>

</body>

<xml>
<root>
    <name>Greer,Johan</name>
    <email>jurgen.greer@media.com</email>
    <socialSecurity>939-30-3939</socialSecurity>
    <driversLicense>C8392020</driversLicense>
    <homePhone>(838) 030-0303</homePhone>
    <cellPhone>(444) 444-4040</cellPhone>
    <businessPhone>(737) 393-3838</businessPhone>
    <fax>(373) 939-3939</fax>
    <personId>1239997654</personId>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

Figure Customer HTML Rendered (Output Data for Zone)



UI Map Fragments

A UI map may represent a complete HTML map or a fragment that can be embedded in a complete map.

You may use a UI map fragment in the following use cases:

- To share and reuse the same HTML/JavaScript across multiple complete UI maps. The UI map fragment is referenced as part of the HTML definition of the complete UI maps.
- To embed complex HTML/JavaScript logic as part of a map that its HTML is otherwise rendered using UI Hints.
- To implement a filter map for a data explorer search zone where basic filter definition syntax does not handle your business requirements.

Configuration Tool Support

Configuration tool support for UI Maps hinges around the ability to inject and extract an XML document from the HTML. For more information on the specialized support for HTML and JavaScript functionality refer to UI Map Attributes and Functions.

- Ensuring Unique Element IDs for UI Maps

Ensuring Unique Element IDs for UI Maps

The following describes how to modify JavaScript code to ensure the proper rendering of unique element IDs for UI Maps.

The modification is required only for code that renders HTML using a getElementById() (or similar) function to generate list IDs and avoid account verification or related errors.

The following sample snippet contains the necessary modifications:

```
...
function getElementsFromList(namePrefix) {
    var ret = [];
    var elements = document.getElementsByTagName("INPUT");
    for(var i=0;i<elements.length;i++) {
        var elemID = elements[i].id;
        if((id) && (id.startsWith(namePrefix + '_')) {
            ret.push(elements[i]);
```


Copyright © 2026, Oracle and/or its affiliates.

```
        }
    }
...
return ret;
```

Since IDs aren't necessarily unique in generated UI Map IDs, the code shown above ensures uniqueness at runtime by appending an underscore and row number (e.g., myField_1, myField_2) for proper handling by Framework in the rendered HTML, while still allowing you to reference the unmodified IDs contained in the generated UI Map.

A switch in the spl.properties file also permits you to disable the generation of unique IDs for elements in a grid (as described below), though, for standards compliance reasons, it is highly recommended that this switch be left at is default value.

```
Property Name: spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs
File Name: spl.properties (under web project in FW)
Default Value: false
Accepted Values: true or false
Description: This property controls the generation of unique IDs for all
input elements inside
a list. When this value is set to true it disables the generation of unique
IDs, thus
replicating the old behavior. When this property is set to false or this
property is missing
it enables the generation of unique IDs, thus enabling the list to be
standards-compliant.
```

Understanding UI Hints

A schema may be used to describe the structure of a business object or any other XML document. When applicable, the same schema may also embed user interface (UI) "hints" that are interpreted at run time and render the HTML needed to display or maintain that document. Refer to UI Hint Syntax for more information.

The main advantages of this technique are as follows:

- **Less error prone.** A tighter correlation between the data structure and its presentation layer keeps the latter up to date, for example:
  - Adding an element to the schema or removing an element from it result in a corresponding change to the presentation layer. Note that you can explicitly exclude elements from the presentation layer as needed.
  - The HTML widget that is most appropriate to present or input data for an element is determined by its data type as defined by the schema. For example, a date picker is used for an input date element. If the element's data type changes in the schema its presentation adjusts accordingly.
- **Faster to implement.** No HTML skills are needed. The schema definition along with its embedded UI hint instructions is used to render the HTML at run time. Eliminating the need of crafting HTML and maintaining it as a UI Map reduces implementation and maintenance costs.
- **Supports business object inheritance.** Incorporating the presentation layer into the business object schema allows each business object in a hierarchy to contribute their part to the complete HTML that displays the entity. In other words, the complete HTML used to

Copyright © 2026, Oracle and/or its affiliates.

display a subclassed business object is made of the HTML rendered by the included schema of its parent business object, followed by elements extended by the subclass.

The topics in this section describe concepts and guidance relevant to the UI Hints functionality.

Display vs. Input Modes

The same schema may be used to generate either a display version of its HTML map or an input map. In the display version, element values are presented as read-only, corresponding descriptions are shown instead of codes, and additional artifacts such as links are available for navigation to related pages. In the input version, the element is editable, using the appropriate HTML widget for its data type. For example, a foreign key element is rendered along with search components, and a date element incudes a date picker widget.

The type of map generated depends on the context in which it is used:

- The display map is generated when the schema is used to present content in a map zone.
- The input map is generated when the schema is used for data entry as part of a BPA script step or a process flow panel. Refer to Generate Edit Map Statements and process flows for more information.

By default, each element in the schema is assumed to be included in both the display and input presentation maps. UI hints allow an element to be excluded from presentation altogether, included only in the display map or included only in the input map.

Not Just Business Object Schemas

While mostly used to describe the display and maintenance maps of business objects, UI Hints may also be embedded as part of other schema based objects:

- You may use a data area to describe an input form and prompt the user to fill it in using a BPA script step or a process flow panel.
- You may embed UI hints in a business service or service script schema for the purpose of displaying the content prepared by the service in a portal zone. Refer to map zones for more information.
- You may embed UI hints in a service script's schema to prompt for user input before calling the script to perform a task and presenting results. Refer to Generate Edit Map Statements for more information.

Definition Order is Presentation Order

Elements are presented in the order they are defined in the expanded schema. For example, a subclassed business object that includes its parent business object followed by extension elements is presented with the parent portion of the HTML, followed by the extension fields.

Use UI Map Fragments for Complex HTML

The following items describe basic capabilities supported by the UI Hints functionality:

- Control which elements are presented in the display and input maps.
- Provide a title for the input map.
- Introduce sections around elements by injecting start and end section instructions. The order in which the elements are defined within a section is also their presentation order.
- Basic visibility rules are supported at the element and section level.

For more complex requirements around a specific element, you may craft the necessary HTML/JavaScript for the element, capture it as a UI Map Fragment, embed the fragment as

Copyright © 2026, Oracle and/or its affiliates.

part of the schema in the appropriate presentation order, and suppress the element from basic rendering for the corresponding input or display map.

> **Note**
>
> You may also use this technique to display elements in a different presentation location. For example, when a business object's display script needs to display additional elements that are not part of the business object's schema. In this situation, the script's schema introduces the elements as suppressed and the business object schema includes an embedded HTML UI map fragment to display them in the appropriate section. Note that this may only work when both the business object and script are owned by your organization.

Refer to UI Hint Syntax for more information.

UI Hint Syntax

Working Examples

For working examples of uiHint functionality, refer to the following business objects:

BOs with User Assigned Keys

The following examples illustrate the patterns used to enable uiHints on an object with a user specified key.

- `F1-OutcomeStyleLookup`. This extendable lookup BO does not require state transition, but does allow duplicate and delete actions.
- `F1-TodoSumEmailTyp`. This request type illustrates the hints required to support state transition on a display map.
- `F1-WebSvc`. This web service BO is a good example for management of complex JavaScript requirements. Both display and input maps have functionality that requires specialized javascript.

BO with System Generated Key

The following example illustrates the pattern used to enable uiHints on an object with a system generated key.

- `F1-GenericAttachment`. This attachment BO has a system assigned key, which entails the following special handling:
  - `F1-AttachmentMain`. This is the main section data area contains the elements common to all attachments, including the key, bo, and version. Because this data area is used to define the main section of the generated maps, the main section of the map can be extended by an implementation via data area extension functionality.
  - `F1-AttachmentActions`. This record actions map contains the standard actions, Edit and Delete, plus custom actions used only by attachments, View and Upload.
  - `F1-AttachmentIDFrag`. This record information map contains the primary key of the attachment.

Display Map Service Script

Display map service scripts can be fully supported via dynamic HTML generation. However, to help eliminate the need for a display service script, self-contained uiHint functionality has been

Copyright © 2026, Oracle and/or its affiliates.

developed to write the business object status and determine valid state transitions. So the two most common reasons to craft a display service script have been eliminated.

A typical reason to use a display pre-script is if you have an embedded map fragment that contains a business service schema. The display service script can be used to invoke the business service. Both the map fragment and the display service script must declare the business service schema to support this scenario.

> **Warning**
>
> The zone used to display the object's map must have a derivation script, like `F1-GncDsMpDZ` or `F1-GenDss`, that will invoke a display service script for the business object if it has been defined as a BO option - but not require an explicit display map BO option. In addition, the display service script's schema must be enabled for uiHint functionality - as the script's schema will be dynamically rendered by the zone - and not the BO schema.

- `F1-ExcelSpreadsheet`. This attachment BO has a display service script used to manipulate the attachment business object before displaying it:
- `F1-AttchDtlU`. This display map service script's schema has been defined with the uiHint namespace, and will have a display map generated for it.

Maintenance Pre-Processing Service Script

Maintenance pre-processing service scripts can be used with uiHints.

- `F1-ExcelSpreadsheet`. This attachment BO has a maintenance pre-processing service script used to manipulate the attachment business object before rendering the maintenance map:
- `F1-AttchPre`. This pre-processing service script's schema mimics a maintenance map schema with embedded boGroup and action elements. It will be invoked before the maintenance map is rendered.

Maintenance Post-Processing Service Script

Maintenance post-processing service scripts can be used with uiHints.

- `F1-ExcelSpreadsheet`. This attachment BO has a maintenance post-processing service script used to manipulate the attachment business object after rendering the maintenance map:
- `F1-AttchPost`. This post-processing service script's schema mimics a maintenance map schema with embedded boGroup and action elements. It will be invoked after the maintenance map is rendered.

Technical Notes

The following prerequisites are required to support dynamic HTML generation:

Schema Requirements

To support automated UI generation, the business object schema must contain the following:

- <schema **xmlns:uiHint="http://oracle.com/ouafUIHints"**>. The schema node must name the uiHint namespace.
- **isPrimeKey="true".** Every element of the business object schema that is part of the primary key must be identified.

Copyright © 2026, Oracle and/or its affiliates.

Maintenance Script Requirements

The maintenance script for the MO must be enabled for dynamic generation.

> **⚠ Caution**
>
> The business object maintenance BPA script must be declared as an MO Option for uiHint maintenance functionality to work!

If the script performs `F1-BOProc` then it is likely no special functionality is needed. However, if the maintenance script contains its own call to `F1-GetValOpt` then the following statement is required prior to that call:

```
move 'false' to "F1-GetBOOpts/input/maintenanceMapRequired";
performScript 'F1-GetValOpt';
```

After the call to `F1-GetValOpt` the following logic must be included to dynamically declare the map schema if the business object does not have a maintenance map of its own:

```
 // Perform Main Processing
if ("F1-GetBOOpts/output/maintenanceMap = $BLANK")
  declareBOWithBOGroup "$bo" as 'map_schema';
else
  declareMap "F1-GetBOOpts/output/maintenanceMap" as 'map_schema';
end-if;
```

Format an Input Map Title

> **ⓘ Note**
>
> Throughout this topic the term "field" to refer to both the generic concept of displaying and capturing data in a 'field' as well as referring to the meta-data object supplied in the product to define Fields. When referring to the latter, the term "MD Field" (meta-data Field) is used.

A uiHint element can be used to build a title for a maintenance map. The title will only print on the maintenance map, not on the display map. It will be printed as the first line in the map, centered, with a heading style.

Syntax	Description	Examples
`<uiHint:title mdField=""/>`	Displays the label of a referenced MD field as the title.	`<schema xmlns:uiHint="http://oracle.com/ouafUIHints"> <uiHint:title mdField="STATUS_RSN_LBL"/> ... </schema>`

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Description	Examples
`<uiHint:title text=" "/>`	Displays the indicated text as the title. (Do not use this mechanism when multiple languages are supported.	`<schema xmlns:uiHint="http://oracle.com/ouafUIHints"> <uiHint:title text="Status Reason"/> ... </schema>`

Create a Section

The uiHint namespace supports the definition of a UI map section. Note that sections are currently created in generated UI Maps when the schema has a group or list node with a label or mdField. The functionality described here enables the creation of a section without requiring a labeled group or list node within the schema. Every section must be bounded by **startSection** and **endSection** element pair.

Syntax	Supporting Attributes	Description
`<uiHint:startSection .../>`	`sectionColumn="left | right | fullWidth | float"`	The default is that the section will be the full width in display maps. To override that setting, specify if you want a half-width section to appear in either the left (**left**) or right (**right**) column or to float (**float**). Sections that are marked as 'float' will display half-width and be aligned according to whether prior sections are displayed or conditionally hidden. For example, if a left-aligned section is followed by a floating section, the floating section will appear in the right column if the left section is populated but will display in the left column if the left section is hidden / collapsed.
	`editColumn="left | right | fullWidth | float"`	By default a section appears as full width in maintenance maps. To override that setting, specify if you want a half-width section to appear in either the left (**left**) or right (**right**) column or to float (**float**). The behavior is the analogous to the **sectionColumn** behavior.
	`sectionOpen="false"`	By default a section is open on initial display. Specify this attribute to initially display the section as closed (collapsed).
	`mdField=" "`	Specify the name of a MD field whose label should be used as the section heading.
	`label=" "`	Specify the explicit text to use as the section heading.
	`visibleOn="displayMap | inputMap"`	By default a section appears on both the display and the input maps. Use this attribute to limit the display of the section to either the display map (**displayMap**) or input map (**inputMap**).

The syntax for the end section attribute is `<uiHint:endSection/>`

Copyright © 2026, Oracle and/or its affiliates.

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <uiHint:startSection label="Main" sectionColumn="left"/>
    ...
    <uiHint:endSection/>
</schema>


<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <uiHint:startSection mdField="F1-ADD-INFO" sectionColumn="fullWidth"
editColumn="float" sectionOpen="false" visibleOn="displayMap"/>
    ...
    <uiHint:endSection/>
</schema>
```

> **Note**
>
> The sectionColumn, editColumn and sectionOpen attributes are available for group and list nodes as well.

Include a Map Fragment

You can specify a UI map fragment to inject HTML into a generated map using the **includeMap** element name. This allows for you to support more sophisticated behavior on your user interface. For any element that is included for rendering in the map fragment, be sure to suppress the element in its schema definition, otherwise HTML will automatically be generated for the element.

Syntax	Supporting Attributes	Description
`<uiHint:includeMap ../>`	`map=" "`	Specify the name of the map.
	`visibleOn="displayMap | inputMap"`	By default the details from the map fragment appear on both the display and the input maps. Use this attribute to limit the display of the section.

Example:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  ...
  <uiHint:includeMap map="StandardActionButtons" visibleOn="displayMap"/>
  ...
</schema>
```


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Important note on the map fragment schema:** If a map fragment contains a schema, then the fragment schema structure will be injected into the dynamically generated schema when the business object is rendered for input. Technically, the fragment schema will be inserted after the boGroup structure within the map's schema. This method may be used to support the implementation of maintenance pre and post script processing for a business object and oraInvokeBS function calls within embedded JavaScript.

If JavaScript is required within an XHTML UI Map fragment, it is necessary to bound it within a **![CDATA[ ]]** tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">
/* <![CDATA[ */
//
//javascript
//
/* ]]> */
</script>
```

Flush the cache: For performance reasons, the Framework automatically caches business object schemas, data areas, and UI maps. When you update a business object, the cache is automatically flushed. However, if the business object includes either a data area or embedded UI map fragment, the cache must be manually flushed in order for your changes to be recognized. Refer to Server Cache for more information.

Build A Dropdown

Syntax is provided to build a dropdown list in an edit map. The dropdown may be built using data returned from a service script, a business service or a table.

Syntax	Description
`uiHint:select="ss: "`	Specify the name of the service script after the colon.
`uiHint:select="bs: "`	Specify the name of the business service after the colon.
`uiHint:select="table: "`	Specify the name of the table after the colon.

When specifying a service script or a business service, extra mapping information is needed to pass data to and from the service.

Syntax	Values	Description
`uiHint:selectIn=" "`	serviceXPath:element	Used to pass the value of another element into the service (mapping to the service's XPath).
	serviceXPath:'Literal'	Used to pass a constant or literal to the service (mapping to the service's XPath).

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Values	Description
`uiHint:selectOut="valuePath: ; descPath: "`	See examples below.	Used to indicate which element in the service's output holds the values and which one holds the descriptions.

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <boStatus mapField="BO_STATUS_CD" uiHint:select="bs:F1-BOStateReasonList"
    uiHint:selectIn="boStatusBO:boStatusBO"
uiHint:selectOut="valuePath:results/status;
    descPath:results/description"/>
    ...
    <algorithm mdField="ALG_CD" uiHint:select="bs:F1-
RetrieveSysEvtAlgorithms"
     uiHint:selectIn="algorithmEntity:'F1AA';"
uiHint:selectOut="valuePath:results/algorithm;
     descPath:results/description"/>
    ...
    <outboundMsgType mdField="OUTMSG_TYPE_CD" required="true" fkRef="F1-
OMTYP" uiHint:select="table:F1_OUTMSG_TYPE"/>
</schema>
```

Conditionally Hide Elements

The **displayNone** attribute is used to suppress elements on the map based on conditions.

Syntax	Values	Description
`uiHint:displayNone=`	"'XPath','value','!=' | '='"	Used to conditionally hide this element based on the value of another element (referenced using its XPath). Enter a value of `' '` to interrogate a blank value. By default the operator is `'='`. This may be overridden using `'!='`.
	"function name, **true** | **false**"	Used to indicate a JavaScript function, which must return a Boolean.

> **Warning**
>
> Embedded spaces are not supported within the comma separated string values of this attribute.
>
> This setting may be used on group nodes, list nodes, and elements - except for elements within a list. Elements within a list cannot be hidden conditionally.

The following example illustrates that two elements (currency reference and lookup) that will be hidden or displayed based on the value of the data type element. Note that this example also

Copyright © 2026, Oracle and/or its affiliates.

illustrates Trigger Dependent Behavior because the data type element's value may change and if it does, the condition for hiding the subsequent elements should be re-evaluated.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    ...
    <dataType mdField="F1_SE_DATA_TYPE" dataType="lookup"
lookup="F1_SE_DATA_TYPE"
     uiHint:dependents="currencyRef;lookup; "/>
     <currencyRef mdField="F1_SE_CURR_REF_LBL"
uiHint:displayNone="'dataType','F1MO','!='"/>
     <lookup mdField="F1_SE_LOOKUP_LBL" fkRef="F1-LKUPF"
uiHint:displayNone="'dataType','F1LP','!='"/>
     ...
</schema>
```

The following example illustrates referring to a function where the function receives parameters:

```
<uiHint:startSection mdField="F1_SE_DEFAULT_SECT"
    uiHint:displayNone="isApplicableForSchemaType(item,'F1MP'),true"/>
```

Conditionally Protect Elements

The protect attribute is used to protect elements on the map based on other factors.

Syntax	Values	Description
`uiHint:protect=`	"'XPath','value','!=' | '='"	Used to conditionally protect this element based on the value of another element (referenced using its XPath). Enter a value of ' ' to interrogate a blank value. By default the operator is `'='`. This may be overridden using `'!='`.
	"function name, **true** | **false**"	Used to indicate a JavaScript function, which must return a Boolean.
	"'**action**','A' | 'C','!=' | '='"	Use the `'action'` setting to protect the element based on the current action. For example, certain elements may only be specified when adding a record. Any subsequent changes to the record should protect the element from being changed. When using this option, the valid values for the *'value'* are `A` (add) and `C` (change).

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Embedded spaces are not supported within the comma separated string values of this attribute.
>
> The protect UI Hint may be used on group nodes, list nodes, and elements - except for elements within a list. Elements within a list cannot be protected conditionally.

The following UI Hint will protect the statistics category when the action is 'C'.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     ...
     <statisticsCategory dataType="lookup" mapField="STAT_CATEGORY_FLG"
    lookup="STAT_CATEGORY_FLG" uiHint:protect="'action','C','='"/>
     ...
    </schema>
```

Trigger Dependent Behavior

The dependents attribute is used to trigger behavior on a child element when a parent element is changed.

Syntax	Values
`uiHint:dependents=" "`	A list of one or more dependent elements separated by semicolons.

The following example illustrates that the dropdown list of one element is driven by the value of another element. In this example, when the Country changes, the list of States to choose from should change to only show the states for the indicated country.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  <country label="Country" uiHint:select="table:CI_COUNTRY"
uiHint:dependents="state"/>
  <state label="State" uiHint:select="ss:CM-RetrieveCountryStates"
   uiHint:selectIn="input/country:country;"
uiHint:selectOut="valuePath:output/state/stateCode;
   descPath:output/state/stateDesc"/>
    ...
</schema>
```

> **Note**
>
> Dependent targets may only name elements, not group or list nodes.
>
> Do not modify the "id" attribute value of dependent and parent element. Data population in dependent is done based on the "id" attribute value.

Copyright © 2026, Oracle and/or its affiliates.

Control Rendering Target

By default all elements that are not suppressed are visible on both the display map and the input map. Use the **visibleOn** attribute to limit the inclusion of an element to either the display or input map.

Syntax	Values
`uiHint:visibleOn=`	`"displayMap"` `"inputMap"`

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  ...
  <uiHint:includeMap map="StandardActionButtons" visibleOn="displayMap"
  ...
</schema>
```

Generate a Text Area

By default, a standard text box is rendered in an input map for any string element. If the field is larger and you wish to have a bigger text area (with a scroll bar), use the **textArea** attribute.

Syntax
`uiHint:textArea="true"`

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
   ...
  <message label="Message" uiHint:textArea="true"/>
  ...
</schema>
```

Modify FK Reference Defaults

By default, when an element with **fkRef** is displayed, an info string, context menu, navigation, and search are enabled (if the FK reference has been configured accordingly). Syntax is provided to allow you to selectively turn off any of these features.

Syntax
`uiHint:fkRef="info:false;context:false;navigation:false;search:false;"`

Only the feature that you wish to turn off needs to be specified. The following example illustrates turning off the navigation capability, meaning the text will not be rendered as hypertext.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
   ...
   <attachmentID fkRef="F1-ATTCH" primeKey="true" suppress="input"
uiHint:fkRef="navigation:false;"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   ...
</schema>
```

> **NOT_SUPPORTED**
>
> Refer to FK Reference Formatting in the UI Map Attributes section for more information on each FK reference setting.

Suppress Automatic Number Formatting

By default numeric fields (`dataType="number"`) are formatted as numbers. An attribute is provided to instead apply alphanumeric formatting.

Note: If **dataType** is not specified explicitly, it is be derived from **mdField** or **mapField**.

Syntax
`uiHint:alphaFormat="true|false"`

By default, its value is **false** (and therefore can be left out altogether).

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     ...
     <numberCount mdField="" dataType="number" uiHint:alphaFormat="true"/>
        ...
    </schema>
```

Auto Capitalize the Input Data

The uiHint provides syntax to automatically capitalize input data.

Syntax
`uiHint:capitalize="true|false"`

By default, its value is **false** (and therefore can be left out altogether).

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     <toDoTypeCd mdField="TD_TYPE_CD" uiHint:capitalize='true'
isPrimeKey="true"/>
</schema>
```

> **Note**
>
> This attribute is ignored if `uiHint:textArea="true"` is configured.

The attribute is only available in the schema designer when the **isPrimeKey** is set to **true**. The attribute may be added to any string element when using the source viewer.

Copyright © 2026, Oracle and/or its affiliates.

Override Row Header for Accessibility Purposes

When generating the HTML for a list, the default is to assign scope="row" to the non-suppressed primary key elements, if any, or the first non-suppressed element in the list. This uiHint provides syntax to override the defaults when they are not the appropriate elements to be used by a screen reader.

Syntax
`uiHint:rowHeader="true|false"`

```
<iwsAnnotationTypeParameter mdField="F1_ANN_TYPE_PARMS_LBL" type="list"
mapChild="F1_IWS_ANN_TYPE_PARM">
        <sequence dataType="number" mapField="SEQ_NUM"
uiHint:rowHeader="false"/>
        <parameterName mapField="PARM_NAME" uiHint:rowHeader="true"
dataType="string"/>
        <parameterValue mapField="PARM_VALUE" dataType="string"/>
        <isRequired dataType="lookupBO" mapField="REQUIRED_SW" lookupBO="F1-
BooleanValues"/>
        <javaPackage mapField="JAVA_PACKAGE" dataType="string"/>
        <version suppress="true" dataType="number" mapField="VERSION"/>
        <description mapField="DESCR" dataType="string"/>
        <longDescription mapField="DESCRLONG" uiHint:textArea="true"
dataType="string"/>
        <owner suppress="input" dataType="lookup" mapField="OWNER_FLG"
lookup="OWNER_FLG"/>
    </iwsAnnotationTypeParameter>
```

The attribute is only applicable to elements in a list.

Defining UI Maps

Use this page to define basic information about a user interface (UI) Map.

You can access the portal from the **Admin Menu**, and then **System**, and then **UI Map**. You are brought to a query portal with options for searching for a specific UI Map. Once a UI Map has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new UI Map, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding UI Maps for more information.

Use **UI Map Type** to indicate the map type. The system recommends using a **Complete XHTML Document** (the default) or **XHTML Fragment**. The values **Complete HTML Document** and **HTML Fragment** remain for backward compatibility but are not recommended.

The following zones may appear as part of the portal's **Main** tab page.

Copyright © 2026, Oracle and/or its affiliates.

- **UI Map.** This zone provides information about the selected UI Map.
- **HTML Editor.** This zone allows you to edit the HTML document of the map.

> **Note**
>
> Refer to UI Map Attributes and Functions and UI Map Standards for more information about HTML definition syntax. These topics describe good ways to produce simple HTML, however, they are not an HTML reference. Note that you can use a variety of HTML editors to compose your HTML, which you can then cut and paste into your UI map.

> **Note**
>
> **Providing Help.** A tool tip can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map. Refer to UI Map Attributes and Functions for more information on how to enable and provide UI map help.

- **HTML Override Editor.** This zone allows you to provide an override HTML document for a base product map.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the UI Map's schema using the Schema Designer tool. The purpose of the schema is to describe the UI Map's properties.

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Tip**
>
> Click the **View Schema** hyperlink to view the UI Map's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **UI Map References.** This zone list various configuration entities referencing the current UI Map, its HTML or its schema.

Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the UI Map's expanded schema definition. Doing this opens the schema viewer window.
- **Test UI Map.** Use the hyperlink to render your HTML in a test window. This action also exercises the proprietary functionality that binds an XML element with an HTML element so you can get immediate feedback on your HTML syntax.
- UI Map Attributes and Functions

Copyright © 2026, Oracle and/or its affiliates.

- UI Map Standards

UI Map Attributes and Functions

> **Note**
>
> Use the UI Map Query's Additional Information option to review HTML examples provided by the product.

Bind XML to HTML

Only two different attributes are required to bind a UI Map's XML to its HTML. Both of these attributes require an XML document embedded within the HTML, where the XML is bounded by <xml> nodes.

> **Warning**
>
> You must embed a pair of <xml></xml> tags within your HTML document for binding to occur.

Linking a Field

Syntax	Values	Description
oraField=" "	Field element XPath	This attribute is used to link an HTML element directly with an XML element, where the XML element is defined within the UI Map's XML schema. The attribute can be used with any *rendering* HTML element, such as: , , and <input>.

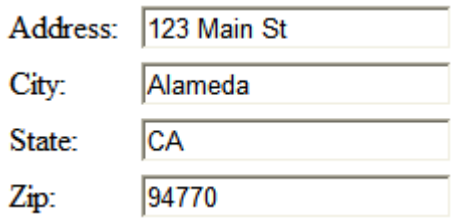
- HTML for input element:

```
<html>
<body>
<table>
   <tr>
      <td>Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
   <root>
      <address>123 Main St</address>
      <city>Alameda</city>
      <state>CA</state>
      <zip>94770</zip>
   </root>
</xml>
</html>
```

Rendered HTML



- HTML for span and div elements:

```
<html>
<body>

<div oraField="address"></div>
<span oraField="city"></span>
<span>,</span>
<span oraField="state"></span>
<span oraField="zip"></span>
<span oraField="country"></span>

</body>
<xml>
   <root>
      <address>123 Main St</address>
      <city>Alameda</city>
      <state>CA</state>
      <zip>94770</zip>
   </root>
</xml>
</html>
```

HTML rendered:

123 Main St
Alameda, CA 94770

Linking a List

Copyright © 2026, Oracle and/or its affiliates.

This attribute is used to link an HTML table with an XML list, where the XML list is defined within the UI Map's XML schema. The purpose of the element is to trigger the framework to replicate the table's HTML for each occurrence of the list.

Syntax	Values	Description
oraList=" "	List element XPath	This attribute is used to link an HTML table with an XML list, where the XML list is defined within the UI Map's XML schema. The purpose of the element is to trigger the framework to replicate the table's HTML for each occurrence of the list.

> **Note**
>
> The oraField attributes embedded within the list must contain XPath navigation relative to the list. See below for an example.

```
<html>
<head><title>Bind xml list element</title></head>
<body>
<table oraList="payment">
   <thead>
      <tr>
         <th><span>Pay Date</span></th>
         <th><span>Amount</span></th>
        </tr>
        <tr/>
    </thead>
    <tr>
      <td>
         <span oraField="date" oraType="date"></span>
      </td>
      <td align="right">
         <span oraField="amount" oraType="money"></span>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
    <payment>
        <date>2008-01-01</date>
        <amount>44.28</amount>
    </payment>
    <payment>
        <date>2008-02-01</date>
        <amount>32.87</amount>
    </payment>
    <payment>
        <date>2008-03-01</date>
        <amount>21.76</amount>
    </payment>
</root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

HTML rendered:

Pay Date	Amount
01-01-2008	$44.28
02-01-2008	$32.87
03-01-2008	$21.76

Build a Dropdown List

The following attributes are provided to build an HTML ‘select’ element, also called a dropdown, based on various sources.

Source	Syntax	Values	Examples

Lookup	`oraSelect="lookup: ;"`	Lookup field	`...`
			`<td>House Type:</td>`
			`<td>`
			`<select oraField="houseType" oraSelect="lookup:HOUSE_TYPE;" >`
			`</select>`
			`</td>`

Extendable Lookup	`oraSelect="lookupBO: ;"`	BO code	`...`
			`<td>UI Device Display Type:</td>`
			`<td>`
			`<select oraField="uiDeviceType" oraSelect="lookupBO: F1-DeviceDisplayTypes;"></select>`
			`</td>`

Characteristic Type (pre-defined)	`oraSelect="charType: ;"`	Characteristic Type code	`...`
			`<td>Usage:</td>`
			`<td>`
			`<select oraField="statusReasonUsage" oraSelect="charType:F1-SRUSG;"></select>`
			`</td>`
			`...`

Copyright © 2026, Oracle and/or its affiliates.

Source	Syntax	Values	Examples
Control Table	`oraSelect="table: ;"`	Table name	``` ... <td>Currency: </td> <td> <select oraField="currency" oraSelect="table:CI_CURRENCY_CD;"> </select> </td> ... ```

Note

This attribute only works with tables that follow the esta

Source	Syntax	Values	Examples
		ndardcontroltablestructurewhereretherereisarelatedlanguageta	

Copyright © 2026, Oracle and/or its affiliates.

Source	Syntax	Values	Examples
		bl eth at in cl ud es th at in cl ud es th e co lu mn **D E S C R** or D E S C R 1 0 0 as it	

Copyright © 2026, Oracle and/or its affiliates.

Source	Syntax	Values	Examples
		sdescriptioncolumn.Gotothe Field maintenancepage foreachfiel	

Copyright © 2026, Oracle and/or its affiliates.

Source	Syntax	Values	Examples

dtoseewhattablescontainthefields.

Warning

TheoraSelect

Source	Syntax	Values	Examples
		function will only work if less than 500 values are displayed.	

Copyright © 2026, Oracle and/or its affiliates.

Source	Syntax	Values	Examples

Page Service	`oraSelect="service: ;"`	Page Service name	`...`
			`<td>Country:</td>`
			`<td>`
			`<select oraField="country" oraSelect="service:CIPTCNTW;">`
			`</select>`
			`</td>`
			`...`

Embedded List	Used to build a select dropdown based on a list within the map's XML. `oraSelect="valuePath: ;descPath: "`	After the valuePath, indicate the XPath of the element that holds the values. After the descPath, indicate the XPath of the element that holds the descriptions.	`<html>`
			`<body>`
			`<table summary="" border="0" cellpadding="1" cellspacing="1">`
			`<tr>`
			`<td>Select: </td>`
			`<td><select oraSelect= "valuePath:list/value; descPath:list/desc" oraField="target"></ select></td>`
			`</tr>`
			`</table>`
			`</body>`
			`<xml>`
			`<root>`
			`<target>10</target>`
			`<list>`
			`<value>10</value>`
			`<desc>Ten</desc>`
			`</list>`
			`<list>`
			`<value>20</value>`
			`<desc>Twenty</desc>`
			`</list>`
			`<list>`
			`<value>40</value>`
			`<desc>Forty</desc>`
			`</list>`
			`</root>`
			`</xml>`
			`</html>`

Service Script	`oraSelect="ss: ; "`	Service Script code	See below for additional syntax needed when using this function.
Business Service	`oraSelect="bs: ; "`	Business Service code	See below for additional syntax needed when using this function.

Copyright © 2026, Oracle and/or its affiliates.

When specifying a service script or a business service, extra mapping information is needed to pass data to and from the service.

Syntax	Values	Description
oraSelectIn=" ;"	serviceXPath:element;	Used to pass the value of another element into the service (mapping to the service's XPath).
	serviceXPath:'Literal';	Used to pass a constant or literal to the service (mapping to the service's XPath).
oraSelectOut= "valuePath: ; descPath: "	See examples below	Used to indicate which element in the service's output holds the values and which one holds the descriptions.

Example using a business service:

```
...
<td>External System: </td>
<td>
   <select oraField="externalSystem"
    oraSelect="bs:F1-RetrieveExternalSystems"
    oraSelectIn="outboundMsgType:boGroup/parameters/outboundMsgType;"
    oraSelectOut="valuePath:results/externalSystem;
    descPath:results/description">
   </select>
</td>
...
```

This method for building dropdowns is often used when there is a dependency between elements and the list of valid values in a dropdown (for the child element) is based on another element in the map (the parent element). When the parent element is changed, it may be required to refresh the child element. This behavior can be implemented using the function called within an **onChange** event in the map. The syntax is **oraHandleDependentElements('dependent element');**. Multiple target elements (dependents) can be named.

The following example is related to the above business service example where the list of external systems is specific for a given outbound message type, which is passed in as input. The snippet below shows the configuration for the outbound message type element to trigger a refresh of the external system’s dropdown list.

```
...
<div>
   <label oraLabel="boGroup/parameters/outboundMsgType"></label>
      <span>
         <select oraSelect="table:F1_OUTMSG_TYPE"
          oraField="boGroup/parameters/outboundMsgType"
          onChange="oraHandleDependentElements('boGroup/parameters/
externalSystem');">
         </select>
      </span>
</div>
...
```


Copyright © 2026, Oracle and/or its affiliates.

Format Input and Output Fields

The following attributes are designed to apply data type formatting for input and output fields.

Automatic Formatting

Syntax
`oraSchemaDataTypes="false"`

This attribute is used to trigger automatic formatting in the rendered HTML document. The automated formatting will occur according to the data type attributes defined in the UI map's schema. For details on specific data type formatting, please refer to the oraType attribute descriptions below.

> **Warning**
>
> The attribute **oraSchemaDataTypes="true"** will be automatically injected into the UI map's HTML! If you do not wish to apply the schema's data types to the rendered HTML then you must specify this attribute in the body node with a value of false. The attribute **<body oraSchemaDataTypes="false">** is required to avoid automatic formatting!

- UI Map schema:

```
<schema>
    <schemaDate dataType="date"/>
    <schemaDateTime dataType="dateTime"/>
    <schemaFkRef fkRef="F1-MOTBL"/>
    <schemaLookup dataType="lookup" lookup="ACCESS_MODE"/>
    <schemaMoney dataType="money"/>
    <schemaNumber dataType="number"/>
    <schemaTime dataType="time"/>
</schema>
```

- UI Map HTML:

```
<html>
<body oraSchemaDataTypes="true">
<table border="1" cellpadding="1" cellspacing="1">
   <tr><th>dataType</th><th>input result</th><th> display-only result</
th></tr>
   <tr>
      <td>date</td>
      <td><input oraField="schemaDate"/></td>
      <td><span oraField="schemaDate"></span></td>
   </tr>
   <tr>
      <td>dateTime</td>
      <td><input oraField="schemaDateTime"/></td>
      <td><span oraField="schemaDateTime"></span></td>
   </tr>
   <tr>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <td>fkRef</td>
      <td><input oraField="schemaFkRef"/></td>
      <td><span oraField="schemaFkRef"></span></td>
   </tr>
   <tr>
      <td>lookup</td>
      <td><input oraField="schemaLookup"/></td>
      <td><span oraField="schemaLookup"></span></td>
   </tr>
   <tr>
      <td>money</td>
      <td><input oraField="schemaMoney"/></td>
      <td><span oraField="schemaMoney"></span></td>
   </tr>
   <tr>
      <td>number</td>
      <td><input oraField="schemaNumber"/></td>
      <td><span oraField="schemaNumber"></span></td>
   </tr>
   <tr>
      <td>time</td>
      <td><input oraField="schemaTime"/></td>
      <td><span oraField="schemaTime"></span></td>
   </tr>
</table>

</body>
<xml>
<root>
   <schemaDate>2007-11-02</schemaDate>
   <schemaDateTime>2007-11-02-23.45.00</schemaDateTime>
   <schemaFkRef>CI_MD_TBL</schemaFkRef>
   <schemaLookup>A</schemaLookup>
   <schemaMoney>1000000</schemaMoney>
   <schemaNumber>5661976.11548</schemaNumber>
   <schemaTime>23.45.00</schemaTime>
</root>
</xml>
</html>
```

HTML rendered.

dataType	input result	display-only result
date	11-02-2007	11-02-2007
dateTime	11-02-2007 11:45:00PM	11-02-2007 11:45:00PM
fkRef	CI_MD_TBL Table	Table
lookup	A	Add
money	$1,000,000.00	$1,000,000.00
number	5,661,976	5,661,976
time	11:45:00PM	11:45:00PM

Date and Time Formatting

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Valid Values	Description
`oraType="date"`		This function is used to display a date according to the user's display profile. For input fields, the setting formats the data when the user tabs out of the field.
`oraType="time"`		This function is used to display a time according to the user's display profile. For input fields, the setting formats the data when the user tabs out of the field.
`oraType="dateTime; time:suppress"`		This function is used to display a timestamp according to the user's display profile. If this function is used for an input element, it is broken into two pieces, one for date and one for time. Optionally, the time portion of the date time element can be suppressed using the attribute value 'time:suppress'.
`oraType="dateTime; stdTime:true;"`		This function is used to render a date / time element that is stored in standard time according to the daylight savings time schedule of the 'base' time zone. The 'base' time zone is specified on the Installation table and represents the database time zone. For input elements with this setting, all time entered is assumed to correspond with the daylight savings time schedule of the base time zone. If a time is entered that cannot be unambiguously translated to standard time, then the user will be required to provide a time zone label to indicate whether daylight savings time, or standard time, has been entered.
`oraType="dateTime; stdTimeRef:;"`	Reference an XPath after the colon.	This function is used to render a date / time element according to the daylight savings time schedule of a time zone whose XPath is referenced. Note that the time processed is assumed to have been stored in the standard time of the referenced time zone - so only daylight savings time shifting will execute - not time zone shifting.
`oraType="dateTime; legalTime:true;"`		This function is used to render a date / time element that is stored in legal time of the 'base' time zone. The 'base' time zone is specified on the Installation table and represents the database time zone.
`oraType="dateTime; legalTimeRef: ;"`	Reference an XPath after the colon.	This function is used to render a date / time element according to the daylight savings time schedule of a time zone whose XPath is referenced. Note that the time processed is assumed to have been stored in the legal time of the referenced time zone.
`oraType="dateTime; … ; displayRef:;"`	Reference an XPath after the colon.	This function is used to display a date/time element that is explicitly defined as stored in standard or legal time (using either stdTime | stdRef | legalTime | legalRef) from its storage time zone to the referenced time zone in legal time before display and vice versa, from the display time zone to the storage time zone on save.

```
<html>
<body oraSchemaDataTypes="true">
<table border="1" cellpadding="1" cellspacing="1">
   <tr><th>dataType</th><th>input result</th><th> display-only result</
th></tr>
   <tr>
      <td>date</td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <td><input oraField="schemaDate" oraType="date;"/></td>
      <td><span oraField="schemaDate" oraType="date;"></span></td>
   </tr>
   <tr>
      <td>time</td>
      <td><input oraField="schemaTime" oraType="time;"/></td>
      <td><span oraField="schemaTime" oraType="time;"></span></td>
   </tr>
   <tr>
      <td>dateTime</td>
      <td><input oraField="schemaDateTime" oraType="dateTime;"/></td>
      <td><span oraField="schemaDateTime" oraType="dateTime;"></span></td>
   </tr>
   <tr>
      <td>dateTimeSuppress</td>
      <td><input oraField="schemaDateTimeSup" oraType="dateTime;
time:suppress"/></td>
      <td><span oraField="schemaDateTimeSup" oraType="dateTime;
time:suppress"></span></td>
   </tr>
   <tr>
     <td>Base Time Zone (Standard)</td>
     <td><input oraField="schemaDateTimeStd" oraType="dateTime;
stdTime:true;"/></td>
     <td><span oraField="schemaDateTimeStd" oraType="dateTime;
stdTime:true;"></span></td>
   </tr>
   <tr>
     <td>Base Time Zone (Legal)</td>
     <td><input oraField="schemaDateTimeLgl" oraType="dateTime;
legalTime:true;"/></td>
     <td><span oraField="schemaDateTimeLgl" oraType="dateTime;
legalTime:true;"></span></td>
   </tr>
   <tr>
     <td>Referenced Time Zone (Standard)</td>
     <td><input oraField="schemaDateTimeStdRef" oraType="dateTime;
stdTimeRef:timeZone;"/></td>
     <td><span oraField="schemaDateTimeStdRef" oraType="dateTime;
stdTimeRef:timeZone;"></span></td>
   </tr>
   <tr>
     <td>Referenced Time Zone (Legal)</td>
     <td><input oraField="schemaDateTimeLglRef" oraType="dateTime;
legalTimeRef:timeZone;"/></td>
     <td><span oraField="schemaDateTimeLglRef" oraType="dateTime;
legalTimeRef:timeZone;"></span></td>
   </tr>
   <tr>
     <td>Stored in Base Time Zone (Standard) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeStdDisp" oraType="dateTime;
stdTime:true; displayRef:timeZoneDisp;"/></td>
     <td><span oraField="schemaDateTimeStdDisp" oraType="dateTime;
stdTime:true; displayRef:timeZoneDisp;"></span></td>
   </tr>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   <tr>
     <td>Stored in Base Time Zone (Legal) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeLglDisp" oraType="dateTime;
legalTime:true; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeLglDisp" oraType="dateTime;
legalTime:true; displayRef:timeZoneDisp"></span></td>
   </tr>
   <tr>
     <td>Stored in a Referenced Time Zone (Standard) - displayed in another
Time Zone</td>
     <td><input oraField="schemaDateTimeStdRefDisp" oraType="dateTime;
stdTimeRef:timeZone; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeStdRefDisp" oraType="dateTime;
stdTimeRef:timeZone; displayRef:timeZoneDisp"></span></td>
   </tr>
   <tr>
     <td>Stored in a Referenced Time Zone (Legal) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeLglRefDisp" oraType="dateTime;
legalTimeRef:timeZone; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeLglRefDisp" oraType="dateTime;
legalTimeRef:timeZone; displayRef:timeZoneDisp"></span></td>
   </tr>

  </table>

</body>
<xml>
<root>
   <timeZone>US-EASTERN</timeZone>
   <timeZoneDisp>US-CENTRAL</timeZoneDisp>
   <schemaDate>2007-11-02</schemaDate>
   <schemaTime>23.45.00</schemaTime>
   <schemaDateTime>2007-11-02-23.45.00</schemaDateTime>
   <schemaDateTimeSup>2007-11-02-23.45.00</schemaDateTimeSup>
   <schemaDateTimeStd>2007-11-02-23.45.00</schemaDateTimeStd>
   <schemaDateTimeStdRef>2007-11-02-23.45.00</schemaDateTimeStdRef>
   <schemaDateTimeLgl>2007-11-02-23.45.00</schemaDateTimeLgl>
   <schemaDateTimeLglRef>2007-11-02-23.45.00</schemaDateTimeLglRef>
   <schemaDateTimeStdDisp>2007-11-02-23.45.00</schemaDateTimeStdDisp>
   <schemaDateTimeStdRefDisp>2007-11-02-23.45.00</schemaDateTimeStdRefDisp>
   <schemaDateTimeLglDisp>2007-11-02-23.45.00</schemaDateTimeLglDisp>
   <schemaDateTimeLglRefDisp>2007-11-02-23.45.00</schemaDateTimeLglRefDisp>
</root>
</xml>
</html>
```

HTML rendered.

Copyright © 2026, Oracle and/or its affiliates.

dataType	input result	display-only result
date	11-02-2009	11-02-2009
time	12:28:54AM	12:28:54AM
dateTime	11-02-2009 12:28:54AM	11-02-2009 12:28:54AM
dateTimeSuppress	11-02-2009	11-02-2009
Base Time Zone (Standard) - Duplicate hour	11-01-2009 1:28:54AM PDT	11-01-2009 1:28:54AM PDT
Base Time Zone (Standard)	11-02-2009 12:28:54AM PST	11-02-2009 12:28:54AM PST
Base Time Zone (Legal)	11-02-2009 12:28:54AM PST	11-02-2009 12:28:54AM PST
Referenced Time Zone (Standard)	11-02-2009 12:28:54AM EST	11-02-2009 12:28:54AM EST
Referenced Time Zone (Legal)	11-02-2009 12:28:54AM EST	11-02-2009 12:28:54AM EST
Stored in Base Time Zone (Standard) - displayed in another Time Zone	11-02-2009 2:28:54AM CST	11-02-2009 2:28:54AM CST
Stored in Base Time Zone (Legal) - displayed in another Time Zone	11-02-2009 2:28:54AM CST	11-02-2009 2:28:54AM CST
Stored in a Referenced Time Zone (Standard) - displayed in another Time Zone	11-01-2009 11:28:54PM CST	11-01-2009 11:28:54PM CST
Stored in a Referenced Time Zone (Legal) - displayed in another Time Zone	11-01-2009 11:28:54PM CST	11-01-2009 11:28:54PM CST

When an element is associated with a time zone the corresponding time zone name is displayed along with the value. This is specifically useful in the example referenced as "Base Time Zone (Standard) - Duplicate hour". The time 1:28 is ambiguous without a time zone label. Legally, November 1, 2009 1:28 AM occurs twice because daylight savings time (DST) is removed at 2:00 AM. The time zone name indicates whether the first or second hour is specified for the time zone.

Duration Formatting

Syntax

```
oraType="duration"
```

This function is used to display time duration. For an input element, the value entered by the user is translated from minutes to hour and minutes as appropriate. For example, an entered value of '90', is converted to '00:01:30' when tabbing out of the input field.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Duration: </td>
      <td><span oraField="duration" oraType="duration"></span></td>
   </tr>
   <tr>
      <td>Duration: </td>
      <td><input oraField="duration" oraType="duration"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <duration>90</duration>
</root>
</xml>
</html>
```

HTML rendered.

Duration: 00:01:30

Duration: 00:01:30

Copyright © 2026, Oracle and/or its affiliates.

Day in Month Formatting

Syntax
`oraType="dayInMonth"`

This function is used to display a concatenated day and month.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Day In Month: </td>
      <td><span oraField="dayMonth" oraType="dayInMonth"></span></td>
   </tr>
   <tr>
      <td>Day In Month: </td>
      <td><input oraField="dayMonth" oraType="dayInMonth"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <dayMonth>0228</dayMonth>
</root>
</xml>
</html>
```

HTML rendered.

Day In Month: 02-28

Day In Month: 02-28

Month In Year Formatting

Syntax
`oraType="monthInYear"`

This function is used to display a concatenated month and year.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Month In Year: </td>
      <td><span oraField="month" oraType="monthInYear"></span></td>
   </tr>
   <tr>
      <td>Month In Year: </td>
      <td><input oraField="month" oraType="monthInYear"/></td>
   </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
   <month>200811</month>
</root>
</xml>
</html>
```

HTML rendered.



Monetary Formatting

This function is used to display a number as a monetary amount. See the table for configuration options with respect to the currency. For input elements, an error is issued if a non-numeric value is entered.

Syntax	Description
`oraType="money: "`	Directly specify a currency code after the colon.
`oraType="money;currencyRef: "`	Reference an XPath (after the colon) for an element that references a currency code.
`oraType="money"`	If no currency or currency reference is specified, the installation currency is used.

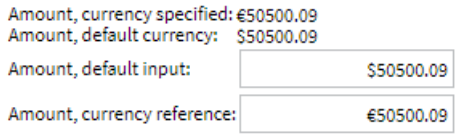
> **Note**
>
> You must specify the stylesheet reference cisDisabled in the map's header for right alignment. The stylesheet controls how the field will be rendered. If you want to alter the rendering you must override the oraMoney style.

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Amount, currency specified:</td>
      <td><span oraType="money:EUR" oraField="totalAmt"></span></td>
   </tr>
   <tr>
      <td>Amount, default currency:</td>
      <td><span oraType="money" oraField="totalAmt"></span></td>
   </tr>
   <tr>
      <td>Amount, default input:</td>
      <td><input oraType="money" oraField="totalAmt"/></td>
   </tr>
   <tr>
      <td>Amount, currency reference:</td>
      <td><input oraType="money;currencyRef:cur" oraField="totalAmt"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
<root>
   <totalAmt>50500.09</totalAmt>
   <cur>EUR</cur>
</root>
</xml>
</html>
```

HTML rendered



Number Formatting

This function is used to display a number or validate an input value. For input elements, the system will return an error if a non-numeric value is entered.

Syntax
`oraType="number"`

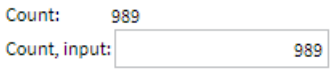
> **Note**
>
> You must specify the stylesheet reference cisDisabled in the map's header for right alignment. The stylesheet controls how the field will be rendered. If you want to alter the rendering you must override the oraNumber style.

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Count:</td>
      <td><span oraType="number" oraField="count"></span></td>
   </tr>
   <tr>
      <td>Count, input:</td>
      <td><input oraType="number" oraField="count"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <count>989</count>
</root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

HTML rendered.



FK Reference Formatting

By default, when an element with an **fkRef** oraType is displayed, an info string, context menu, navigation, and search are enabled (if the FK reference has been configured accordingly). Syntax is provided to allow you to selectively turn off any of these features.

Note that you can enable the foreign key hyperlink separately as well, refer to Embed Framework Navigation for more information. The various attributes used to control foreign key reference functionality are as follows. Note that in every case, the default value is **true**. A value of **false** should be used to disable the feature.

Syntax
`oraType="fkRef:true|false; info:true|false; context:true|false; navigation:true|false; search:true|false"`

- **fkRef**. A value of 'true' enables all of the following foreign key reference processing. Use a value of 'false' to disable automatic FK Reference processing.
- **info**. A value of 'true' renders an information string on the UI map, if applicable.
- **context**. A value of 'true' renders a context menu to appear before the foreign key reference element, if applicable.
- **navigation**. A value of 'true' causes the information string to be rendered as a hyperlink, if applicable. Clicking the hyperlink navigates to the appropriate page.
- **search**. A value of 'true' displays a search icon that launches the search zone if applicable.

> **Note**
>
> Foreign key navigation and context menu functionality is only available for UI maps presented in a portal zone. UI Maps presented during BPA script processing cannot support navigation options. Search functionality is only available for input HTML elements.

- UI Map schema:

```
<schema>
  <bo fkRef="F1-BOMO"/>
</schema>
```

- UI Map HTML:

```
<html>
<body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Business Object</td>
      <td><span oraField="bo" oraType="fkRef:true; info:true;
context:true; navigation:true;"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <bo>F1-COUNTRY</bo>
</root>
</xml>
</html>
```

- HTML rendered.



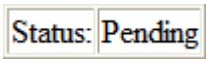
Lookup Formatting

This function is used to display the description of a lookup value.

Syntax	Valid Values
`oraType="lookup: "`	Lookup field name after the colon.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Status:</td>
      <td><span oraField="status" oraType="lookup:BATCH_JOB_STAT_FLG"></
span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <status>PD</status>
</root>
</xml>
</html>
```

HTML rendered.



Extendable Lookup Formatting

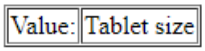
This function is used to display the description of an extendable lookup value.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Valid Values
`oraType="lookupBO: "`	Business Object code name after the colon.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Value:</td>
      <td><span oraField="status" oraType="lookupBO:F1-DeviceDisplayTypes"></
span></td>
   </tr>
</table>
</body>
<xml>
<root>
    <status>oraTablet</status>
</root>
</xml>
</html>
```

HTML rendered.



Characteristic Type Formatting

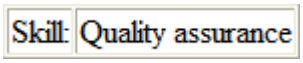
This function is used to display the description of a predefined characteristic value.

Syntax	Valid Values
`oraType="charType: "`	Characteristic Type code after the colon.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td>Skill:</td>
        <td><span oraType="charType:CM-SKILL" oraField="skill"></span></td>
    </tr>
</table>
</body>
<xml>
<root>
    <skill>10</skill>
</root>
</xml>
</html>
```

HTML rendered.

Copyright © 2026, Oracle and/or its affiliates.



Control Table Formatting

This function is used to display the description of a control table that has an associated language table.

Syntax	Valid Values
`oraType="table: "`	Table code after the colon.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Currency:</td>
      <td><span oraType="table:CI_CURRENCY_CD" oraField="curr"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <curr>USD</curr>
</root>
</xml>
</html>
```

HTML rendered.



Add / Remove Grid Formatting

Syntax	Description
`oraType="addGridRow"`	The addGridRow function is used to build "insert row" dialog into the UI map. • An 'add' image is displayed. • Clicking the image inserts a new row in the grid. • If the list is empty, by default, an empty grid row is automatically be added. This means that the user will always see at least one grid row when this attribute is used.
`oraType="deleteGridRow"`	The deleteGridRow function is used to build "delete row" dialog into the UI map. • A 'delete' image is displayed. • Clicking the image removes the adjacent row from the grid.

> **Note**
>
> Because add and delete dialogs are relevant only inside a table, these attributes must be specified within a <td> element.

Copyright © 2026, Oracle and/or its affiliates.

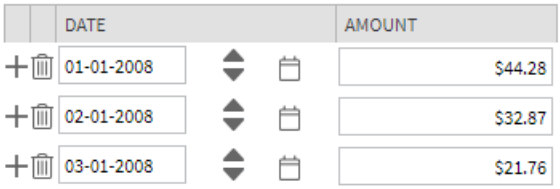
> **Warning**
>
> These attributes are designed to work with the business object action of 'replace' rather than 'update'. Therefore, if the map contains a grid, the business object action of 'replace' must be used to update the business object. Refer to BO Replace Action for more information.

Example:

```
<html>
<head>
<title>Demonstrate Grid Add and Grid Delete OraTypes</title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table oraList="listEntry">
   <thead>
      <tr>
         <th/>
         <th/>
         <th><span>Date</span></th>
         <th><span>Amount</span></th>
      </tr>
      <tr/>
   </thead>
   <tr>
      <td oraType="addGridRow"></td>
      <td oraType="deleteGridRow"></td>
      <td>
         <input oraField="date" oraType="date"></input>
      </td>
      <td align="right">
         <input oraField="amount" oraType="money"></input>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <listEntry>
      <date>2008-01-01</date>
      <amount>44.28</amount>
   </listEntry>
   <listEntry>
      <date>2008-02-01</date>
      <amount>32.87</amount>
   </listEntry>
   <listEntry>
      <date>2008-03-01</date>
      <amount>21.76</amount>
   </listEntry>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.



Unformatted Elements

This function is used to display the contents of an element that contains 'raw' data as defined for the schema element being rendered.

Syntax
`oraType="raw"`

- UI Map schema:

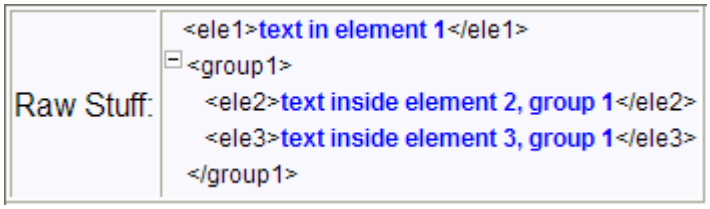
```
<schema>
    <rawStuff type="raw"/>
</schema>
```

- UI Map HTML:

```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Raw Stuff:</td>
      <td><span oraType="raw" oraField="rawStuff"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <rawStuff>
      <ele1>text in element 1</ele1>
      <group1>
         <ele2>text inside element 2, group 1</ele2>
         <ele3>text inside element 3, group 1</ele3>
      </group1>
   </rawStuff>
</root>
</xml>
</html>
```

HTML rendered.

Copyright © 2026, Oracle and/or its affiliates.



String Formatting

This function is used to display the contents of an element, as XML pretty-print, when the element contains escaped XML.

Syntax
`oraType="xmlString"`

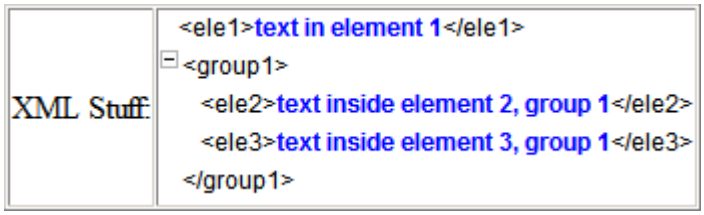
> **Note**
>
> It is not required, but the pretty print of the rendered XML is enhanced if you specify the stylesheet reference cisDisabled in the map's header.

Example:

```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>XML Stuff:</td>
      <td><span oraType="xmlString" oraField="xmlStuff"></span></td>
      </tr>
</table>
</body>
<xml>
<root>
   <xmlStuff>
      <ele1>text in element 1</ele1>
      <group1>
         <ele2>text inside element 2, group 1</ele2>
         <ele3>text inside element 3, group 1</ele3>
      </group1>
   </xmlStuff>
</root>
</xml>
</html>
```

HTML rendered.

Copyright © 2026, Oracle and/or its affiliates.



HTML rendered without oraType="xmlString"

XML Stuff:	<ele1>text in element 1</ele1><group1><ele2>text inside element 2, group 1</ele2><ele3>text inside element 3, group 1</ele3></group1>

HTML Formatting

This function is used to display the contents of an element as HTML as opposed to plain text. An element defined as oraType="fkref" is automatically rendered as HTML.

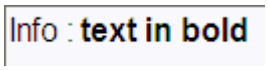
Syntax
`oraType="html"`

> **Warning**
>
> To avoid execution of malicious HTML not all HTML tags are supported. The list of supported tags is defined in the "F1-HTMLWhiteList" managed content definition.
>
> If unsupported HTML is detected the entire element is escaped and rendered as plain text. It is therefore recommended to properly escape any source string that contributes to the final HTML element if it is not expected to contain valid HTML. This way only the offending string is escaped and not the entire element.
>
> If the HTML element is composed in scripting refer to the 'escape' function described in the Edit Data Syntax for more information. Use the WebStringUtilities.asHTML java API for escaping text composed in Java.

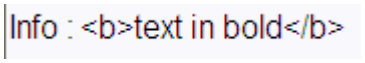
```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
   <td>Info :</td>
   <td><span oraType="html" oraField="info"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <info><b>text in bold</b></info>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.



HTML rendered without oraType="html"



Display Labels

Derive Label from an Element

This attribute is used to obtain a language sensitive label for a , <td>, or <input> HTML element.

Syntax	Valid Values
`oraLabel=" "`	XPath of an element in the UI map schema. The element must reference either the `mapField=` , `mdField=` , or `label=` attribute.

> **Note**
>
> You can also define a field directly in your HTML for label definition, refer to Deriving Label from a Field for more information.

> **Note**
>
> If the schema contains multiple attributes, the oraLabel attribute will pick the label to render according to the following hierarchy: The label attribute overrides the mdField attribute, which in turn will override the mapField attribute.

- UI Map schema:

```
<schema>
   <user mapField="USER_ID"/>
   <info type="group" mapXML="CLOB">
      <city label="Metro Area"/>
      <age mdField="AGE_LBL"/>
   </info>
</schema>
```

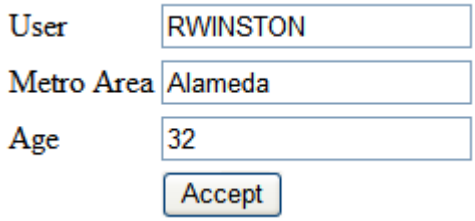
- HTML:

```
<html>
<head><title oraMdLabel="BUS_LBL"></title></head>
<body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<table>
   <tr>
      <td oraLabel="user"></td>
      <td><input oraField="user"/></td>
   </tr>
   <tr>
      <td oraLabel="info/city"></td>
      <td><input oraField="info/city"/></td>
   </tr>
   <tr>
      <td oraLabel="info/age"></td>
      <td><input oraField="info/age"/></td>
   </tr>
   <tr>
      <td/>
      <td><input type="button" oraMdLabel="ACCEPT_LBL"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <user>RWINSTON</user>
   <info>
      <city>Alameda</city>
      <age>32</age>
   </info>
</root>
</xml>
</html>
```

HTML rendered:



Deriving Label from a Field

This attribute is used to obtain a language sensitive label for a , <td>, <input>, or <title> HTML element. The label text is derived from the field referenced.

Syntax	Valid Values
`oraMdLabel=" "`	MD Field code.

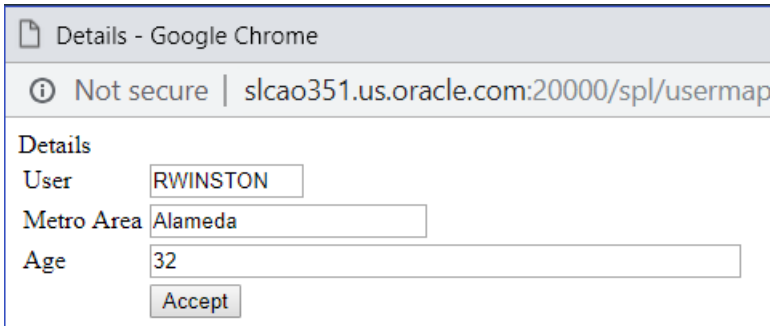
> **Note**
>
> You can also define labels derived from the map's schema definition, refer to Derive Label from an Element for more information.

Copyright © 2026, Oracle and/or its affiliates.

- HTML:

```
<html>
<head><title oraMdLabel="F1_DETAILS_LBL"></title></head>
<body>
<table>
   <tr>
      <td oraLabel="user"></td>
      <td><input oraField="user"/></td>
   </tr>
   <tr>
      <td oraLabel="info/city"></td>
      <td><input oraField="info/city"/></td>
   </tr>
   <tr>
      <td oraLabel="info/age"></td>
      <td><input oraField="info/age"/></td>
   </tr>
   <tr>
      <td/>
      <td><input type="button" oraMdLabel="ACCEPT_LBL"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <user>RWINSTON</user>
   <info>
      <city>Alameda</city>
      <age>32</age>
   </info>
</root>
</xml>
</html>
```

HTML rendered:



Enable UI Map Help

The Display Labels section describes ways to derive the label for an element using an underlying MD Field. In addition, if the same MD field contains help text, the system will automatically generate a tool tip adjacent to the element label. Clicking the tool tip allows the user to view the help text.

Copyright © 2026, Oracle and/or its affiliates.

It is possible to change the rendering of the tool tip. Refer to Custom Look And Feel Options for more information

Search Using a Pop-Up Explorer Zone

Search Option

This attribute is used to enable search zone functionality for input HTML elements.

Syntax	Valid Values
`oraSearch=" "`	Zone code.

> **Note**
>
> The oraSearch attribute is similar to the oraType attribute, because it will be 'automatically' included into HTML via the oraSchemaDataTypes attribute. This means that coding the oraSearch attribute into UI Map HTML is only required if a search zone has not been specified in the schema, or in the schema element's FK reference.

- Example of defining the search in the HTML. UI Map's Schema:

```
<schema>
   <uiMap/>
</schema>
```

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap" oraSearch="F1-UISRCH"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

- Example of defining the search in the schema. UI Map's Schema:

```
<schema>
   <uiMap search="F1-UISRCH"/>
</schema>
```


Copyright © 2026, Oracle and/or its affiliates.

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

- Example where the FK reference defines the search zone. UI Map's Schema:

```
<schema>
   <uiMap fkRef="F1-UISRC"/>
</schema>
```

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

In all cases, the HTML rendered is the same.



Initializing Search Fields

This optional attribute is used to initialize search zone filters. Multiple filters may be initialized. This attribute can only be used in conjunction with the oraSearch attribute.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Valid Values	Field Value Options	Comments
`oraSearchField=" "`	One or more pairs of field name and field value separated by colon. Each pair is separated by a semi-colon. `oraSearchField="fieldName:fieldValue: ..."` The field name is used to identify the zone filter to initialize when the search is launched. The field name must match the value of the searchField mnemonic specified on a search zone user filter or hidden filter parameter.	No value	If you do not specify a field value, then the value of the input element containing the oraSearchField attribute will be used.
		XPath	Indicate the XPath to the schema element that contains the value to use.
		'literal'	Indicate a literal value to supply.

> **Note**
>
> If you do not specify an oraSearchField attribute and the schema element has a search enabled fkRef specified, the framework automatically builds an oraSearchField where the field name is equal to the FK reference's key (MD) fields.

> **Warning**
>
> The pop-up explorer zone can be invoked one of two ways: By clicking on the search button, or by hitting the enter key from the field to the left of the button. If you click on the button then no search field information will be passed to the zone. Search field information is only used to initialize zone filter values when enter is pressed.

Two filter values are initialized as shown in the following example:

```
<schema>
   <bo/>
   <uiMap/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap" oraSearch="F1-UISRCH"
oraSearchField="MAP_CD; BUS_OBJ_CD:bo;"></td>
    </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
   <bo/>
   <uiMap/>
</root>
</xml>
</html>
```

Mapping Returned Search Fields

This optional attribute is used to direct values returned by the search zone. Multiple fields may be specified. This attribute can only be used in conjunction with the oraSearch attribute.

Syntax	Valid Values	Field Value Options	Comments
`oraSearchOut=" "`	One or more pairs of field name and field value separated by colon. Each pair is separated by a semi-colon. `oraSearchOut="field name:xpath target; ..."` The field name is used to identify the search result returned from the query zone. The field name must match the ELEMENT_NAME mnemonic defined within the explorer zone's search results parameter.	No value XPath	If you do not specify a field value, then the input element containing the oraSearchField attribute receives the returned value. Indicate the XPath to the schema element that should receive the returned value.

> **Note**
>
> If you do not specify an oraSearchOut attribute, the framework will build a default where the field name will be set equal to the oraSearchField's field name.

Two values are returned in the following example:

```
<schema>
   <bo/>
   <mo/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>BO Search </td>
      <td><input oraField="bo" oraSearch="Z1-BOSRCH"
oraSearchOut="BUS_OBJ_CD; MO_CD:mo;"></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
<root>
   <bo/>
   <mo/>
</root>
</xml>
</html>
```

Display Errors

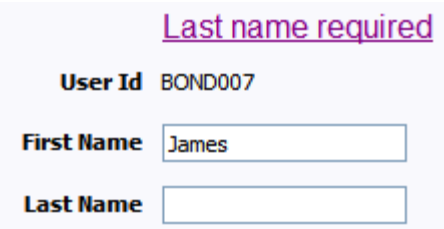
Display Error Variables

One of the following error variables may be displayed.

Syntax
`oraErrorVar="ERRMSG-TEXT"` `oraErrorVar="ERRMSG-LONG"`
`oraErrorVar="ERRMSG-CATEGORY"`
`oraErrorVar="ERRMSG-NUMBER"`

```
...
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
...
```

HTML rendered



Highlight a Field in Error

> **Note**
>
> For more information on throwing an error, refer to the Terminate Statement in the Edit Data Syntax.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Comments
`oraError="automate :true|false; prefix: "`	Specifying `automate:true` automatically enables highlighting of the element in error when issuing an error. Note that **true** is the default and doesn't need to be specified. Specify `automate:false` to turn off field highlighting. The system uses a match on the element name referenced in the error to the element names in the UI map. If the elements in the schema are within an XPath that may not match what is referenced by the error, use `prefix:XPath` to specify that.

> **Note**
>
> The stylesheet reference cisDisabled must be specified for reference of the oraError style. The stylesheet controls how the field in error will be rendered. If you want to alter the rendering you must override the oraError style.

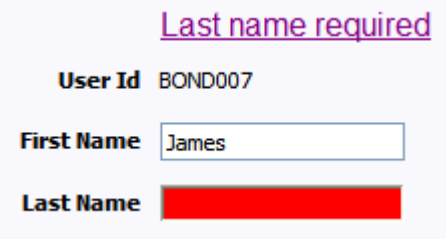
The following HTML example shows that the elements in the map are defined within a group called **boGroup**. The element name returned by the error will not include this group so in order for the field highlighting to work properly the `prefix:` attribute must indicate the group name.

```
<html>
<head>
   <title>User Zone Input</title>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body oraError="automate:true; prefix:boGroup">
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table width="100%" border="0" cellpadding="4">
   <tr style="padding-top:30px;">
      <td align="right" class="label">User Id</td>
      <td>
         <span oraField="boGroup/userId" class="normal"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">First Name</td>
      <td>
         <input oraField="boGroup/firstName" class="normal"/>
      </td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
   <tr>
      <td align="right" class="label">Last Name</td>
      <td>
         <input oraField="boGroup/lastName" class="normal"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <boGroup>
      <userId>BOND007</userId>
      <firstName>James</firstName>
      <lastName></lastName>
   </boGroup>
</root>
</xml>
</html>
```

HTML rendered, where the error element thrown is equal to 'lastName':



Overriding the Error Element Name

In the rare occasion where the element name returned by the error doesn't match the element name in the map, you can add an explicit attribute to indicate the error element name.

Syntax	Valid Values	Comments
`oraErrorElement=`	"element name"	The element name referenced here must exactly match the name of the error element assigned when the error was thrown. More than one HTML field can be referenced by the same error element name.

> **Note**
>
> The stylesheet reference cisDisabled must be specified for reference of the oraError style. The stylesheet controls how the field in error will be rendered. If you want to alter the rendering you must override the oraError style.

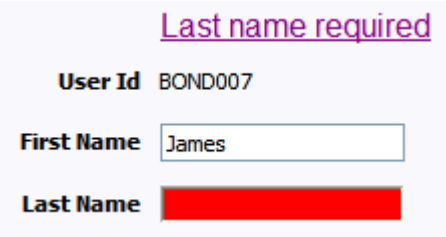
This illustrates a scenario where the element name associated with the error differs from the element name on the map.

```
<html>
<head>
   <title>User Zone Input</title>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</head>
<body>
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table width="100%" border="0" cellpadding="4">
   <tr style="padding-top:30px;">
      <td align="right" class="label">User Id</td>
      <td>
         <span oraField="userId" class="normal"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">First Name</td>
      <td>
         <input oraField="firstName" class="normal"
oraErrorElement="firstName"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">Last Name</td>
      <td>
         <input oraField="familyName" class="normal"
oraErrorElement="lastName"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <userId>BOND007</userId>
   <firstName>James</firstName>
   <familyName></familyName>
</root>
</xml>
</html>
```

HTML rendered.



Display Error Pop-Up

Copyright © 2026, Oracle and/or its affiliates.

When the error text is displayed, this function may be used to pop-up the standard error dialog (which displays more information) when a user clicks the error message.

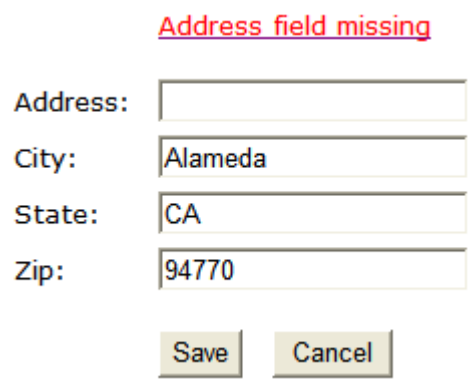
Syntax

```
oraShowErrorAlert(); return false;
```

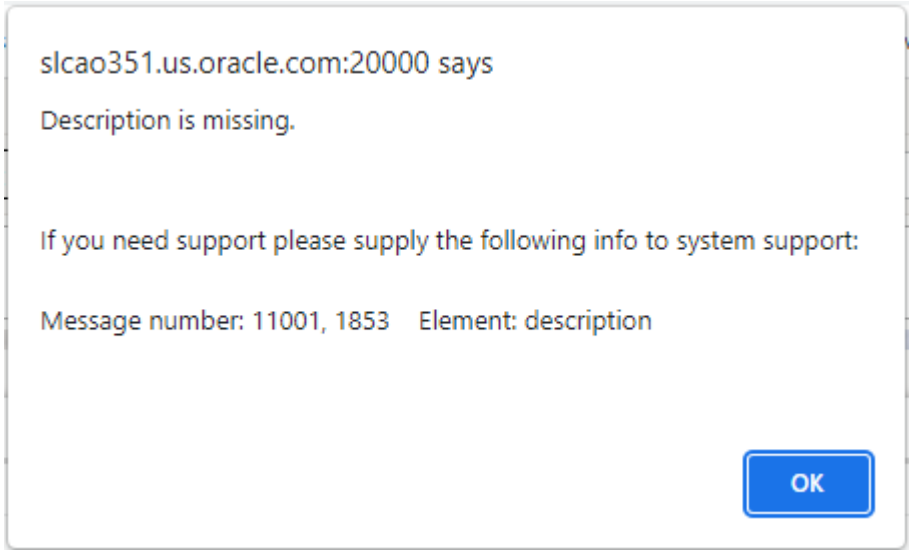
```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table>
   <tr>
      <td >Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
   </tr>
   <tr>
      <td/>
      <td style="padding-top=15px;">
         <oraInclude map="F1-SaveCancelButtons"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.



Standard error pop-up dialog launched via click on error message:



Fire JavaScript for Browser Events

Working with the JavaScript Framework

There are many JavaScript events that can be used within the HTML/Javascript environment. These include events such as onLoad, onBlur, onChange, etc. The UI Map Framework also makes use of some of these events. It is important that any UI Map you develop works with the framework in order to obtain consistent results (events may not always be executed in the same order at all times!).

Copyright © 2026, Oracle and/or its affiliates.

Warning

The following describes the recommended approach for safely handling loading and processing field updates in your UI Maps.

If JavaScript is required within an XHTML UI Map or fragment, it will be necessary to bound it within a `![CDATA[ ]]` tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">
/* <![CDATA[ */

//
// javascript
//

/* ]]> */

</script>
```

Element Change Event

Syntax	Valid Values
`oraChange=" "`	A JavaScript function.

At the time of UI Map load, if there is an event handler already attached to an HTML element, the framework removes it and attaches a combined event handler. The combined handler calls any **framework handler first** and then calls the other (custom) handlers.

Warning

Note that the function must not be used to execute logic that will modify the associated field data value again - or an endless loop will occur.

In the following example the **oraInvokeBS** function is executed when the button is clicked.

```
<html>
  <head>
    <title>oraInvokeBS test</title>
  </head>
  <body>
    <table>
      <tr>
        <td>User Id:</td>
        <td>
          <input oraField= "xmlGroup/userId"/>
          <input type="button" value="Search"
oraChange="oraInvokeBS('UserSearch','xmlGroup');"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          </td>
        </tr>
        <tr>
          <td/>
          <td>Search Results</td>
        </tr>
        <tr>
          <td/>
          <td>
            <table oraList="xmlGroup/searchList">
              <tr>
                <td><span oraField="userId"></span>
                </td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
    </body>
    <xml>
      <root>
        <xmlGroup>
          <userId/>
          <searchList>
              <userId></userId>
          </searchList>
        </xmlGroup>
      </root>
    </xml>
</html>
```

Page Load Event

Syntax	Valid Values
`oraLoad=" "`	A JavaScript function.

Warning

When executing oraLoad within a fragment UI map, and you need to execute a JavaScript function during page load (where the function invokes a business object, business service, or service script) you can use the special syntax "`oraLoad[$SEQUENCEID]`". For other special syntax used for map fragments, refer to theConstruct Portal Zone Map Fragments section.

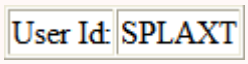
- In the following example, the oraDisplayNone function is executed during page load:

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td oraLoad="oraDisplayNone(item,'userId','')">User Id:
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </td>
            <td><span oraField="userId"></span></td>
        </tr>
    </table>
    </body>
    <xml>
    <root>
        <userId>SPLAXT</userId>
    </root>
    </xml>
    </html>
```

- HTML rendered



After Page Load Event

Syntax	Valid Values
`oraAfterLoad=""`	A JavaScript function.

In the following example the oraGetValueFromScript function is executed after page load.

```
<div>
  <label for="boGroup_parameters_externalSystem" oraLabel="boGroup/
parameters/externalSystem">
  </label>
    <span>
      <select oraSelect="bs:F1-RetrieveExternalSystems"
class="oraInput"
        id="boGroup_parameters_externalSystem" oraField="boGroup/
parameters/externalSystem"
        oraSelectOut="valuePath:results/externalSystem;
descPath:results/description"
        oraSelectIn="outboundMsgType:boGroup/parameters/
outboundMsgType"
        oraAfterLoad

="oraGetValueFromScript(document.getElementById('boGroup_parameters_ext
ernalSystem');">
      </select>
    </span>

</div>
```

Hide Elements

Hide Using a Function

The system provides multiple techniques for hiding an HTML element.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Valid Values	Comments
`oraDisplayNone(item );`	(item, 'XPath', 'value', 'operator')	Used to hide an element based on the value of another element (referenced using its XPath). Enter a value of ' ' to interrogate a blank value. By default the operator is `'='`. This may be set instead to another operator such as `'!='`, `'>'`, or `'<'`.
	(item, function name,**true** | **false**)	Used to indicate a JavaScript function, which must return a Boolean.
`oraIsSuppress="true"`		Use this option for fields that need to be hidden always and not dependent on something else. Set this attribute on the HTML element (div element).

- Example where the User Id label is hidden when no User Id value exists.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td oraLoad="oraDisplayNone(item,'userId','')">User Id: </td>
        <td><span oraField="userId"></span></td>
    </tr>
</table>
</body>
<xml>
<root>
    <userId></userId>
</root>
</xml>
</html>
```

- Example where the Save button is hidden when the user does not have security access to the action of change ('C') for the application service 'F1-DFLTS'.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
   <td oraLoad="oraDisplayNone(item, oraHasSecurity('F1-DFLTS', 'C'),
false );">
      <input name="Save" type="button" onclick="oraInvokeBO('CM-
IndividualTaxpayer', null, 'replace')"/>
   </td>
    </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
    <userId></userId>
</root>
</xml>
</html>
```

Check User's Security Access

The system provides two functions to check a user's security access to a given application service and access mode. These are commonly used for hiding elements.

Syntax	Parameters
`oraHasSecurity( )`	'Application Service code' 'Access Mode'
`oraHasSecurityPath('x','y')`	'Application Service XPath' 'Access Mode XPath'

See the previous section for an example of the `oraHasSecurity` function. The following shows an example where the status button is hidden when the user does not have security access to the access mode 'ACT' of the application service 'FORMTST'.

```
<html>
<body>
<table>
    <tr>
   <td oraLoad="oraDisplayNone(item, oraHasSecurityPath('appService',
'accessMode'), false );">
      <input oraField="statusLabel" type="button"
onclick="oraRunScript('UpdateState','status');"/>
   </td>
    </tr>
</table>
</body>
<xml>
<root>
    <status>ACTIVATE</status>
<statusLabel>Activate</statusLabel>
<appService>FORMTST</appService>
    <accessMode>ACT</accessMode>
</root>
</xml>
</html>
```

Invoke Schema Based Services

The system provides functions for invoking a business object, business service or service script.

Invoke BO Function

This function is used to perform a BO interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
`oraInvokeBO( )`	'BO Name'	
	'XPath' or **null**	Identifies a group element via XPath. If you specify the word **null**, then the entire embedded XML object will be passed.
	'action'	Indicate the action to use. Valid values are **add**, **delete**, **read**, **update**, **replace**, **fastAdd** and **fastUpdate**.

NOT_SUPPORTED

Refer to BO Actions for more information about the various BO actions.

Example with the statement invoked in a JavaScript function.

```
function invokeBO {
     if (!oraInvokeBO('F1-User','xmlGroup','read')) {
          oraShowErrorAlert();
          return;
}
}
```

Example with the statement invoked within onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeBO('F1-User');"/>
```

Invoke BS Function

This function is used to perform a business service interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.

Syntax	Parameters	Comments
`oraInvokeBS( )`	'BS Name'	
	'XPath' or **null**	Identifies a group element via XPath. If you specify the word **null**, then the entire embedded XML object will be passed.

Example with the statement coded within a JavaScript function.

```
function invokeBS {
     if (!oraInvokeBS('F1-UserSearch','xmlGroup')) {
          oraShowErrorAlert();
          return;
  }
}
```


Copyright © 2026, Oracle and/or its affiliates.

Example with the statement invoked via onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeBS('F1-RetrieveInfo');"/>
```

Invoke SS Function

This function is used to perform a service script interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.

Syntax	Parameters	Comments
`oraInvokeSS( )`	'Service Script Name' 'XPath' or **null**	Identifies a group element via XPath. If you specify the word **null**, then the document belonging to the parent node will be passed. If the parent node is not enough, then the entire document can always be passed using the following syntax: `oraInvokeSS('service script', null, null, [$SEQUENCEID])`

Example with the statement invoked within a JavaScript function:

```
function invokeSS {
     if (!oraInvokeSS('F1-GetUsers','xmlGroup')) {
          oraShowErrorAlert();
          return;
}
}
```

Example with the statement invoked within onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeSS('F1-GetUserInfo');"/>
```

Refresh a Rendered Map or Portal Page

Refresh Map

This function is used to 'Refresh' only the map zone issuing the command.

Syntax
`oraRefreshMap;`

```
...
    <tr>
        <td/>
        <td><input type="button" onclick="oraRefreshMap();"
value="Refresh"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </tr>
...
```

Refresh Page

This function is used to refresh all zones in the portal.

Syntax
`oraRefreshPage;`

```
...
    <tr>
        <td/>
        <td><input type="button" onclick="oraRefreshPage();"
value="Refresh"/></td>
    </tr>
...
```

Embed Framework Navigation

Navigate using Navigation Option

This function is used to navigate to another page using the information defined on a navigation option.

Syntax	Parameters
`oraNavigate( );`	'Navigation Option code' 'Key XPath'

> **Warning**
>
> This function is only intended for a UI map defined within a portal zone. It should not be used within a UI map launched by a BPA script.

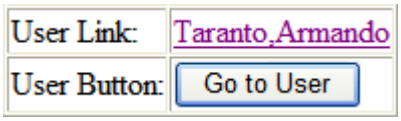
The following example exhibits two possible uses of this function: as a URL and as a button. Note that the UI Map schema must contain a fkRef attribute. Refer to FK Reference Formatting for more information.

```
<schema>
    <userId fkRef="CI_USER"/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
       <td>User Link: </td>
       <td><a href="" onclick="oraNavigate('userMaint','userId'); return
false;">
         <span oraField="userId" oraType="fkRef:CI_USER"></span></a>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      </td>
    </tr>
    <tr>
        <td>User Button: </td>
        <td><input type="submit" onclick="oraNavigate('userMaint','userId')"
          value="Go to User"/></td>
    </tr>
</table>
</body>
<xml>
<root>
    <userId>SPLAXT</userId>
</root>
</xml>
</html>
```

HTML rendered.



Launch a View Details Drawer

Launch a drawer to view additional details for a specific context.

This function is used for a specific element to launch a side panel Drawer to present related extended information. The launched drawer references a panel name defined by the current portal and that may include one or more zones.

A drawer may be launched from a display zone or a row in a data explorer zone.

> **Warning**
>
> This function is only intended for a UI map defined within a portal zone. It should not be used within a UI map launched by a BPA script. Launching a drawer is also not supported from the Sidebar nor the Assistant portals.

Syntax	Parameters in UI Map	Parameters in Data Explorer	Comments
Add the following attribute to an HTML element `viewDetails="..."`	`detailPanelName:LITERAL | x[xpath] | c[CONTEXTNAME];`	`detailPanelName:LITERAL | Cx ;`	You must supply a reference to either a panel name defined for the current portal or a zone that is not defined on the portal.

> **Note**
>
> When referencing a panel name, the

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters in UI Map	Parameters in Data Explorer	Comments
	`detailZoneCode:LITERAL | x[xpath] | c[CONTEXTNAME];`	`detailZoneCode:LITERAL | Cx ;`	drawer width is explicitly defined on the portal for that panel. When referencing a zone, drawer width is determined by the zone's width. Use a 1/3 width for a small drawer, a 2/3 width for a medium drawer, and a full width for a large drawer. From a UI map, the panel name or zone may be specified as a literal without any quotes, as an xpath to an element in the schema or a reference to a global or portal context name. From a data explorer column the non-literal reference may be to another column.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters in UI Map	Parameters in Data Explorer	Comments
	`CONTEXTNAME: LITERAL | x[xpath] | c[CONTEXTNAME];` The context name can also be a reference to an element `x[xpath]`	`CONTEXTNAME: LITERAL | Cx;`	Optionally provide values to one or more portal context names that should be set before the drawer is launched. From a UI map, the value may be specified as a literal, an xpath, or a context name. From a data explorer column the non-literal reference may be to another column.
	`showPanelIcon :true|false;`		By default, including the "view details" attribute on an HTML element also renders a designated icon for launching the drawer. When the launching of the drawer should be made from the HTML element itself, use this parameter to request that the icon is not shown. For example, you can turn the icon off when launching the drawer should be from a "View more information" link element.

This attribute can be added to most standard HTML elements (, , etc.) and the new icon, if not disabled, will be presented after the specified element. The following examples shows how to launch a drawer with and without the standard icon and how to provide values of various types.

```
<span oraField="user" id="user" viewDetails="detailPanelName:VIEWDET;
USERID:x[user];"/>
<span oraMdField="MORE_DETAIL" viewDetails="detailPanelName:x[panelName];
showPanelIcon:false; UOM:c[UOM];"/>
<span oraField="user" id="user" viewDetails="detailZoneCode:DETLSZN;
USERID:x[user];"/>
```

Launch a drawer via a function

The system also provides a function **oraViewDetails** that may be used to launch a drawer. This supports cases where the portal name or the context name needs to be derived. It also supports the case where an object may not have a drawer defined. It may be that if a drawer is defined then launch it, otherwise, navigate to a related page. The syntax is the same as the viewDetails syntax for a UI map. Refer to the UI map F1-LaunchDrawer for an example.

Launch BPA Script

Syntax	Parameters	Comments
`oraRunScript( );`	'BPA script code'.	
	'XPath Element'	One or more element values may be passed into the BPA where it may be referenced as temporary variables.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> This function is only applicable to UI maps displayed in portal zones. UI maps launched within a running BPA script cannot directly launch another BPA script from the UI map's HTML. Instead, return a value from the UI map and execute a Perform Script or Transfer Control step type.

> **Note**
>
> It is incumbent on the script author to pull information out of temporary storage in the initial steps of the script.

In the following example, a temporary variable named 'personId' is created with value '1234567890' and the BPA script named 'Edit Address' is launched.

```
<html>
<body>
<table>
   <tr>
      <td>
         <div oraField="address"></div>
         <span oraField="city"></span>
         <span>,</span>
         <span oraField="state"></span>
         <span oraField="zip"></span>
         <span oraField="country"></span>
         <a href="" onClick="oraRunScript('Edit
Address','personId');">edit</a>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <personId>1234567890</personId>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```

HTML rendered.

123 Main St
Alameda, CA 94770 edit

Launch BPA Script With Values

This function is used to launch a BPA, providing name/value pairs to push into temporary storage. Multiple values can be passed. The BPA script can then reference the temporary variables by name.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
`oraRunScriptWithValues( );`	'BPA script code'. 'XPath Element Name':value	 One or more pairs of element names and values.

> **Note**
>
> You would use this JavaScript function, instead of oraRunScript, if you need to push values to the BPA script that are not located in the UI Map's XML structure.

In the example below, a JavaScript function named 'editUser()' is responsible for launching the BPA script named 'UserEdit'. The temporary variable named 'userId' will be created with value 'CMURRAY'.

```
<html>
<head>
<script type="text/javascript">

function editUser() {
    var values = {'userId': 'CMURRAY'};
    oraRunScriptWithValues('UserEdit', values);
    return;
}

</script>
</head>
<body>
...
</body>
</html>
```

Exit UI Map with Bound Values

This function is used to exit a UI Map. When you quit the map you can specify a value to return to the script and, in addition, whether to return updated XML.

Syntax	Parameters	Comments
`oraSubmitMap( );`	'Return Value' Boolean value	 Indicates if the updated XML should be returned. Default is **true**.

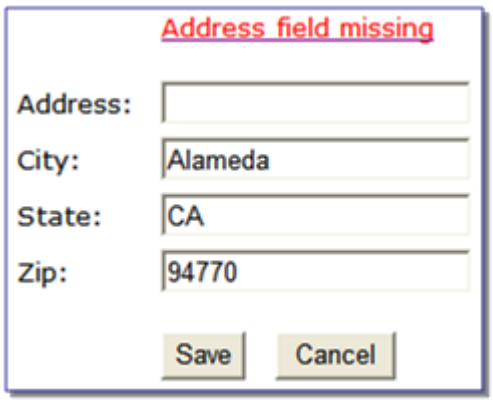
In the following example, the Save button will return updated information, the Cancel button will not.

```
<html>
<body>
<table>
   <tr>
      <td/>
      <td style="padding-bottom:15px;">
         <a href="" onclick="oraShowErrorAlert(); return false;">
```


Copyright © 2026, Oracle and/or its affiliates.

```
         <span oraErrorVar="ERRMSG-TEXT"></span></a>
      </td>
   </tr>
   <tr>
      <td >Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
   </tr>
   <tr>
      <td/>
      <td style="padding-top=15px;">
         <input type="button" value="Save" onClick="oraSubmitMap('SAVE');"/>
         <input type="button" value="Cancel"
onClick="oraSubmitMap('CANCEL',false);"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```

Save and Cancel buttons rendered:



Copyright © 2026, Oracle and/or its affiliates.

Include a Map Fragment

This function is used to embed a map fragment within another UI map. Note that it is possible to use the include node within a map or a map fragment.

Syntax	Parameters	Comments
`<oraInclude map=' ' prefixPath=' '/>`	map='UI Map Code' prefixPath='Xpath'	Optionally specify an xpath prefix to be appended onto every included oraField, oraLabel, oraList, oraSelect valuePath and descPath, oraDownloadData, and oraUploadData attribute value defined within the included UI map fragment's HTML.

> **Note**
>
> This functionality only applies to XPath attribute values when those values do not appear beneath an oraList attribute. Any XPath value within a table containing an oraList attribute will not be affected by a prefixPath.

- An example of a map fragment with two buttons, named 'F1-SaveCancelButtons'.

```
<input onClick ="oraSubmitMap('SAVE');" oraMdLabel="SAVE_BTN_LBL"
class="oraButton"
type="button"/>
<input onClick ="oraSubmitMap('CANCEL',false);" oraMdLabel="CANCEL_LBL"
class="oraButton"
type="button"/>
```

- An example of a map that includes the map fragment named 'F1-SaveCancelButtons'.

```
...
<tr>
   <td colspan="2" align="center">
<oraInclude map="F1-SaveCancelButtons"/>
   </td>
</tr>
...
```

Include a Web Component

A web component is a custom HTML element that produces a specific user interface feature. Each supported web component typically has one or more defined HTML attributes that affect how the web component functions and how it appears. The more complex web components may have a corresponding system configuration object that defines the behavior and information displayed by the resulting user interface element. For example the web component

Copyright © 2026, Oracle and/or its affiliates.

that produces a "tree" of information has a Tree configuration table, which defines the details about how to retrieve information for the tree and how to display each node.

Common Attributes

Some common attributes that may be used for each web component are:

- A reference to the specific instance of the configuration object that governs the details to produce.
- The ability to pass in context data as name and value pairs.
- A 'mode' that support showing a preview of the feature

Some attributes support a **single value** that is referenced. For these types of attributes, the value may be provided using one of the following references:

- Literal value. The value can be referenced directly. Example:

  ```
  treename="F1-Tree"
  ```

- XPath Reference. XPath references must use the explicit syntax `x[value]`. Example:

  ```
  treename="x[migrationTree]"
  ```

- Global or Portal context reference. These references must use the explicit syntax `c[value]`.

  ```
  treename="c[TREE_NAME]"
  ```

- Field label reference. These references must use the explicit syntax `md[value]`.

  ```
  text=md[F1_ADD_LBL]"
  ```

- Field help text reference. These references must use the explicit syntax `mdh[value]`.

  ```
  text=mdh[F1_OPERATIONS_LBL]"
  ```

For **complex attributes** that support one or more name / value pairs, the syntax for defining the names and values follows the same rules as above for literal values, XPath references and context. Colons (":") should be used to separate the name from the value and semi-colons (";") separate each pair.

```
context="FIELD_NAME1:x[value1];FIELD_NAME2:F1AB;x[fieldName3]:c[FIELD_NAME3];"
```

A given web component may have additional HTML attributes specific to its functionality.

> **Note**
>
> Any UI map that defines a web component must include the F1-OJETLIBSR map fragment within the "body" tag.

```
<body>
<oraInclude map="F1-OJETLIBSR"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
..
</body>
```

> **Note**
>
> Standard HTML attributes may also be provided (e.g ID=".." and style=".."). These are defined as normal. It should be noted that the web components themselves may control some styling aspects directly and thus override any style settings provided on the HTML web component. For example, the Contextual Insights configuration provides settings for text color that would override any color defined in the HTML.

The following sections describe each supported web component.

Trees

Refer to Trees for more information about tree configuration.

Syntax	Parameters	Comments
`<ou-tree...></ou-tree>`	`treename="singleValue"`	You must supply a reference to the name of the tree object that defines the configuration of the tree to display. This attribute expects a single value that follows the syntax described in the single value section above.
	`context="..."`	Optionally pass in one or more name / value pairs. The syntax follows the example shown in the complex attributes information above.

The following example uses an explicit reference to a tree and does not pass any context. The particular tree relies on global context, which the tree node algorithm has access to.

```
<ou-tree treename="C1-PremiseTree"></ou-tree>
```

Contextual Insight

Refer to Contextual Insights for information about the configuration related to defining contextual insights.

Syntax	Parameters	Comments
`<ou-insights ...></ou-insights>`	`insightType="singleValue"`	Use this attribute to specify an insight type to display. Either this attribute or the insight class attribute must be supplied. This attribute expects a single value that follows the syntax described in the single value section above.
	`insightClass="singleValue"`	Use an insight class to display all the insight types for the insight group related to this class. This attribute expects a single value that follows the syntax described in the single value section above.
	`context="..."`	Optionally pass in one or more name / value pairs. The syntax follows the example shown in the complex attributes information above.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
	`mode="preview"`	Optionally pass in a mode of **preview** to indicate to the underlying insight algorithm that prepares the data that the insight is being called in preview mode, with no context available. Ideally the algorithm returns sample information.
	`class=" "`	Optionally provide a CSS helper class to control aspects of the rendered insight. Supported values: `class="card-full-width"` - Applicable for card insights. Use this option class to ensure that the width for a card insight adjusts to the width of the zone that contains it. Card insights that don't set this parameter use a fixed width.

The following example is from the insight preview map where the insight type is the one in context. No context data is needed and the mode is 'preview'.

```
<ou-insights insightType="x[insightType]" mode="preview"></ou-insights>
```

The following example uses an insight class and passes in context values from XPath elements with explicitly defined context names.

```
<ou-insights insightClass="C1CI"
context="PER_ID:x[personId];ACCT_ID:x[accountId];PREM_ID:x[premiseId];"></ou-
insights>
```

Visualization Widgets

This web component may be used to render various types of visualization widgets such as cards and badges.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
`<ou-visualization .. .></ou-visualization>`	`type="LITERAL | x[xpath]"`	You must specify the type of widget to render. The valid literal values are: • **`badge`** - a non-clickable badge widget • **`card`** - A card provides a quick view on an issue or a task. At a minimum, it includes a primary text describing the highlighted topic. It may also include optional components: secondary and tertiary lines of text, a badge, a over-line text adjacent to the badge, an icon and action buttons. • **`message`**- a message text along with an optional icon. The type may be specified by a reference to an element that contains one of the valid values.

	`color="`
	`badge:x[xpath]|LITERAL;`
	`icon:x[xpath]|LITERAL;"`
		Complex attribute that sets the color option to use for various aspects of the widget.
		Note that the reference is not a color option and not a specific color.
		The **`badge:`** attribute is applicable to a **badge** and a **card**.
		The **`icon:`**attribute is applicable to a **card** and a **message**.

	`imgRef="x[xpath]| LITERAL"`	References an icon defined as a Display Icon. Applicable to a **card** and a **message**.
	`imgSize="x[xpath]| LITERAL"`	May be used to override the standard size of the icon. Valid values are: s (small), m (medium) and l (large). Applicable to a **card** and a **message**.

	`text="`
	`badge:x[xpath]|LITERAL;`
	`overline:x[xpath]| LITERAL;`
	`main:x[xpath]|LITERAL;`
	`secondary:x[xpath]| LITERAL;`
	`tertiary:x[xpath]| LITERAL;`
		Applicable to all types of visualization widgets.
		`badge:` is text to use on a badge.
		`main:` is the main text to use on the widget.
		`secondary:` and `tertiary:`are optional 2nd and 3rd text lines on a **card**.
		`overline:`is text next to the badge on a **card**.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
	`actions="x[xpath]"`	This is an xpath reference to a list element describing actions that may be available to the user as part of the widget. Applicable to a **card** widget. The list element must be defined as follows: `<somelist type="list"> <includeDA name="F1-ActionDetails"/> </somelist>`

The following is an example of a badge.

```
<ou-visualization type="badge" text="badge:x[text];"
color="badge:x[colorOption];"></ou-visualization>
```

The following is an example of a card with only a primary text.

```
<ou-visualization type="card" text="main:x[main];"></ou-visualization>
```

The following is an example of a card with more components.

```
<ou-visualization type="card"
text="badge:x[text];main:x[mainText];secondary:x[secondaryText]"
color="badge:x[badgeColor];icon:x[iconColor]" imgRef="x[icon]"
actions:x[actions]></ou-visualization>
```

The following is an example of a message with an icon.

```
<ou-visualization type="message" text="main:x[mainText];"
color="icon:x[iconColor]" imgRef="x[icon]"></ou-visualization>
```

Note the following when designing your actions and populating the actions list:

- An action may be of the following types:
  - Navigate to page. The 'action key' in the action's list entry is the navigation option to navigate to.
  - Launch a BPA script. The 'action key' in the action's list entry is the script to launch.
  - Launch a view details panel in a drawer. The 'action key' is the panel name defined on the current portal.
  - Launch a Knowledge Article. The 'action key' is the knowledge article name as defined in the Knowledge Management Articles extendable lookup.
- Context details may be provided for any action as needed.
- By default, actions are rendered as buttons, except for the action to launch a knowledge article that is rendered as the knowledge book icon.

Copyright © 2026, Oracle and/or its affiliates.

- An action may optionally be associated with a role:
  - One action may be set with a **primary** role to highlight it as the main action. This is applicable to redwood view mode only.
  - If the card itself is clickable then one of the actions needs to be set with the **card** role. This action is excluded from the list of buttons.

Knowledge Article

This web component may be used to associate a UI location with a knowledge article.

Refer to Knowledge Articles for information about knowledge article configuration.

Syntax	Parameters	Comments
`<ou-km-link ... ></ou-km-link>`	`kmCode="LITERAL"`	Reference the knowledge article code defined in the **Knowledge Management Articles** extendable lookup.

The following is an example of a knowledge icon added to a UI map.

```
<ou-km-link kmCode="extendableLookupCode"></ou-km-link>
```

Button

The button web component provides support for the standard set of button formats used within the system.

Syntax	Parameters	Comments
`<ou-button...></ou-button>`	`type="singleValue"`	You must define the type of button to display. This attribute expects a single value that follows the syntax described in the single value section above. The valid type values are: • `image` - image only • `text` - text only • `imageText` - image followed by text • `menu` - text plus a dropdown caret to open the menu • `imageMenu` - image plus a dropdown caret to open the menu If no type attribute is supplied, a 'broken button' image will be displayed.
	`img="singleValue"`	Define the path of the image to display. Either this attribute or the image reference attribute must be supplied for a button type that includes an image. This attribute expects a single value that follows the syntax described in the single value section above. Note that for SVG images, if the reference includes only the SVG file name with no path, the path is assumed to be the Framework SVG repository.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
	`imgRef=”singleValue”`	Define the display icon reference of the image to be used. Either this attribute or the image attribute must be supplied for a button type that includes an image. This attribute expects a single value that follows the syntax described in the single value section above.
	`text="..."`	Define the various text elements needed by the button. The syntax follows the example shown in the complex attributes information above. The valid text element name values are: • `label` - the label text to be used for the button • `tooltip` - the hover text to be used for the button • `aria-label` - the ARIA label for the button
	`menu=”singleValue”`	Define the menu to be used with the button. This attribute must be supplied for a button type that includes a menu control. This attribute expects a single value that follows the syntax described in the single value section above.
	`disabled=”singleValue”`	This is a standard HTML attribute. A value of "true" or just the presence of the text "disabled" will result in an initially disabled button.
	`security="..."`	Optionally define the security settings for the button. The button will be hidden if the user does not have the appropriate security access. The syntax follows the example shown in the complex attributes information above. The valid security element name values are: • `service` - the application service whose security access governs access to the button • `accessMode` - the specific access mode for the service that determines a more granular level of access to the button

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
	`clickAction="..."`	Optionally define the action to be taken when the button is ‘clicked’. The syntax follows the example shown in the complex attributes information above. There are specific ‘name’ values for the attribute as follows: • `action` - the action to be taken. Valid values are ‘BPA’ and ‘NAVIGATE’ • `script` - the script to be performed when the action is ‘BPA’ • `navopt` - the navigation option to be used when the action is ‘NAVIGATE’ The attribute may also include other name/value pairs that define information to be passed into the script or used as navigation context fields. In addition to the security attribute settings, the system will check that the user has access to the script or navigation path before performing the action. The standard HTML ‘onClick’ attribute may still be used. if the HTML includes an ‘onClick’attribute as well as the ‘clickAction’, the ‘onClick’ attribute will take precedence

The following is an example of a simple button with text.

```
<ou-button type="text" text="label:md[F1_BOOKMARK_LBL];"></ou-button>
```

The following is an example of an icon button using a display icon.

```
<ou-button type="image" imgRef="F1CMNT"></ou-button>
```

The following is an example of an image menu button.

```
<ou-button type="imageMenu" imgRef="F1STTGS" menu="CI_CONTEXTSCRIPT"
text="tooltip:md[F1INSTY_VIS_LBL];></ou-button>
```

The following is an example of text button that navigates to another page.

```
<ou-button type="text" text="label:md[ALG_CD];tooltip:md[GOTO_ALG]"
clickAction="action:NAVIGATE; navopt:f1algqTabMenu; ALG_CD:x[algorithm];></ou-
button>
```

Decorator

The decorator web component may be used to render an icon for decoration purposes only. This is not an actionable item.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Parameters	Comments
<ou-decorator...></ou-decorator>	img=”singleValue”	Refer to ou-button syntax.
	imgRef=”singleValue”	Refer to ou-button syntax.

The following is an example of a decorator icon using a display icon.

```
<ou-decorator imgRef="F1CMNT"></ou-decorator
```

Show Schema Default on Add

Default values within in the UI map's schema will be displayed on a UI map's input fields if an embedded <action> node has a value of 'ADD' or blank.

Syntax
<action>ADD</action>
<action> </action>

The schema default for the <description> element will be displayed:

```
<schema>
    <action/>
    <boGroup type="group">
        <key/>
        <description default="enter description here"/>
    </boGroup>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td>Description   </td>
        <td><input oraField="boGroup/description"></td>
    </tr>
</table>
</body>
<xml>
<root>
    <action>ADD</action>
    <boGroup>
        <key/>
        <description/>
    </boGroup>
</root>
</xml>
</html>
```

HTML rendered.

Copyright © 2026, Oracle and/or its affiliates.



Configure a Chart

In addition to integrating charts with zones, the system supports using the same attributes within the UI Map HTML to configure a graphical representation of an XML list. The designer can control the type, size, position, and contents of the chart using these attributes. For details of the attributes and syntax, refer to Data Explorer Graph Configuration.

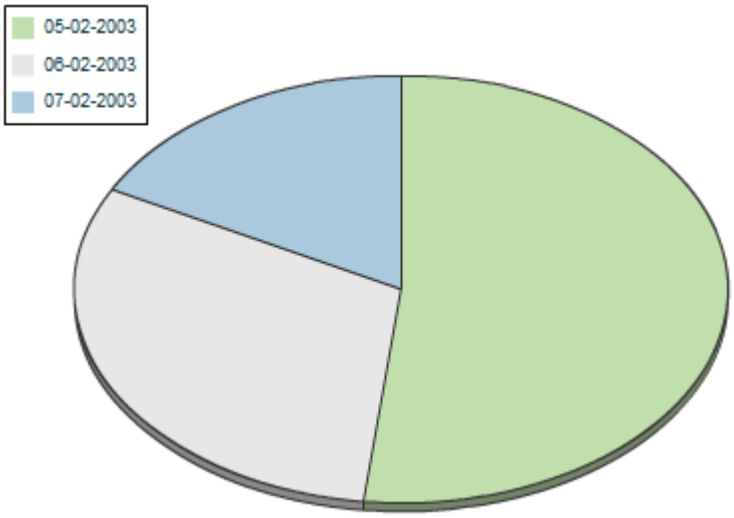
Graph Examples

- Sample of a pie chart configuration:

```
<html>
<head>
<title>Pie Chart</title>
</head>
<body>

<div style="width:100%; height:290px;"
    oraChart="type:pie;"
    oraChartSeries1="list:set; labelPath:date; amount:amount; "
    oraChartBroadcast="BILL_ID:billId;">
</div>

</body>

<xml>
<root>
  <set>
<date>05-02-2003</date>
<amount>163.24</amount>
<billId>592211649441</billId>
  </set>
  <set>
<date>06-02-2003</date>
<amount>97.29</amount>
<billId>592211649442</billId>
  </set>
  <set>
<date>07-02-2003</date>
<amount>54.38</amount>
<billId>592211649443</billId>
  </set>
</root>
</xml>
</html>
```

- A pie chart rendered for a single series:

Copyright © 2026, Oracle and/or its affiliates.



- Sample of a line, cluster, or stacked graph configuration - each with two series:

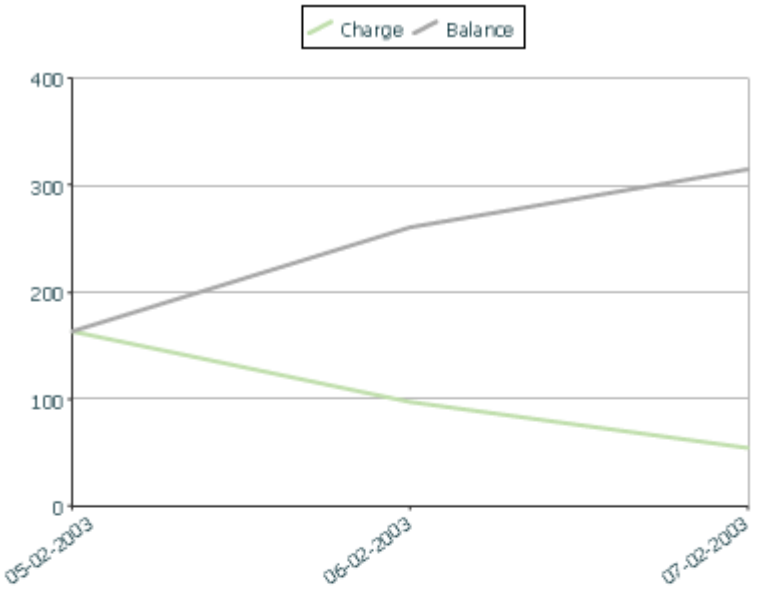
```
<html>
<head>
<title>Stacked Chart</title>
</head>
<body>

<div style="width:100%; height=300px;"
    oraChart="type:line;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
    oraChartBroadcast="BILL_ID:billId;">
</div>

<div style="width:100%; height=300px;"
oraChart="type:cluster;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
oraChartBroadcast="BILL_ID:billId;">
</div>

<div style="width:100%; height=300px;"
oraChart="type:stacked;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
oraChartBroadcast="BILL_ID:billId;">
</div>

</body>

<xml>
<root>
  <set>
<date>05-02-2003</date>
<amount>163.24</amount>
<balance>163.24</balance>
<billId>592211649441</billId>
```

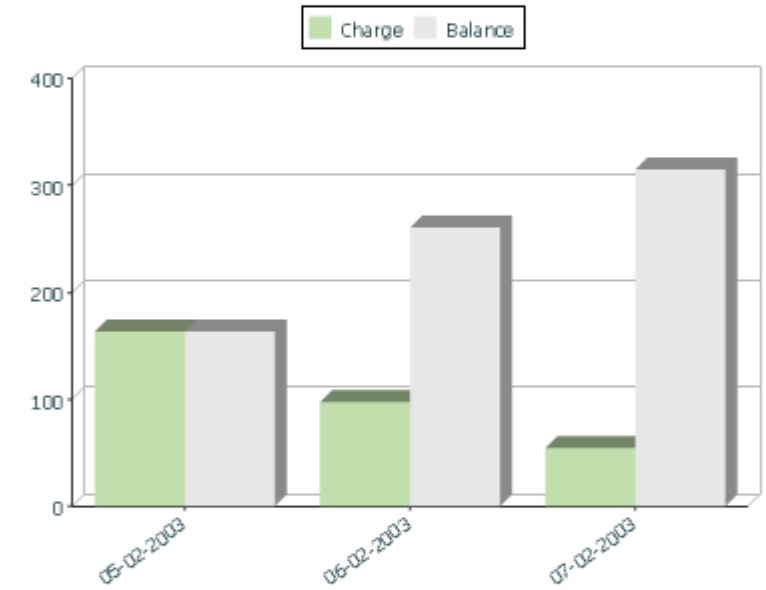

Copyright © 2026, Oracle and/or its affiliates.

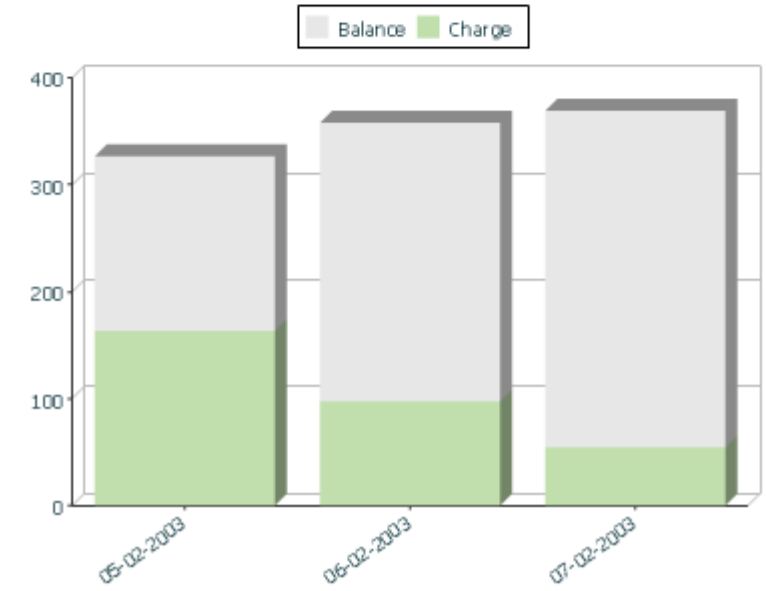
```
  </set>
  <set>
<date>06-02-2003</date>
<amount>97.29</amount>
<balance>260.53</balance>
<billId>592211649442</billId>
  </set>
  <set>
<date>07-02-2003</date>
<amount>54.38</amount>
<balance>314.91</balance>
<billId>592211649443</billId>
  </set>
</root>
</xml>
</html>
```

- Three types of chart rendered for two series each: line, cluster, and stacked.

Copyright © 2026, Oracle and/or its affiliates.







Copyright © 2026, Oracle and/or its affiliates.

Upload and Download a CSV File

The following HTML attributes can be used to manage both an upload and a download between a list defined within the map's schema and a CSV (comma separated value) file. Note that this technique is only recommended for a small to medium volume of data, for example no more than several hundred rows. For higher volumes, it is recommended to use batch upload / download functionality instead.

The syntax is `oraUploadData="type:embed;path:list xpath;useLabels:true;showCount:true"`

Upload configuration requires you to name a CSV file to be uploaded, and an XML list as target. By convention, each CSV row will create a separate list instance. Each comma-separated field in the file will be uploaded as a separate element in the list. To embed an upload dialog within a map, the **oraUploadData** attribute must be associated with a container element such as a div, td, or span.

The optional `useLabels:true` value indicates that while parsing the upload CSV file, the headers are expected to be labels

> **Note**
>
> If you do not specify the `useLabels:true` value and the XML target element name is "camelCase" then the corresponding spreadsheet header should be title case with a space between words, e.g.;"Camel Case". Letters and special characters are not considered a different word, for example Address1 must be uploaded into the target XML element address1.

Specifying the optional `showCount:true` value will display the number of records uploaded.

> **Caution**
>
> If you are using a grid in conjunction with the **oraUploadData** function, then you must maintain the grid's list with a 'replace' business object action. Refer to BO Replace Action for more information.

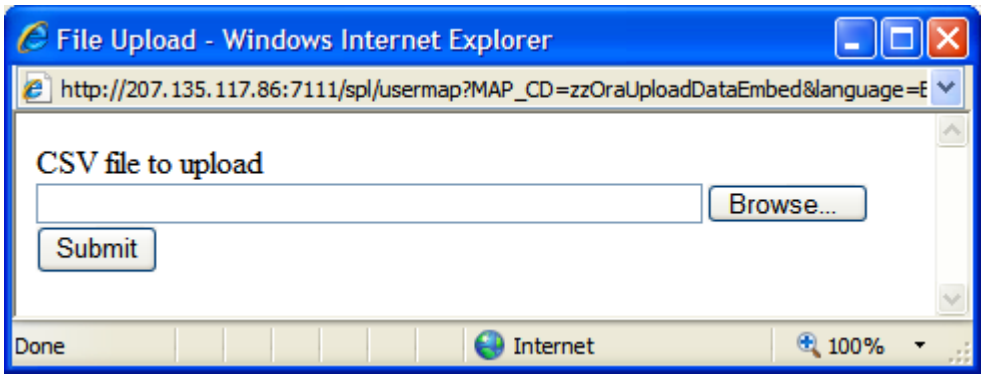
Sample of `oraUploadData="embed"` within a div element.

```
<html>
<head>
    <title>File Upload</title>
</head>
<body>

    <div oraUploadData="type:embed;path:myList"> </div>

</body>

<xml>
<root>
    <myList>
<id>838383930</id>
        <name>Janice Smith</name>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </myList>
    <myList>
<id>737773730</id>
        <name>Bill McCollum</name>
    </myList>
</root>
</xml>
</html>
```

This file upload dialog will be embedded into the body of the page where the oraUploadData is defined.



oraUploadData="type:popup;path:list xpath;useLabels:true;showOk:true;showCount:true"

Upload configuration requires you to name a CSV file to be uploaded, and an XML list as target. By convention, each CSV row will create a separate list instance. Each comma-separated field in the file will be uploaded as a separate element in the list. To upload a CSV file using a pop-up dialog, the oraUploadData attribute must be associated with an input element such as a button, text link, or image.

The optional useLabels:true value is used to indicate that while parsing the upload CSV file, the headers are expected to be labels

> ⓘ **Note**
>
> If you do not specify the useLabels:true value and the XML target element name is "camelCase" then the corresponding spreadsheet header should be title case with a space between words, e.g., "Camel Case". Letters and special characters are not considered a different word, for example Address1 must be uploaded into the target XML element address1.

Specifying the optional showOk:true value will display an "Ok" button once the upload finishes. The popup will stay open until the button is pressed. Additionally, specifying the showCount:true value will display number of records uploaded.

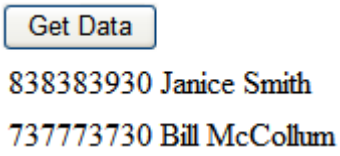
> ⚠ **Caution**
>
> If you are using a grid in conjunction with the **oraUploadData** function, then you must maintain the grid's list with a 'replace' business object action. Refer to BO Replace Action for more information.

Copyright © 2026, Oracle and/or its affiliates.

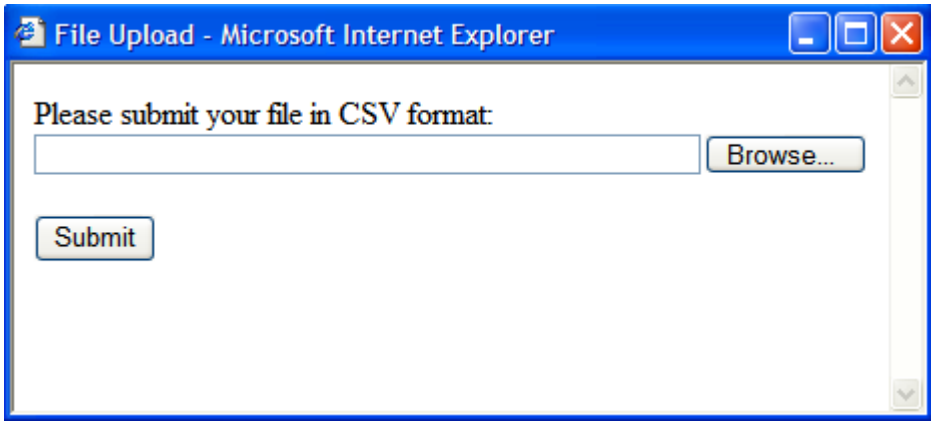
Sample of oraUploadData="popup" associated with a button:

```
<html>
<head>
    <title>File Upload</title>
</head>
<body>
   <input type="button" name="submitButton"
oraUploadData="type:popup;path:myList;" value='Get Data'>
   <table oraList="myList">
      <tr/>
      <tr>
         <td><span oraField="id"/></td>
         <td><span oraField="name"/></td>
      </tr>
   </table>
</body>
<xml>
<root>
   <myList>
      <id>838383930</id>
      <name>Janice Smith</name>
   </myList>
   <myList>
      <id>737773730</id>
      <name>Bill McCollum</name>
   </myList>
</root>
</xml>
</html>
```

HTML Rendered:



Pressing the "Get Data" button will launch a standard file upload dialogue (provided by Framework) as shown below.



oraDownloadData="list xpath"

Copyright © 2026, Oracle and/or its affiliates.

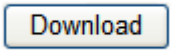
Download configuration requires you to name an XML list to be downloaded. By convention, each list instance will represent a separate row in the created file. By default every element of the list will be comma separated in the file.

> **Note**
>
> The number formatting is based on the user profile setting. For localities where the decimal symbol is a comma, an implementation may configure a property setting (spl.csv.delimiter.useFromDisplayProfile=true) to cause the system to use a semicolon as the delimiter that separates the elements rather than a comma.

Sample of oraDownloadData.

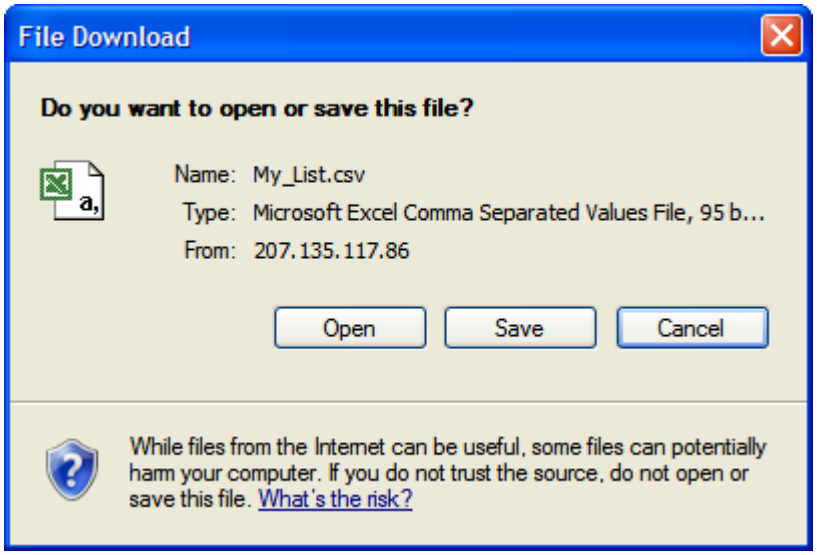
```
<html>
<head>
<title>File Download</title></head>
<body>
<input type="button" name="downloadButton" oraDownloadData="myList"
value="Download"/>
</body>
<xml>
<root>
   <myList>
      <id>881-990987</id>
      <name>John Mayweather</name>
   </myList>
   <myList>
      <id>229-765467</id>
      <name>Anna Mayweather</name>
   </myList>
   <myList>
      <id>943-890432</id>
      <name>Andrew Brewster</name>
   </myList>
</root>
</xml>
</html>
```

HTML Rendered:

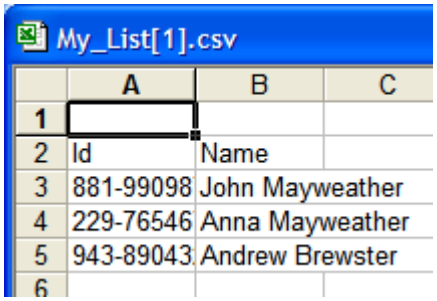


Pressing the "Download" button will launch a standard file download dialogue (provided by Framework) as shown below.

Copyright © 2026, Oracle and/or its affiliates.



A successful download will result in a CSV file:



To download data from a sub list use the attribute oraDownloadDataInList instead of oraDownloadData. The attribute oraDownloadDataInList will have the sub list name. The XPath of the sub list is used to pick data of the specific row from the parent list. Thus only the specific sub list is downloaded.

oraDownloadDataUseLabels="true"

The oraDownloadDataUseLabels attribute can be used in conjunction with the oraDownloadData attribute described above. Specify oraDownloadDataUseLabels if you want the generated CSV file to use the element labels for columns headers rather than element names.

Construct Portal Zone Map Fragments

Portal zones can reference a UI map for the filter area. This UI map is not a complete HTML document, but is instead configured as a UI Map fragment. When constructing a zone map fragment you can reference the following substitution variables. Note that these variables will be dynamically populated at run time with information particular to the map's zone within the portal:

Variable	Replacement Logic
[$ZONEDESCRIPTION]	Zone's description text.
[$SEQUENCEID]	Zone's sequence ID.
[$ZONENAME]	Zone's name.
[$HELPTEXT]	Zone's help text.

Copyright © 2026, Oracle and/or its affiliates.

[$ZONEPARAMNAME]	Zone parameter's value (or blank if it has not been specified).

Warning

- These maps make use of the oraInclude tag to incorporate HTML fragments for the framework actions. Refer to the zone type parameters for the UI Map fragments you should include in your HTML.
- If you wish to have the "help text" icon appear next to your zone description, you should have id="title_[$SEQUENCEID]" on the <td> that contains your description.
- If it is necessary to encapsulate JavaScript within a UI Map fragment, it will be necessary to bound the JavaScript within a `![CDATA[ ]]` tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">

/*<![CDATA[ */

//
//javascript
//
/*]]> */
</script>
```

Note

If you wish to preserve the values of a filter input field, within a filter map fragment, for the framework 'Go Back' and 'Go Forward' functionality, you must associate the input field (text box, select, etc.) with a unique HTML id. Input field values associated with a unique id will be captured in the framework's 'memento'. The 'memento' is used to rebuild the input map when the portal zone is navigated to using the 'Go Back' or 'Go Forward' functionality.

Note

Many specialized functions exist to manipulate zone behavior, for example:

- `oraGetZoneSequence(zoneName)`. Uses the zone's code to retrieve its sequence number.
- `oraIsZoneCollapsed(sequenceId)`. Uses the zone's sequence to determine if collapsed.
- `oraHandleCollapse(seq)`. Collapse a zone.
- `oraHandleExpand(seq,refresh)`. Expand and/or refresh a zone.

All of these, and many more functions, are located within the JavaScript library userMapSupport.js described below.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When executing oraLoad within a fragment UI map, and you need to execute a JavaScript function during page load (where the function invokes a business object, business service, or service script) you can use the special syntax "oraLoad[$SEQUENCEID]". Refer to the Load Page Event section for more information.

Example of oraLoad[$SEQUENCEID] used within a function:

```
<script type="text/javascript">
function oraLoad[$SEQUENCEID]() {
checkRebateClaimStatus();
}

function checkRebateClaimStatus() {
    var work = id(''analyticsFilterText[$SEQUENCEID]'',
document).cells[0].innerText.split('' '');
    var rebateClaimId = work[work.length - 3];
    id(''rebateClaimId'', document).value = rebateClaimId;
oraInvokeSS(''C1-CheckRCSt'',''checkRebateClaimStatus'', false);
    var statusIndicator = id(''statusInd'', document).value;
    if (statusIndicator == ''C1PE'' || statusIndicator == ''C1ID'') {
       id(''addRebateClaimLine'', document).style.display = '''';
    } else {
id(''addRebateClaimLine'', document).style.display = ''none'';
    }
}
</script>
```

Invoking a Business Object

The oraInvokeBO function may be used within a zone filter map. It is similar to the command described in Invoke BO Function which allows for a business object to be invoked within the UI map's HTML. Refer to that section for a description of the first three parameters.

Syntax	Parameters	Comments
`oraInvokeBO( )`	'BO Name' 'XPath' or **null** 'action'	
	`null`	This must be specified as the fourth argument.
	`[$SEQUENCEID]`	This must be specified as the fifth argument.
	`false`	Specify `false`. The 'true' option is no longer applicable.

Example:

```
oraInvokeBO('CM-User','xmlGroup','read', null, [$SEQUENCEID], false)
```

Invoking a Business Service

Copyright © 2026, Oracle and/or its affiliates.

The oraInvokeBS function may be used within a zone filter map. It is similar to the command described in Invoke BS Function which allows for a business service to be invoked within the UI map's HTML. Refer to that section for a description of the first two parameters.

Syntax	Parameters	Comments
`oraInvokeBS( )`	'BO Name'	
	'XPath' or **null**	
	`null`	This must be specified as the fourth argument.
	`[$SEQUENCEID]`	This must be specified as the fifth argument.
	`false`	Specify `false`. The 'true' option is no longer applicable.

Example:

```
oraInvokeBS('CM-UserSearch','xmlGroup', null, [$SEQUENCEID], false)
```

Invoking a Service Script

The oraInvokeSS function may be used within a zone filter map. It is similar to the command described in Invoke SS Function which allows for a service script to be invoked within the UI map's HTML. Refer to that section for a description of the first two parameters.

Syntax	Parameters	Comments
`oraInvokeSS( )`	'Service Script Name'	
	'XPath' or **null**	
	`null`	This must be specified as the fourth argument.
	`[$SEQUENCEID]`	This must be specified as the fifth argument.
	`false`	Specify `false`. The 'true' option is no longer applicable.

Example:

```
oraInvokeSS('UserSearch','xmlGroup', null, [$SEQUENCEID], false)
```

Detecting Unsaved Changes

Use this function to return a Boolean set to true if there are unsaved changes. The system will interrogate the function when the user attempts to navigate and issue a warning accordingly. This function is only needed if a UI map is using custom javascript to manage elements such that the system is not able to detect whether changes have been made. Also note that it's the responsibility of the UI map javascript to manage the values in the Boolean used for this function.

```
function hasUnsavedChanges(){
       return isDirtyFlag;
   }
```

Hiding Portal Tabs

The product provides the ability to use JavaScript to hide a tab on the current portal based on some condition using the oraAuthorizeTab JavaScript API. This API accepts a function as a parameter and turns off the tab index indicated.

Copyright © 2026, Oracle and/or its affiliates.

For example, the UI Map may have a function to turn off one or more tab indexes.:

```
function overrideTabIndex(index){
    if (index == 2) return false;
    if (index == 3) return false;
  }
```

The JavaScript is referenced “on load”:

```
<body class="oraZoneMap"
onLoad="oraAuthorizeTab(overrideTabIndex);">
```

Required JavaScript Libraries

All of the functionality described in this document depends on a pair of JavaScript libraries. If you are writing and executing your maps entirely within the UI map rendering framework - you do not need to manually insert the following libraries - the framework will insert them for you when the UI Map is rendered.

> **Warning**
>
> When executing HTML outside of the framework you must include the following references explicitly within your HTML. In addition, the tool you use to render the HTML must have access to a physical copy of privateUserMapSupport.js for bind support.

```
src="privateUserMapSupport.js"
```

Your HTML document must reference this library to execute binding in a stand-alone environment.

> **Warning**
>
> Referencing functions within this JavaScript library is dangerous - because these functions are owned by framework and they may be changed during version upgrade or via the normal patch process.

```
<script type="text/javascript" src="privateUserMapSupport.js"></script>
```

```
src="userMapSupport.js"
```

To take advantage of optional toolset features, you must reference this library.

> **Note**
>
> You can reference the functions within this JavaScript library to write custom functions within the UI map..

```
<script type="text/javascript" src="userMapSupport.js"></script>
```


Copyright © 2026, Oracle and/or its affiliates.

```
onload="oraInitializeUserMap();"
```

To execute binding in a stand-alone environment, you must embed the following onload function into the <body> node.

```
<body onload="oraInitializeUserMap();">
```

UI Map Standards

Basic UI Map Templates

All UI Maps share the same basic structure regardless of placement (page area, zone, pop-up) or usage (display only, input).

Sample XML

All information in this document is based upon the following XML structure.

```
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
```

Display Only UI Map

```
<html>
<head>
    <title oraMdLabel="ADDRESS_LBL"></title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body class="oraZoneMap">
<table cellspacing="4" width="100%">
    <colgroup>
        <col class="oraLabel oraTableLabel" />
        <col class="oraNormal oraTableData" />
    </colgroup>
    <tr>
        <td oraLabel="address"></td>
        <td oraField="address"></td>
    </tr>
    <tr>
        <td oraLabel="city"></td>
        <td oraField="city"></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </tr>
    <tr>
        <td oraLabel="state"></td>
        <td oraField="state"></td>
    </tr>
    <tr>
        <td class="oraSectionEnd" oraLabel="zip"></td>
        <td class="oraSectionEnd" oraField="zip"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionHeader"
oraMdLabel="CONTACT_LBL"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionStart oraEmbeddedTable">
            <table oraList="contactInformation" cellspacing="2">
                <thead >
                    <tr>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/type></span>
                        </th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/number"></span>
                        </th>
                    </tr>
                </thead >
                <tbody>
                    <tr>
                        <td class="oraNormalAlt oraDisplayCell">
                            <span oraField="type"></span>
                        </td>
                        <td class="oraNormal oraDisplayCell">
                            <span oraField="number"></span>
                        </td>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
</table>
</body>
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

Input UI Map

```
<html>
<head>
    <title oraMdLabel="ADDRESS_LBL"></title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table width="100%" cellpadding="12">
  <tr class="oraErrorText">
    <td><a href="" onclick="oraShowErrorAlert(); return false;">
      <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span></a>
    </td>
  </tr>
</table>
<table cellspacing="4" width="100%">
    <colgroup>
        <col class="oraLabel oraTableLabel" />
        <col class="oraNormal oraTableData" />
    </colgroup>
    <tr>
        <td oraLabel="address"></td>
        <td><input type="text" oraField="address"/></td>
    </tr>
    <tr>
        <td oraLabel="city"></td>
        <td><input type="text" oraField="city"/></td>
    </tr>
    <tr>
        <td oraLabel="state"></td>
        <td><input type="text" oraField="state"/></td>
    </tr>
    <tr>
        <td oraLabel="zip"></td>
        <td><input type="text" oraField="zip"/></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionHeader"
oraMdLabel="CONTACT_LBL"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionStart oraEmbeddedTable">
            <table oraList="contactInformation" cellspacing="2">
                <thead >
                    <tr>
                        <th class="oraGridColumnHeaderButton"></th>
                        <th class="oraGridColumnHeaderButton"></th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/type></span>
                        </th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/number"></span>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                        </th>
                    </tr>
                </thead >
                <tbody>
                    <tr>
                        <td oraType="addGridRow"></td>
                        <td oraType="deleteGridRow"></td>
                        <td>
                            <input type="text" oraField="type"/>
                        </td>
                        <td>
                            <input type="text" oraField="number"/>
                        </td>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
    <tr>
      <td colspan="2" class="oraSectionStart oraEmbeddedTable">
        <table cellspacing="2">
          <tr>
            <td>
              <input class="oraButton" oraMdLabel="C1_SAVE_LBL" type="button"
                onClick="oraSubmitMap('OK');"/>
            </td>
            <td>
              <input class="oraButton" oraMdLabel="CANCEL_LBL" type="button"
               onClick="oraSubmitMap('CANCEL',false);"/>
            </td>
          </tr>
        </table>
      </td>
    </tr>
</table>
</body>
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

Basic HTML and Styles

The basic templates introduced the standard HTML and styles used for UI Maps. These standards are described individually in the following sections.

Stylesheets

The styles to apply the standard look to the maps are all contained in stylesheets. These stylesheets should be included in all UI Maps.

```
...
            <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
...
```

Title

Each UI Map should have a <title> tag.

```
...
            <title oraMdLabel="ADDRESS_LBL"></title>
...
```

This will give the UI Map a descriptive title.

- If the UI Map is presented in a "pop-up", the title will be in the window title bar.
- If the UI Map is presented in the page area, the title will be added as a tag to the UI Map and will appear at the top of the UI Map.
- If the UI Map is presented as a zone map, it will be ignored. The <title> tag should still be included in the HTML as standard.

Zone Maps

When the map is presented in a zone as part of a portal, the UI Map should have a border so that the information is "contained" within the zone.

```
...
            <body class="oraZoneMap">
...
```

Page Area Maps vs Pop-Up Maps

The presentation of the UI Maps can vary from design to design. The following standards have been applied to decide when to use a Page Area UI Map and when to use a Pop-Up Map:

- If there are multiple UI Maps in the sequence, always use the Page Area.
- If the UI Map has many input fields, always use a Page Area.
- If the UI Map is a "confirmation" type dialog or only has one or two input fields, use a Pop-Up.

> **Note**
>
> The difference between "just a few input fields" and "many input fields" can be discretionary. The final decision should rest with the dialog designer.

Copyright © 2026, Oracle and/or its affiliates.

Error Messages

Input maps have a ability to present error messages to the User.

```
...
<table width="100%" cellpadding="12">
     <tr class="oraErrorText">
         <td><a href="" onclick="oraShowErrorAlert(); return false;">
        <span class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span></a></td>
     </tr>
</table>
...
```

This HTML structure provides the provides the necessary elements and functions to display errors to the User. It should be directly after the <body> tag. When there is no error, nothing will be visible on the UI Map. It will be made visible if an error occurs and the UI Map is re-presented to the User. Clicking on the link (when visible) will result in a pop-up alert appearing with the long error message text.

Standard Layout and Styles

The information is presented on the UI Map by using a <table> to organize the information in rows and columns.

```
...
<table cellspacing="4" width="100%">
<colgroup>
<col class="oraLabel oraTableLabel" />
             <col class="oraNormal oraTableData" />
             </colgroup>
...
```

The <colgroup> and <col> tags allow for the application of classes to the columns (the label is in the first column and the data is in the second column.). Using these tags mean that the class attribute (to apply styles) does not need to be defined on every <td>.

Grids (Tables of Data)

A UI Map could contain information that is best presented as a grid. These are referred to as "Embedded Tables". The embedded table can be used to display information or input information.

Example Embedded Table HTML

The embedded table will be included within a row (<tr>) of the basic layout:

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
        <table oraList="contactInformation" cellspacing="2">
            <thead >
                <tr>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/type></span>
                    </th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/number"></span>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    </th>
                </tr>
            </thead >
            <tbody>
                <tr>
                    <td class="oraNormalAlt oraDisplayCell">
                        <span oraField="type"></span>
                    </td>
                    <td class="oraNormal oraDisplayCell">
                        <span oraField="number"></span>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
...
<xml>
    <root>
        <address> 123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
```

Embedding the Table

The embedded table is included within the overall table structure. The colspan attribute ensures that the embedded table can span the standard two columns of the overall layout table.

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
    ...
    ...
    ...
    </td>
</tr>
...
```

Embedded Table Structure

Copyright © 2026, Oracle and/or its affiliates.

The embedded table is very similar to the basic layout table.

```
...
<table oraList="contactInformation" cellspacing="2">
<thead>
          ...
          ...
</thead>
<tbody>
          ...
          ...
</tbody>
</table>
...
```

- The <table> tag has a slightly smaller cellspacing and it defines the "list" element contained in the XML that will be used to provide the data.
- The <thead> element is used to give the embedded table headings for the columns.
- The <tbody> element is the element that will be repeated for each referenced "list" element in the XML. In the previous example, there are two "contactInformation" list elements, so the displayed embedded table will have two rows.

Column Headings

Embedded tables should have headings for the displayed columns. The <thead> tag defines these.

```
...
<thead>
    <tr>
        <th class="oraGridColumnHeader" nowrap="nowrap">
            <span oraLabel="contactInformation/type></span>
        </th>
        <th class="oraGridColumnHeader" nowrap="nowrap">
            <span oraLabel="contactInformation/number"></span>
        </th>
    </tr>
</thead>
...
```

The "nowrap" attribute prevent the column heading from taking multiples lines. If multiples lines are required, the "nowrap" may be removed.

Input Fields

Embedded tables may be used for input as well as display only. The framework provides a convenient control to assist in the creation of editable embedded tables.

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
        <table oraList="contactInformation" cellspacing="2">
            <thead >
                <tr>
                    <th class="oraGridColumnHeaderButton"></th>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <th class="oraGridColumnHeaderButton"></th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/type></span>
                    </th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/number"></span>
                    </th>
                </tr>
            </thead >
            <tbody>
                <tr>
                    <td oraType="addGridRow"></td>
                    <td oraType="deleteGridRow"></td>
                    <td>
                        <input type="text" oraField="type"/>
                    </td>
                    <td>
                        <input type="text" oraField="number"/>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
...
```

There are two new columns added to the input embedded table.

- oraType="addGridRow" will add a "+" button to the row. This will allow the User to add an additional row to the existing grid.
- oraType="deleteGridRow" will add a "-" button to the row. This will allow the User to delete an existing row from the grid.

> **Note**
>
> The <thead> tag also requires these two new columns to be added.

These controls are, as standard, placed at the beginning of the row in the order shown. Either of the controls may be omitted if required (if, for example, Users are not permitted to delete information).

The presence of either of these controls will activate the "empty list" process. This means that if the XML has no data for the "list" specified, the input grid will display with an empty row ready for the input of new information.

Action Buttons

Example Action Button HTML

Action buttons are used to perform some specified function from the UI Map. The actions are as varied as the information being displayed/updated. Below are two common examples:

- Save. Normally used on an Input UI Map to allow a User to save any changes they have made.

Copyright © 2026, Oracle and/or its affiliates.

- Cancel. Normally used on an Input UI Map to allow a User to cancel changes in progress.

```
...
<tr>
  <td colspan="2" class=" oraEmbeddedTable">
    <table cellspacing="2">
      <tr>
        <td>
          <input class="oraButton" oraMdLabel="C1_SAVE_LBL" type="button"
              onClick="oraSubmitMap('OK');"/>
        </td>
        <td>
          <input class="oraButton" oraMdLabel="CANCEL_LBL" type="button"
                          onClick="oraSubmitMap('CANCEL',false);"/>
        </td>
      </tr>
    </table>
  </td>
</tr>
...
```

Button Standards

The following points highlight some standards related to buttons.

- Buttons are included as an embedded table.
- Buttons should be grouped together. They should not be placed in different areas of the UI Map.
- The location of the buttons depends mainly on the type of UI Map.
  - Display Only UI Maps should have a Record Actions section in the upper right section of the UI map.
  - Input UI Maps should have the buttons at the foot of the UI Map (after all input fields).

Available Styles

Styles are all contained in the referenced CSS stylesheets. They are applied by the HTML "class" attribute. The actual style settings used are not documented here as they may be adjusted. This section only specifies when a particular style should be used.

> **Note**
>
> The "class" attribute may reference more than one style (class="oraLabel oraSectionEnd")

Copyright © 2026, Oracle and/or its affiliates.

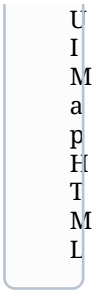
Style	Comments	Example
oraButton	Applied to <input> elements where the type is button.	... <input class="**oraButton**" oraMdLabel="CANCEL_LBL" type="button" onClick="oraSubmitMap('CANCEL',false);"/> ...
oraDisplayCell	Applied to the <td> tag of an embedded table. It defines how the table cell looks (not the data contained inside the cell).	... <td class="**oraDisplayCell**"> </td> ...
oraEmbeddedTable	Applied to the <td> tag that will contain the embedded table.	... <tr> <td colspan="2" class=" oraEmbeddedTable"> <table cellspacing="2"> ... ... </table> </td> </tr> ...

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`oraError`	This style is applied to elements that are identified as "error elements". Refer to Display Errors for more information.	

> **Note**
>
> This style is not normally applied directly in the

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
		
oraErrorText	This style is applied to the elements concerned with error messages.	... <table width="100%" cellpadding="12"> <tr class="oraErrorText"> <td> </td> </tr> </table> ...
oraGridColumnHeader	This style is applied to the <td> tags that define column headers within embedded table.	... <thead > <tr> <th class="oraGridColumnHeaderButton"></th> <th class="oraGridColumnHeaderButton"></th> <th class="**oraGridColumnHeader**" nowrap="nowrap"> <span oraLabel="contactInformation/type> </th> <th class="**oraGridColumnHeader**" nowrap="nowrap"> </th> </tr> </thead > ...

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`oraGridColumnHeaderButton`	This style is applied to the <td> tags that define the column headers for the "+" and "-" buttons used on editable embedded tables.	... <thead > <tr> <th class="**oraGridColumnHeaderButton**"> </th> <th class="**oraGridColumnHeaderButton**"> </th> <th class="oraGridColumnHeader" nowrap="nowrap"> <span oraLabel="contactInformation/type> </th> <th class="oraGridColumnHeader" nowrap="nowrap"> </th> </tr> </thead > ...

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`oraInput`	This style is applied to input fields: • <input type="text"> • <input type="checkbox"> • <select> • <textarea>	``` ... <input type="text" class="oraInput" oraField="address"/> ... ```

> **Note**
>
> This can normally be omitted as input styles are

Style	Comments	Example

```
appliedautomaticallywhenoraSchemaDataTypes="true".
```


Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

`oraInputMoney`	This style is applied to input fields:
	• <input type="text">
	• <select> (rare)
	• <textarea> (not recommended)
		`...`
		`<input type="text" class="oraInputMoney"`
		`oraField="amount"/>`
		`...`

Note

This can normally be omitted as input styles ar

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

```
eappliedautomaticallywhenoraSchemaDataTypes="true".
```


Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`oraInputNumber`	This style is applied to input fields: • <input type="text"> • <select> (rare) • <textarea> (not recommended)	``` ... <input type="text" class="oraInputNumber" oraField="count"/> ... ```

> **Note**
>
> This can normally be omitted as input styles ar

Copyright © 2026, Oracle and/or its affiliates.

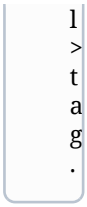
Style	Comments	Example

```
eappliedautomaticallywhenoraSchemaDataTypes="true".
```


Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`oraLabel`	This style is applied to standard label fields that are right aligned.	``` ... <td class="oraLabel" oraLabel="address"></td> ... ```

> **Note**
>
> This can normally be omitted as it is applied by the <co

Style	Comments	Example
		

`oraLabelAlt`	This style is applied to standard label fields only if it is desired to have the label aligned to the left.	`...`
		`<td class="oraLabelAlt"`
		`  oraLabel="address"></td>`
		`...`

`oraLabelCenter`	This style is applied to standard label fields only if it is desired to have the label aligned in the center of the cell.	`...`
		`<td class="oraLabelCenter"`
		`  oraLabel="address"></td>`
		`...`

`oraLink`	This style is applied to foreign key references (links). This is automatically added by the UI Map framework but can also be used manually if desired.	`...`
		`<td class="oraLabel">`
		`  <a href="www.google.com"`
		`  class="oraLink">Google</a></td>`
		`...`

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
oraNormal	This style is applied to standard data fields (display only) that are left aligned.	``` ... <td class="oraNormal" oraField="address"></td> ... ```

> **Note**
>
> This can normally be omitted as it is applied by the <c

Style	Comments	Example
		ol>tag.
oraNormalAlt	This style is applied to standard data fields (display only) only if it is desired to have the data aligned to the right.	... <td class=" **oraNormalAlt**" oraField="address"></td> ...
oraNormalCenter	This style is applied to standard data fields (display only) only if it is desired to have the data aligned in the center of the cell.	... <td class=" **oraNormalCenter**" oraField="address"></td> ...

Copyright © 2026, Oracle and/or its affiliates.

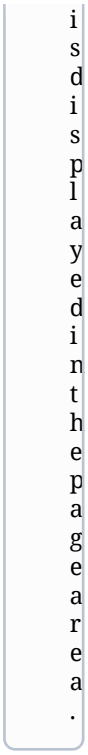
Style	Comments	Example

`oraPageTitle`	This style is applied to the element that contains the page title.	`...`
		`<span class=" oraPageTitle"`
		`  oraMdField="PAGE_TITLE_LBL"></span>`
		`...`

> **Note**
>
> This style is not normally applied directly in the UI M

Style	Comments	Example
		apHTML.TheiscreatedbytheUIMapframeworkwhenthetheUIMap

Copyright © 2026, Oracle and/or its affiliates.

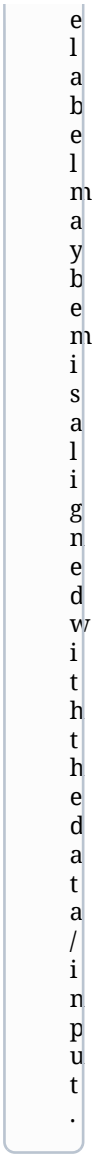
Style	Comments	Example
		

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

`oraSectionEnd`	This style is applied to the <td> tags at the end of a "section" (group of elements). It provides some space to separate the section from the following information.	`...`
		`<tr>`
		`  <td class="oraSectionEnd"`
		`  oraLabel="zip">`
		`  </td>`
		`  <td class="oraSectionEnd"`
		`  oraField="zip">`
		`  </td>`
		`</tr>`
		`...`
		Note
		The style must be applied to both <td> tags or th

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
		

Copyright © 2026, Oracle and/or its affiliates.

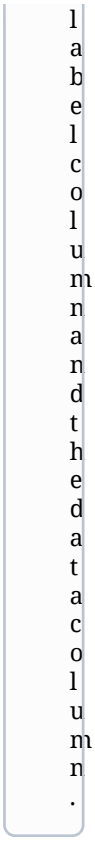
Style	Comments	Example

`oraSectionHeader`	This style is applied to the <td> tag used to give a heading for a section within the information being displayed. It does not provide spacing before or after itself. The oraSectionStart and oraSectionEnd classes are used for this.	```
		...
		<tr>
		  <td class="oraSectionHeader"
		  colspan="2"
		  oraMdField="DATA_SECTION_LBL"></td>
		</tr>
		...
		``

Note

These section headers should span both the

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
		

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

`oraSectionStart`	This style is applied to the <td> tags at the start of a "section" (group of elements). It provides some space to separate the section from the previous information (often a section header).	```
		...
		<tr>
		  <td class="oraSectionStart"
		   oraLabel="zip"></td>
		  <td class="oraSectionStart"
		   oraField="zip"></td>
		</tr>
		...
		``

Note

The style must be applied to both <td> tags or

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
		the label may be mis aligned with the data/input.
oraTableData	This style is applied to the <col> tag for the data column of the main table (second column). It is used to provide a percentage width for the horizontal space to be used for the information.	... <colgroup> <col class="oraLabel oraTableLabel"/> <col class="oraNormal **oraTableData**"/> </colgroup> ...

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
oraTableLabel	This style is applied to the <col> tag for the label column of the main table (first column). It is used to provide a percentage width for the horizontal space to be used for the labels.	... <colgroup> <col class="oraLabel **oraTableLabel**" /> <col class="oraNormal oraTableData"/> </colgroup> ...
oraTinyText	This style is typically applied directly beneath an <input> tag to provide information or a hint to the user concerning information relevant to the input. For example, name or address format.	... <tr> <td oraLabel="address"></td> <td> <input type="text" oraField="address"/> </td> </tr> ...
oraZoneMap	This style is used applied when the UI Map is to be displayed as a zone on a portal.	... <body class="oraZoneMap"> ...
ou-align-items-after-text	Tree nodes may be designed to have their node text directly followed by icons and badges. In some use cases there is a need to align those icons and badges in a column after the text. This is especially true for "narrow" trees where the text tends to wrap. This style aligns tree node icons and badges to the right in a column down the tree.	<ou-tree treename="..." class="ou-align-items-after-text"></ou-tree>

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

`ou-clear-hyperlink-color`	This style may be used to override the standard hyperlink color and use the default text color instead. It applies to all the hyperlinks in the defined container.	`<body class="ou-clear-hyperlink-color"> - Applied to all hyperlinks on the entire page`
		`<div class=ou-clear-hyperlink-color"> - Applied to all hyperlinks in the specified <div>`
		`<a class="ou-clear-hyperlink-color"> - Applied to a single element`

`ou-icon-color-standard`
`ou-icon-color-gray`
`ou-icon-color-red`
`ou-icon-color-green`
`ou-icon-color-paleblue`
`ou-icon-color-yellow`
`ou-icon-color-orange`
`ou-icon-color-white`
`ou-icon-color-reverse-video`
	These styles may be used to change the color of an icon.	`...`
		`<ou-button type="image" imgref="F1NEWWIN" ...`
		`class="ou-icon-color-gray" ...>`
		`</ou-button>`
		`...`

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example
`ou-icon-small` `ou-icon-xsmall`	These styles may be used to change the size of the iconography. u-icon-small makes the icon 16x16 ou-icon-xsmall makes the icon 12x12. This style should only be used for decorator icons. Clickable icons should be standard size for accessibility.	`...` `<ou-decorator imgRef="F1ACCT" class="ou-icon-small"></ou-decorator>`

`ou-show-as-grid`	This style renders single elements as if they were rows in a grid.	`<div class="ou-show-as-grid">`
		`<div>`
		`  <label .../>`
		`  <span .../>`
		`</div>`
		`...`
		`<div>`
		`  <label .../>`
		`  <span .../>`
		`</div>`
		`</div>`

`ou-show-as-label-above`	This style renders an element in redwood view mode which sets the label iabove the field value. The style is ignored in redwood view mode.	`<div class="ou-show-as-label-above">`
		`  <label .../>`
		`  <span .../>`
		`</div>`

`ou-show-as-label-in-front`	This style renders an element in classic view mode which sets the label in front of the field value. The style is ignored in classic view mode.	`<div class="ou-show-as-label-in-front">`
		`  <label .../>`
		`  <span .../>`
		`</div>`

Copyright © 2026, Oracle and/or its affiliates.

Style	Comments	Example

`ou-show-as-standard-lines`	This style may be used to render a simple name/ value two columns table as single elements. It "converts" the table into standard looking line elements.	`<table class="ou-show-as-standard-lines"><tr>`
		`<td>... name ...</td>`
		`<td>... value ...</td>`
		`</tr></table>`

Using OJET

There are some UI maps delivered by the product that use UI widgets provided by Oracle JavaScript Extension Toolkit (OJET). Releases for OJET do not always align with releases of the framework. In addition, there are times when OJET adjusts APIs that the product uses. The framework will attempt to ensure that each release of the product has the latest and greatest version of the OJET libraries. Implementations are discouraged from attempting to use features in OJET that are not used by the product because the product is not necessarily testing those features and is not ensuring that upgrades to the APIs for those features are backward compatible.

Note that the product isolates the references to OJET into a UI map fragment that is included in the maps that use OJET widgets. This is so that changes to future versions of OJET are minimized to a single place. The map is called F1–OJETLIBS. If your implementation wants to use OJET, the recommendation is to use this UI map fragment.

Process Flows

This section describes concepts and provides guidelines related to the configuration of various type of process flows.

- Understanding Process Flows
- Designing Process Flows
- Defining Process Flow Types

Understanding Process Flows

A process flow is a user interface guiding a user through a series of actions in order to accomplish a specific task. The task can be as simple as the collection of information in order to update business data or involve more complex logic such as submitting and tracking batch processes, exchanging messages with an external system, etc.

This section describes topics related to designing and working with process flows.

A Process Flow Is Made Of Panels

Each process flow consist of a number of sequential "steps" needed to accomplish a certain task. Each step is represented by a stop on a progress bar and an associated panel.

Progress is linear in that each step may only lead to a single next step. A process flow always starts at a single initial step but allows for one or more final steps at which the user may choose to complete the process.

Copyright © 2026, Oracle and/or its affiliates.

At any step the user may take the following actions:

- Enter data or take action as prompted by the step's panel.
- Continue to a next step, if any.
- Navigate back to any step they have previously visited.
- Finish the process flow if the current step is a final step.
- Save off their work on the process flow and either continue their work or navigate away. At a later time the user may resume their work on the process flow from where they have left off.
- Cancel the process flow.

A process flow type defines the entire metadata needed to control the behavior of process flows of a given type. This includes the sequence of steps, the panel and rules associated with each step and more.

Panel Presentation

The panel presented on each step of the process flow is rendered based on the panel type selected on the step configuration.

For a **Data Area** type of panel, the panel is rendered using UI Hints built into the schema of a specified data area.

For a **UI Map** type of panel, the panel is rendered using a specified UI map fragment.

For a **Panel Set** type of panel, the panel is rendered as a set of tabs, one for each panel that references the current panel as its parent panel. Clicking on each tab renders the UI based on the selected panel's type. The behavior of a panel set is analogous to or can be thought of as a nested process flow within a single panel. A panel set may not include another panel set.

A Single UI Map

While the user enters data one panel at a time, data is collected and captured at the entire process flow level. All the elements edited and displayed on the various panels of the process flow must be included in the overall process schema and referenced by their appropriate XPath location in that schema. Each type of process flow defines its unique schema on a designated UI map.

The UI map's HTML should simply include the **Process Flow Controller (F1-ProcessController)** UI map fragment and nothing else. The latter is designed to render the user interface for a process flow based on the metadata defined on its process flow type.

Panel Scripts

A single **Pre-Processing** service script may be associated with a panel for the purpose of preparing the data before the panel is presented to the user. The system calls the script each time the user navigates to this panel on a non-finalized process flow.

A single **Post-Processing** service script may be associated with a panel for the purpose of validating the data entered by the user on that panel. The system calls the script each time the user exits the panel on a non-finalized process flow.

Process Flow Scripts

A single **Process Start** service script may be associated with a process flow for the purpose of preparing the data before a newly initiated process flow is presented to the user.

Copyright © 2026, Oracle and/or its affiliates.

A single **Process End** service script is executed when the user clicks on the **Finish** or **Cancel Process** buttons to complete the process flow. The entire set of data collected by the process flow UI map schema is provided to this service script for final processing.

> **Note**
>
> The schema of these scripts should be the same as the **Process Flow UI map** schema. This would ensure that business rules have access to the entire data captured by the user.

Summary Panel

By default, when a user finishes a process flow, as well as when they view an already closed process, the process flow portal displays the last panel the user was working on. For more complex process flows, you may provide a panel that summarizes information from the entire process. When specified on the process flow type, the system displays that summary panel instead a the default view of a closed process. The user may toggle between the summary view and the detailed view at any time.

The summary panel may be implemented as a UI map fragment or a data area like other panels but may not be associated with any panel script.

Launching A New Process Flow

Each process flow must be initiated and executed by a designated BPA script that has the following simple steps:

- Set BPA area height to zero.
- Populate the temporary variable **$processCode** with the code of the specific process flow type.
- Transfer control to the **Process Flow Navigation (F1-PROCEXEC)** common BPA script.

Launching a process flow of a specific type is assumed to be made from a designated menu option or within a specific context applicable to that type of work. As such, this item needs to be configured specifically for each type of process flow. The navigation option associated with that launching option should reference the BPA script associated with the corresponding type of process flow.

Saving Off Work

The user may save off their work on a process flow at any time. If enabled on the process flow type, the system would also automatically save the data when the user navigates to another panel or away from the process flow portal.

On either manual or automatic save operations, the **Process Manager** script defined on the process flow type is called to save off the entire process flow data on a designated record in the database.

The base product provides a generic **Process Flow** maintenance object that is designed to support the storage of any type of process flow as well a corresponding process manager script. Should process flows of a specific type be stored elsewhere, then a dedicated process manager script should be implemented and used instead.

Copyright © 2026, Oracle and/or its affiliates.

Review Process Flow Records

A process flow record is created when a user saved off their work or the process flow type enables automatic saving. When the user finishes a process flow, depending on the process flow type configuration the record may be deleted or retained for audit purposes.

The **Process Flow Query** portal allows a user to resume their work on a in-progress flow and review the information associated with any finalized flows.

Reviewing process flow records of a specific type could also be made from a designated menu option or within a specific context applicable to that type of work. As such this item needs to be configured specifically for each type of process flow. The navigation option associated with that item should reference a query portal designed to query and manage the specific type of process flow records.

Designing Process Flows

A wide range of process flows may be designed to implement online tasks business users may need to perform. Refer to Understanding Process Flows for more information.

Below is a high-level summary of the steps required to design and configure various types of process flows.

Analyze The Task

The following provides a the high-level guidance in analyzing the necessary data and rules needed to support an implementation of a new type of process flow:

- Identify the data and business rules that are needed to complete the entire task.
  - When possible, organize data into logical groups of details defined as data areas.
  - Design logic for the final processing script that accepts the entire data entered along with the user's action to either complete or cancel the task.
  - If data needs to be initially prepared for a newly initiated process flow, design logic for a corresponding initial processing script.
  - Determine whether a summary panel may be useful for the user to review the outcome of a completed process.
- Design the sequence of panels the user should follow to complete the task. Identify the initial and final panels.
- Review each panel and determine the following:
  - Is it a single panel or a nested process flow (i.e. a panel set)?
  - What is the title of the panel and its label on the progress bar?
  - How should the layout be rendered? Use a data area that leverages UI Hints to describe the panel layout when possible. For more complex HTML requirements you may use a UI map fragment.
  - Should details on this panel be defaulted with some initial values when the user enters this panel? If so, design logic for a corresponding pre-processing script.
  - If data entered on this panel should be validated before the user exits the panel, design logic for a corresponding post-processing script.
- Design the summary panel if needed.

Copyright © 2026, Oracle and/or its affiliates.

- Determine whether this type of process flow should be automatically saved by the system or solely rely on the user to control when data is saved.
- Determine whether a record of a completed process flow should be retained for audit purposes or deleted upon completion.
- Determine whether a user may delete in progress records of such process flow.
- Determine whether process flows of this type are saved and stored in a dedicated table in the database or are saved in the generic table provided by the base product. The assumption is that introducing new designated storage maintenance objects is not common. Refer to your edge product documentation for additional maintenance objects that may have been provided to support specific types of process flows shipped with the product.
- Determine the context in which a user can initiate, resume work on, and review process flows of that type.
  - The process flow may be launched from a menu or a favorite script etc.
  - Design the query portal a user would use to find and resume a saved off process flow of this type as well as review completed process flow records. This step is only needed if the base product query portal is not suitable for this type of tasks. Depending on the context of this task, the query may be associated with the same menu line that initiates the process flow.
  - Determine if specific characteristics may need to be stored along with the process flow record for the purpose of identifying in-progress flows by context. Design your process flow service scripts to populate these characteristic values in the control data area available to these scripts.

Using UI Map Panels

Typically a map fragment is rendered as part of a complete UI map but when used as a process flow panel the system renders the panel assuming the fragment contains a complete map.

Therefore, such fragment should includes the following items:

- The standard error message section used for reporting errors.
- This line <xml style="display:none"></xml> at the end of the HTML content for proper binding of data to their HTML elements.

A UI map based panel allows for more control over some of the buttons available to the user. For example, the panel may prevent the user from proceeding to the next panel unless required information is entered. Refer to the explicit APIs listed in the process controller UI Map (F1-ProcessController) for more information.

Configure The Process Flow Type

The following describes the high-level steps needed to configure a new type of process flow:

- Create the UI map for the process flow. Its schema should contain all the details needed to complete the task. Its HTML should only include the process controller UI map fragment.
- Create a start processing service script if needed.
- Create a final processing service script such that it applies all the rules needed to complete or cancel the task.
- For each panel perform the following:

Copyright © 2026, Oracle and/or its affiliates.

  - Create a UI map fragment for it unless it corresponds to a data area already included in the process flow schema. Make sure the XPath of each element references in the data area or UI map is the same XPath of that element in the process flow's UI map schema.
  - Create a pre-processing service script if needed. Not applicable to the summary panel.
  - Create a post-processing service script if needed. Not applicable to the summary panel.
  - Set up a label and title MD fields if existing ones cannot be used.
- Create a BPA script for the process flow.
- Configure the process flow type record.
- Set up a navigation option and reference the BPA script on it. Associate the navigation option with a menu or provide other means for the user to initiate the process flow.
- Create a query portal for the specific type of process flow records. Add the new query portal to a menu or provide other means for the user to resume work on a process flow they have saved off or review completed records.

Control Visibility of Standard Buttons

You may leverage the following javaScript functions to control the visibility of standard process flow buttons:

- Next: `enableNext()`, `disableNext()`, `hideNext()` , `showNext()`.
- Finish: `enableFinish()`, `disableFinish()`, `hideFinish()` , `showFinish()`.
- Save: `enableSave()`, `disableSave()`, `hideSave()` , `showSave()`.

Defining Process Flow Types

This portal is used to maintain process flow types.

Refer to Understanding Process Flows for more information.

You can access the portal from the **Admin Menu**, and then **System**, and then **Process Flow Type.** You are brought to a query portal with options for searching for a specific process flow type. Once a process flow type has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Process Flow Type.** This zone provides information about the selected process flow type.

Action Method

Action method is an object that may abe used to define configuration based on some criteria. For example, when initiating a Start Service scenario, the service request type may differ based on the CIS division and the customer class. When a user initiates a Start Service process, the system can prompt the user for information about the customer that can determine the CIS division and the customer class. The system can then call an algorithm passing this information to retrieve the appropriate service request type to use as defined in the action method. Throughout this section, the term 'criteria' is used to identify the information used to determine the desired configuration and the term 'details' or 'results' is used to identify the configuration information being determined.

If you were to design a use case to use this functionality consider the following:

Copyright © 2026, Oracle and/or its affiliates.

- What information needs to be identified (the 'details'/ 'results')? The example above uses a configuration object as its example (service request type). However, the information you are determining could be anything that can differ by some combination of attributes. You could use this to determine which collection algorithm to execute based on criteria. Or the number of days to wait to continue with some process for some use case.
- When and where do I need to determine this information and what do I do with it once I receive it?
  - Is this needed as part of a user interaction?
  - Or is this needed for internal processing?
- What is the criteria needed to determine the desired information? Do I have this information at the time that I need it? Or do I need to prompt a user for the information?

The framework product supplies the generic objects needed to support this type of functionality. The individual edge products or implementations can build their use cases with the tools provided. The topics in this section further describes the provided features.

Action Method Role

The action method functionality is driven by the Action Method Role, which represents a given use case. Some examples of action method roles that may be relevant in an enterprise product are 'Start Service', 'Stop Service' and 'Transfer Service'. The valid action method roles are defined using an extendable lookup and referenced on the Action Method. One and only one action method may be defined for one action method role.

The action method role record may reference an action method role BPA script, if the use case for this Action Method Role involves user interaction. The responsibility of this script, if needed is to take the action method role value, determine the action method that references the role and identify the more specific Action Processing BPA script (defined on the business object) and transferring control to that script. This reference may be used for audit or information purposes, depending on how you have implemented the user experience. The action method role BPA is ideally configured on an appropriate menu entry or an action button on a specific user interface related to this use case.

If your business use case does not require user interaction, then an action method role BPA script is not necessary.

Action Method Business Object

The action method's business object is used to define the information that needs to be configured for both the criteria and the results. Continuing to use the above example, the business object for a Start Service use case defines a list that captures the CIS division and customer class along with the service request type to use. The information is configured in the XML data area for the action method and the schema design is dictated by the business requirements. For example, it may be that your use case dictates a default value for the results (in our case, the service request type) plus override values based on a division and customer class combination.

Get Action Method Details Plug-in

Besides defining the schema information for capturing the criteria and results, the business object defines the algorithm that should be invoked to retrieve the results based on the criteria.

The plug-in spot is **Get Action Method Details**. Its API is flexible in the information it receives and returns so that each action method use case can design the algorithm based on its business requirements.

Copyright © 2026, Oracle and/or its affiliates.

The algorithm receives a list of 'Criteria Data' using a 'name' and up to 5 values (to handle multi-part primary keys). Typically only 'value 1' is needed for each entry. The algorithm type designed for a given action method business object can determine what information is receives. It could be that it expects the criteria elements defined on the action method. In our example, the algorithm type could expect to receive the CIS division and customer class. Alternatively, the algorithm type could be designed to receive a value that can be used to determine the criteria needed. For example, it could receive the account ID and use that information to determine the CIS division and customer class.

The algorithm returns one or more results as 'Details'. The information returned is based on the use case. In our example, the service request type is returned. Your business requirements may use the action method role to return multiple results, for example a service request type and a process flow type to invoke to capture all the information needed to start service.

The product provides a business service `F1-RetrieveActionMethodDetails` to call the algorithm for a given action method value.

Action Processing BPA

If your business requirement requires user interaction either to determine the criteria to pass to the Get Action Method Details algorithm or to use the resulting information to continue with an online action, an Action Processing BPA is needed. This BPA should be designed and linked to your action method business object as an option using the **Action Processing BPA Script** option type.

The design of this BPA depends on your business requirements. But at a high level, it may need to prompt the user for information needed to determine the criteria associated with the action method details, then call the `F1-RetrieveActionMethodDetails` business service to execute the algorithm for your action method. Once the results are returned, the script would proceed with the next step in the business use case.

> **Note**
>
> If any additional information needs to be gathered by the server before or after the call to the business service that executes the algorithm, then the recommendation is that the BPA should call a service script that does all the server level logic at one time.

Action Method Role vs. Action Method Business Object

The action method role defines the use case. The business object is used to design the schema for the criteria and the results and identifies the action processing BPA and the Get Action Method Details plug-in. It may warrant a unique business object for each action method role. However, there may be cases where more than one action method role and its action method can reuse the same business object. For example, imagine there are different service request types for Start Service, Stop Service and Transfer Service, however in all cases the valid value is driven by the CIS division and customer class. These three separate action method roles and action methods can reuse a common business object defining the criteria of CIS division and customer class determining the appropriate service request type.

- Defining Action Methods

Defining Action Methods

Refer to Action Method for an overview of action method functionality.

Copyright © 2026, Oracle and/or its affiliates.

To maintain the action methods applicable to your implementation, open **Admin**, and then **System**, and then **Action Method.**

This is a standard All-in-One portal and includes the standard List and display zones for an action method.

The information captured on the action method depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

Algorithms

In this section, we describe how to set up the user-defined algorithms that perform many important functions including:

- Validating the format of a phone number entered by a user.
- Validating the format of a latitude/longitude geographic code entered by a user.
- In products that support payment and billing:
  - Calculating late payment charges.
  - Calculating the recommended deposit amount.
  - Constructing your GL account during the interface of financial transactions to your GL
- And many other functions...
- The Big Picture Of Algorithms
- Algorithm Entity
- Defining Algorithm Types
- Defining Algorithms
- Advanced Algorithm Tips

The Big Picture Of Algorithms

Many functions in the system are performed using an algorithm. Algorithms are called at strategic points (often referred to as a "plug-in spot") within system processing to perform specific code. The plug-in spots are preconfigured with the product and vary from object to object. Algorithms are used by the base product to provide out of the box functionality. The algorithm framework provides a means for implementations to insert their own custom code, which can be instead of the base delivered code or in addition to it.

> **Note**
>
> **Algorithm = Plug-in.** We use the terms plug-in and algorithm interchangeably throughout this documentation.

Algorithm Type Versus Algorithm

Often the documentation will just use the term 'algorithm', but in actuality there are two important objects that provide this plug-in functionality.

- An **Algorithm Type** defines program to execute and identifies the plug-in spot. The program may be defined using Java or using a plug-in script. The algorithm type may also define parameters that must be supplied to algorithms of this type.

Copyright © 2026, Oracle and/or its affiliates.

- An **Algorithm** is an instance of an **Algorithm Type**. If the algorithm type does not define any parameters, then there will be only one algorithm for the algorithm type. If the algorithm defines parameters, then there could be several algorithms for a given algorithm type, one for each unique set of parameter values for a given use case. The algorithm is the object that gets referenced on a given control table when configuring your system.

Algorithm Plug-in Spot

The plug-in spot, also called Algorithm Entity and System Event, defines several aspects of the functionality for algorithms that are executed.

- When in system processing are algorithms of this plug-in spot executed? For example, algorithms for the Installation Option - To Do Pre-creation are called when a To Do is getting added, before any insert to the database is performed.
- The API for the plug-in spot. This is often referred to as the "hard parameters" for the plug-in spot. What information is every algorithm program provided when called? Additionally, if the algorithm is expected to return information to the caller, this is also part of the hard parameters. For example, the Installation Option - To Do Pre-creation plug-in spot receives all the information about the To Do that is about to be added. It may add or update values. It may also return an indication to not create the To Do, if desired.

> **Note**
>
> The hard parameters are different from the parameters defined on the algorithm type. Those are referred to as the "soft parameters". Those parameters are used to configure settings that the algorithm type code uses per its specific functionality.

- What is the responsibility of algorithms for the plug-in spot? When defining a plug-in spot, the product defines the responsibility of the code for the algorithm types created for that spot. For example, is the algorithm determining information and returning it? Or is it supposed to perform an update? Algorithms for the Installation Option - To Do Pre-creation plug-in spot should not do any updates. Rather they should adjust the To Do information (if desired) in the hard parameters and return them. The calling logic will use that adjusted information when inserting the To Do record.
- Where do you configure the algorithm? We have been using the Installation Option - To Do Pre-creation plug-in spot as an example. As the description suggests, these algorithms are configured on the Installation Option - Algorithm collection. Each plug-in spot defines where (on which control table) the algorithms are plugged in. The system standard for the description of the Algorithm Entity flag value that defines the available plug-in spots is "Control Table Name - Event". For example the To Do Type - Calculate Priority algorithm plug-in spot means that algorithms are configured on the To Do Type. This information also indicates the granularity of the algorithm. Installation Option algorithms are system wide. But an algorithm plugged in on To Do Type, for example, indicates that only To Do Entries for that To Do Type will have that algorithm executed.
- Are multiple algorithms allowed? There are some plug-in spots where it doesn't make sense to support multiple algorithms. An example of this is the Batch Control - Select Records plug-in. This plug-in is used by the plug-in driven batch process and is responsible for selecting the records to process. Only one algorithm of this type is allowed on a given Batch Control. An example of a plug-in spot that would support multiple algorithms is a validation plug-in spot. Note that when a plug-in spot supports multiple algorithms, there can be yet additional aspects of the plug-in spot design.
  - In some cases, this type of plug-in spot simply executes each algorithm. A validation plug-in spot is an example of this.

Copyright © 2026, Oracle and/or its affiliates.

- In some cases, the algorithm could return an indication to the plug-in spot 'driver' to not execute any subsequent algorithms. Business Object Status - Enter is an example of a plug-in spot that supports this capability.
- In some cases, the plug-in spot may support sharing information from one algorithm to the next. Migration Plan - Pre-Compare is an example of a plug-in spot that supports this capability.

Refer to Algorithm Entity for information about a portal provided to display information about a plug-in spot.

Base Product Algorithms

The base product provides many, many algorithm types and algorithms out of the box. Sometimes algorithms are delivered to provide support for common use cases that we know our clients will need. Many algorithms, however, are provided because we are implementing system functionality using the same infrastructure and capabilities that our implementation use for their own custom code.

When an algorithm type is delivered, the decision as to whether an algorithm is also provided depends on some factors related to whether soft parameters are defined for the algorithm type.

- If no parameters are defined for the algorithm type, a corresponding algorithm is typically also provided.
- If parameters are defined and the values are on configuration data that is normally owned by the implementation, a base algorithm will probably not be defined.
- If parameters are defined and the values are not dependent on configuration data, it's possible that one or more algorithms are provided.

To review the algorithm types that are delivered with the product, go to the Algorithm Type query. The query provides several search options, including searching by the algorithm entity lookup value.

- How To Add A New Algorithm
- Minimizing The Impact Of Future Upgrades

How To Add A New Algorithm

Before you can add a new algorithm, you must determine if you can use one of the algorithm types supplied with the system by using the algorithm type query.

If you can use one of the algorithm types, simply add the algorithm and then reference it on the respective control table. Refer to Defining Algorithms for how to do this.

If you have to add a new algorithm type, you may have to involve a programmer. Let's use an example to help clarify what you can do versus what a programmer must do. Assume that you require an additional geographic type validation algorithm. To create your own algorithm type you must:

- Write a new program to validate geographic type in the appropriate manner. Alternatively, you may configure a plug-in script to implement the validation rules. The advantage of the latter is that it does not require programming. Refer to plug-in script for more information.
- Create an Algorithm Type called **Our Geographic Format** (or something appropriate). On this algorithm type, you'd define the name of the program (or the plug-in script) that performs the function. You'd also define the various parameters required by this type of algorithm.
- After creating the new Algorithm Type, you can reference it on an Algorithm.

Copyright © 2026, Oracle and/or its affiliates.

- And finally, you'd reference the new Algorithm on the Geographic Type that requires this validation.

Minimizing The Impact Of Future Upgrades

The system has been designed to use algorithms so an implementation can introduce their own logic in a way that's 100% upgradeable (without the need to retrofit logic). The following points describe strong recommendations about how to construct new algorithm type programs so that you won't have to make program changes during future upgrades:

- Do not alter an algorithm type's hard parameters. For example, you might be tempted to redefine or initialize parameters defined in an algorithm type's linkage section. Do not do this.
- Follow the naming conventions for the new algorithm type code and your source code, i.e., both the source code and the algorithm type should be prefixed with "CM". The reason for this naming convention is to make it impossible for a new, base-package algorithm type from overwriting your source code or algorithm type meta-data (we will never develop a program or introduce meta-data beginning with CM).
- Avoid using embedded SQL to perform insert/update/delete. Rather, invoke the base-package's object routines or common routines.
- Avoid using base messages (outside of common messages, i.e., those with a message number < 1000) as we may deprecate or change these messages in future releases. The most common problem is caused when an implementation clones a base package algorithm type program because they need to change a few lines of logic. Technically, to be 100% upgradeable, you should add new messages in the "90000" or greater category (i.e., the category reserved for implementation-specific messages) for every message in your new program even though these messages may be duplicates of those in the base package.

Algorithm Entity

This portal is used to view information about a plug-in spot (algorithm entity). Refer to Algorithm Plug-in Spot for more information.

Navigate using **Admin**, and then **System**, and then **Algorithm Entity**.

Use the **Algorithm Entity Query** portal to search for an existing plug-in spot. Each plug-in spot is defined in the Algorithm Entity lookup (`ALG_ENTITY_FLG`). Once an algorithm entity is selected, you are brought to the viewing portal to view details about the algorithm entity.

The following zones appear on the Main page.

- **Algorithm Entity**. This zone provides information about the plug-in spot, including a link to view the Plug-in Spot API, also called the hard parameters for the plug-in spot. In addition it displays the detailed description, which should provide information about the algorithm's responsibility and when it is called.

  > **Note**
  >
  > This information is also available via a help icon on the algorithm type page and when defining a plug-in script, once you select the algorithm entity value.

- **Algorithm Types**. All existing algorithm types for the plug-in spot are listed. If there are algorithms, the count of algorithms are displayed. Clicking the count opens another zone.

Copyright © 2026, Oracle and/or its affiliates.

- **Algorithms**. If you clicked the algorithm count for an algorithm type, another zone appears with information about the algorithms for that type.

Refer to the embedded help in the portal for more information about each zone.

Defining Algorithm Types

The system provides many algorithm types to support base product functionality.

You can access the portal from the **Admin**, and then **System**, and then **Algorithm Type**. You are brought to a query portal with options for searching for a specific algorithm type. Once an algorithm type has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to The Big Picture Of Algorithms for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new algorithm type, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Algorithm Type.** This zone provides information about the selected algorithm type.
  - Algorithm Entity defines where algorithms of this type can be "plugged in". The values for this field are customizable using the lookup table. This field name is `ALG_ENTITY_FLG`.
  - > ⓘ **Note**
    >
    > When adding a new algorithm type that is for a Java program, the parameters are automatically generated based on the Java code. Once an algorithm type exists, any additional parameters defined in the Java code should be manually added to the algorithm type. For other program types, algorithm type parameters must be manually defined.
  - > ⓘ **Note**
    >
    > When a new algorithm type parameter is added for any program type, existing algorithms for the algorithm type do not automatically get updated with the new parameter. The algorithms must be manually updated.
- **Algorithms.** This zone lists available algorithms for the selected algorithm type .

Actions

The page supports the following special actions:

- **View the source.** If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the Javadocs viewer. For plug-in scripts, drill into the plug-in script to view the details.

Copyright © 2026, Oracle and/or its affiliates.

Defining Algorithms

This portal is used to view and maintain information about an algorithm.

You can access the portal from the **Admin Menu**, and then **System**, and then **Algorithm**. You are brought to a query portal with options for searching for a specific algorithm. Once an algorithm has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to The Big Picture Of Algorithms for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new algorithm, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Algorithm.** This zone provides information about the selected algorithm.
- **Other Algorithms.** This zone lists other algorithms of the same algorithm type. The zone is only visible when such algorithms exist.

The following zones may appear as part of the portal's **References** tab page

- **Algorithm References.** This zone list various configuration entities referencing the current algorithm.

> ⓘ **Note**
>
> If the product delivers an algorithm with parameter values defined, an implementation may override the base provided parameter values by adding an additional effective dated collection of parameters.

> ⓘ **Note**
>
> If an algorithm is defined and subsequently a new parameter is added to the algorithm type, existing algorithms for the algorithm type should be updated to add a new effective dated entry to the parameter collection. At this point the latest list of parameters for the algorithm type are visible. Configure the parameters accordingly.

Advanced Algorithm Tips

Initiating Actions that Cannot be Rolled Back

If you have an algorithm that creates a sync request or performs some other logic that could cause a message to be sent and cannot be rolled back, you should be sure to cater for warnings in the design.

- The algorithm should be performed after all validations that could accumulate warnings.

Copyright © 2026, Oracle and/or its affiliates.

- The algorithm should include a step to call the Business Service `F1-CheckWarning` and see if any warnings have been accumulated. If so, don't perform the action. It means that the user will get a warning and if they click "OK", all the processing will happen again, but this time without accumulating any warnings. Refer to the plug-in script `F1-GnChgDCap` for an example.

Script Options

We use the term "script" to define processing rules that your implementation sets up to control both front-end and back-end processing:

- Rules that control front-end processing are called Business Process Assistant (BPA) scripts. For example, your implementation could set up a BPA script to assist a user through your organization's payment cancellation process.
- Rules that control back-end processing are called Server-based scripts. For example, your implementation could set up a server-based script to control the processing that executes whenever a given type of adjustment is canceled.

The topics in this section describe how to configure your scripts.

- The Big Picture Of Scripts
- The Big Picture Of BPA Scripts
- The Big Picture Of Server-Based Scripts
- Defining Scripts
- Merging Scripts
- Maintaining Functions

The Big Picture Of Scripts

This section describes features and functions that are shared by both BPA scripts and server-based scripts.

- Scripts Are Business Process-Oriented
- A Script Is Composed Of Steps
- A Script May Declare Data Areas
- Securing Script Execution
- Script Engine Version

Scripts Are Business Process-Oriented

To create a script, you must analyze the steps used to implement a given business process. For example, you could create a "stop auto pay" BPA script that:

- Asks the user to search and select the customer / taxpayer
- Asks the user to define the date on which the person would like to stop making automatic payments
- Invokes a server-based script that populates the end-date on the account's latest automatic payment instructions.

After you understand the business process, you can set up a script to mimic these steps. If the business process is straightforward (e.g., users always perform the same steps), the script

Copyright © 2026, Oracle and/or its affiliates.

configuration will be straightforward. If the business process has several logic branches, the composition of the script may be more challenging.

A Script Is Composed Of Steps

A script contains one or more steps. For example, a "stop auto pay" BPA script might have three steps:

- Ask the user to search and select the customer / taxpayer
- Ask the customer the date on which they'd like to stop making automatic payments (and default the current date)
- Invoke a server-based script that, in turn, updates the account's auto pay options.

Each step references a step type. The step type controls what happens when a step executes. It might be helpful to think of a script as a macro and each step as a "line of code" in the macro. Each step's step type controls the function that is executed when the step is performed.

> **NOT_SUPPORTED**
>
> Refer to How To Set Up Each Step Type for a detailed description of all available step types and how to set them up.

A Script May Declare Data Areas

Both BPA and server-based scripts may have one or more data areas:

- If the script contains steps that exchange XML documents, you must declare a data area for each type of XML document. For example, if a BPA script has a step that invokes a service script, the BPA script must declare a data area that holds the XML document that is used to pass information to and from the service script.
- You can use a data area as a more definitive way to declare your temporary storage. For example, you can describe your script's temporary storage variables using a stand-alone data area schema and associate it with your script.

Various step types involve referencing the script's data areas as well as support the ability to compare and move data to and from field elements residing in the data areas.

An Edit Data step supports the syntax to dynamically declare data areas as part of the step itself. This technique eliminates the need to statically declare a data area. Refer to Edit Data Syntax for more information on edit data commands and examples of the use of dynamic data areas.

> **Note**
>
> Some server based scripts may not use data areas as means of defining or exchanging data, depending on script type and the chosen scripting technique. Refer to The Big Picture Of Server Based Scripts for an overview of server scripts and their applicable scripting options.

Copyright © 2026, Oracle and/or its affiliates.

Securing Script Execution

The system supports the ability to secure the execution of scripts by associating the script with an Application Service. Refer to The Big Picture of Application Security for more information. Application security is optional for user-invocable BPA scripts. If a script is not associated with an application service, all users may execute the script. Otherwise, only users that have **Execute** access to the application service may execute the script. For service scripts, the application service is required.

Script Engine Version

Script engine version defines key information affecting the context and execution of the script.

- Script engine version values of 1, 2 and 3 define the version of the XML Path Language (XPath) to be used for the script. Versions 2 and 3 use the XPath 2 engine supplied by the XQuery team. This is the same engine used inside the Oracle database. The current script engine version 3 is a modified version that offers performance improvements without impacting existing version 2 scripts.

  The default script engine version is 3.0 for plug-in and service scripts. The default version for BPA scripts is 1.0 as higher level versions are not applicable.

  There are some additional details to note about script engine version 1.0:

  – The XPath library used is Jaxen

  – For BPA scripts, it uses the browser's xpath and XML support except for Internet Explorer where the XSXML parser is used.

  – Xpath 1 (and even JavaScript) uses floating point arithmetic, which means that adding a collection of numbers with two decimal places might end up with a value of 10779.079999999998 instead of 10779.08

- A Script Engine Version value of **Groovy** is only applicable to plug-in scripts and Groovy Library scripts. It is used to indicate that only Groovy related step types are used in the script and signals to the system that there is no need to convert the data to and from an XML interface. This allows for greater efficiency in script execution. This is the default value for Groovy Library scripts and cannot be changed.
- The value **Framework Version 2.1 Compatibility Mode** remains for upgrade purposes. This value should only be applicable to early versions of BPA scripts using syntax that is incompatible with XPath syntax.

The Big Picture Of BPA Scripts

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Scripts to better understand the basic concept of scripts.

Users may require instructions in order to perform certain tasks. You may set up scripts that step a user through your business processes. For example, you might want to create scripts to help users do the following:

- Add a new person to an existing account
- Set up a customer to pay automatically

Copyright © 2026, Oracle and/or its affiliates.

- Modify a customer who no longer wants to receive marketing information
- Modify a customer's web password
- Record a trouble order
- Merge two accounts into one account
- Fix a bill with an invalid rate

Users execute these business processes by initiating BPA scripts (BPA). Users can also define their favorite BPA scripts in their user preferences.

BPA scripts may be designed by either of the following approaches:

- **Self-contained** and standalone flows. With this approach, the script is executed as an independent flow that cannot interact with the current application pages. Every step in these flows is contained within the script itself, including prompting for additional information from the user and performing server functions etc.
- **Guiding** flows that surf on top of one or more application pages and interact with these pages in parallel to script execution.

> ⚠ **Caution**
>
> This approach is considered legacy functionality and as such limited to fixed pages. It does not work with portal pages. With portals being the product's modern user interface technology, the overall support for this type of BPA scripts is planned to be deprecated in a future release.

> ⚠ **Caution**
>
> Future upgrade issues. There may be times when fixed pages are changed or converted to portals to leverage modern user experience capabilities. Such changes made by the base-package will necessitate a redesign of these guided flows. Please refer to the release notes for such changes.

The topics in this section describe background topics relevant to BPA scripts.

- How To Invoke Scripts
- Developing and Debugging Your BPA Scripts
- Launching A Script From A Menu
- Launching A Script When Starting The System
- Executing A Script When A To Do Entry Is Selected
- The Big Picture Of Script Eligibility Rules

How To Invoke Scripts

Refer to Initiating Scripts for a description of how end-users initiate scripts.

The **Accessibility Option** setting for a BPA script may be used to control how it can be initiated. Set this value to **Accessible from Script Menu** for any script that may be launched as a stand-alone script. Scripts with this configuration may be linked to a navigation option so that they may be invoked from a menu and may be configured by a user as a favorite script. Set this value to **Not Accessible from Script Menu** for any script that cannot be launched on

Copyright © 2026, Oracle and/or its affiliates.

its own. For example, any script that is a invoked as a sub-script from another script should have this setting. In addition, any script that is designed to be launched from within a specific portal where certain data is provided to the script should include this setting.

Developing and Debugging Your BPA Scripts

We recommend considering the approaches outlined below when you construct scripts.

While designing your scripts, determine the most maintainable way to set them up. Rather than creating complex, monolithic scripts, we recommend dividing scripts into smaller sections. For example

- Determine if several scripts have similar steps. If so, set up a script that contains these common steps and invoke it from the main scripts using a **Perform script** step.
- Determine if a large script can be divided into logical sections. If so, set up a small script for each section and create a "master script" to invoke each sub script via a **Transfer control** step.

For debugging purposes, you may find it helpful to categorize the step types into two groups: those that involve some type of activity in the script area, and those that don't. The following step types cause activity in the script area: **Height**, **Display text**, **Prompt user**, **Input data**, **Input Map**, **Set focus to a field**.

> ⚠ **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

The rest of the step types are procedural and involve no user interaction. There are two techniques you can use to assist in debugging these step types.

- You can instruct the system to display text in the script area.
- You can display an entire data area (or a portion thereof) in the script area by entering **%+...+%** where **...** is the name of the node whose element(s) should be displayed.

> ⓘ **Note**
>
> **Time saver.** When you develop a new BPA script, change your user preferences to include the script as your first "favorite". This way, you can press `Ctrl+Shift+1` to invoke the script (eliminating the need to select it from the script menu).

Launching A Script From A Menu

You can create menu items that launch BPA scripts rather than open a page. To do this, create a navigation option that references your script and then add a menu item that references the navigation option.

If the navigation option is referenced on a context menu and the navigation option has a "context field", a temporary storage variable will be created and populated with the unique identifier of the object in context. For example, if you add a "script" navigation option to the bill context menu and this navigation option has a context field of BILL_ID, the system will create a

Copyright © 2026, Oracle and/or its affiliates.

temporary storage variable called BILL_ID and populate it with the respective bill id when the menu item is selected.

Launching A Script When Starting The System

You can set the system to launch a script upon startup. This may be used for integrations from an external system or for use cases within the application where a link has been configured to launch the application in a new browser tab, allowing the original page to remain open. Refer to Creating Application Links in External Applications for details of what is supported.

Executing A Script When A To Do Entry Is Selected

The system creates To Do entries to highlight tasks that require attention (e.g., records in error). Users can complete many of these tasks without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected by a user. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill.

The following points provide background information to help you understand how to implement this functionality:

- Every To Do entry references a To Do type and a message category and number. The To Do type defines the category of the task (e.g., bill errors). The message number defines the specific issue (e.g., a valid address can't be found.). Refer to The Big Picture of System Messages for more information about message categories and numbers.
- When a user drills down on a To Do entry, either a script launches OR the user is transferred to the transaction associated with the entry's To Do type. You control what happens by configuring the To Do type accordingly:
  - If you want to launch a script when a user drills down on an entry, you link the script to the To Do type and message number. Keep in mind that you can define a different script for every message (and some To Do types have many different messages).
  - If the system doesn't find a script for an entry's To Do type and message number, it transfers the user to the To Do type's default transaction.

> **Note**
>
> **How do you find the message numbers?** We do not supply documentation of every To Do type's message numbers (this list would be impossible to maintain and therefore untrustworthy). The best way to determine which message numbers warrant a script is during pre-production when you're testing the system. During this period, To Do entries will be generated. For those entries that warrant a script, simply display the entry on To Do maintenance. On this page, you will find the entry's message number adjacent to the description.

- These types of scripts invariably need to access data that resides on the selected To Do entry. Refer to How To Use To Do Fields for the details.

The Big Picture Of Script Eligibility Rules

You can configure eligibility criteria on the scripts to limit the scripts that a user sees in the script search. For example, you could indicate a script should only appear on the script menu if

Copyright © 2026, Oracle and/or its affiliates.

the user belongs to the level 1 customer service representative group. You may also indicate that a script should only appear if the data a user is viewing has certain criteria. For example, if you are using Oracle Utilities Customer Care and Billing, you can indicate that a script should only appear if the current account's customer class is residential. By doing this, you avoid presenting the user with scripts that aren't applicable to the current data in context or the user's role.

The topics in this section describe eligibility rules.

- Script Eligibility Rules Are Not Strictly Enforced
- You Can Mark A Script As Always Eligible
- You Can Mark A Script As Never Eligible
- Criteria Groups versus Eligibility Criteria
- Defining Logical Criteria
- Examples Of Script Eligibility Rules

Script Eligibility Rules Are Not Strictly Enforced

The script search gives a user a choice of seeing all scripts or only scripts that are eligible (given the current data in context and their user profile). This means that it's possible for a script that isn't eligible for the given context data / user to be executed via this search. In other words, the system does not strictly enforce a script's eligibility rules.

It might be more helpful to think of eligibility rules as "highlight conditions". These "highlight conditions" simply control whether the script appears in the script search when a user indicates they only want to see eligible scripts.

You Can Mark A Script As Always Eligible

If you don't want to configure eligibility rules, you don't have to. Simply indicate that the script is always eligible.

You Can Mark A Script As Never Eligible

If you have scripts that you do not want a user to select from the script menu, indicate that it is never eligible. An example of a script that you wouldn't want a user to select from the menu is one that is launched when a To Do entry is selected. These types of scripts most likely rely on data linked to the selected To Do entry. As a result, a user should only launch scripts of this type from the To Do entry and not from the script menu.

Criteria Groups versus Eligibility Criteria

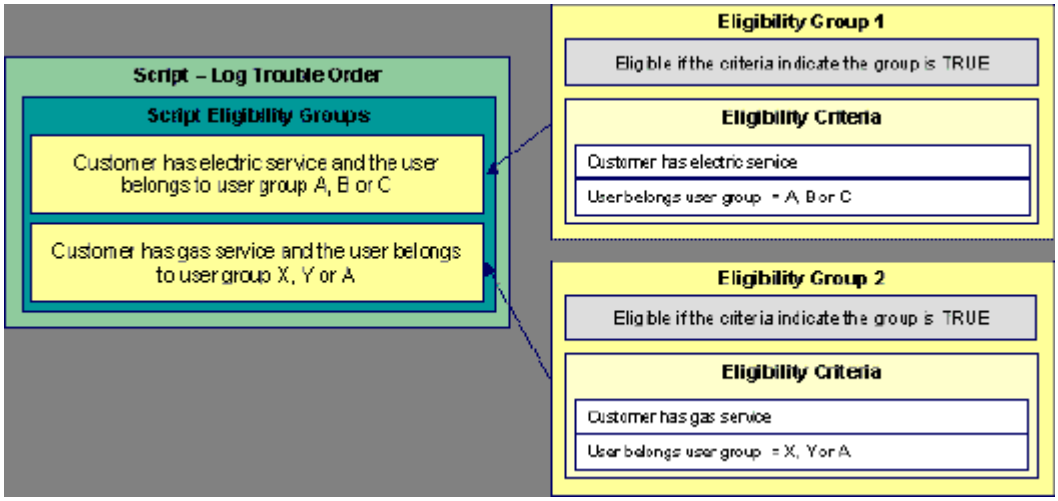
Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A script's criteria groups control whether a user is eligible to choose a script. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered true or false.
- When you create a group, you define what should happen if the group is true or false. You have the following choices:
  - The user is eligible to choose the script.
  - The user is not eligible to choose the script.
  - The next group should be checked.

Copyright © 2026, Oracle and/or its affiliates.

We will use the following example from Oracle Utilities Customer Care and Billing to help illustrate these points. Assume a script is only eligible if:

- The customer has electric service and the user belongs to user group A, B or C.
- OR, the customer has gas service and the user belongs to user group X, Y or A.



This script requires two eligibility groups because it has two distinct conditions:

- IF (Customer has electric service AND (User belongs to user group A, B or C))
- IF (Customer has gas service AND (User belongs to user group X, Y or A))

If either condition is true, the script is eligible.

You would need to set the following criteria groups in order to support this requirement:

Group No.	Group Description	If Group is True	If Group is False
1	Customer has electric service and the user belongs to user group A, B or C	**Eligible**	**Check next group**
2	Customer has gas service and the user belongs to user group X, Y or A	**Eligible**	**Ineligible**

The following criteria are required for each of the above groups:

Group 1: Customer has electric service and the user belongs to user group A, B or C				
Seq	Logical Criteria	If Eligibility Criteria is True	If Eligibility Criteria is False	If Insufficient Data
10	Customer has electric service	**Check next condition**	**Group is false**	**Group is false**
20	User belongs to user group A, B or C	**Group is true**	**Group is false**	**Group is false**

Group 2: Customer has gas service and the user belongs to user group X, Y or A				
Seq	Logical Criteria	If Eligibility Criteria is True	If Eligibility Criteria is False	If Insufficient Data

Copyright © 2026, Oracle and/or its affiliates.

10	Customer has gas service	**Check next condition**	**Group is false**	**Group is false**
20	User belongs to user group X, Y or A	**Group is true**	**Group is false**	**Group is false**

The next section describes how you might configure the specific logical criteria in each of the groups.

Defining Logical Criteria

When you set up an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the *field to be compared* :

- You can execute an algorithm to retrieve a field value from somewhere else in the system.
- Some products may support choosing a characteristic linked to an appropriate object in the system (such as an account or person).

You have the following choices in respect of identifying the *comparison method*:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm that performs the comparison (and returns True, False or Insufficient Data). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The Examples Of Script Eligibility Rules provide examples to help you understand this design.

Examples Of Script Eligibility Rules

The topics in this section provide examples about how to set up script eligibility rules.

- A Script With A Time Span Comparison
- A Script With Service Type Comparison

A Script With A Time Span Comparison

A script that is only eligible for senior citizens has the following eligibility rules:

- Customer class = Residential
- Birth date equates to that of a senior citizen

These rules require only one eligibility group on the script. It would look as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Residential and Senior Citizen	**Eligible**	**Ineligible**

The following criteria will be required for this group:

Group 1: Residential, Calif, Senior

Copyright © 2026, Oracle and/or its affiliates.

Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data
10	Algorithm: retrieve account's customer class	= R	**Check next condition**	**Group is false**	**Group is false**
30	Person characteristic: Date of Birth	Algorithm: True if senior	**Group is true**	**Group is false**	**Group is false**

The first criterion is easy; it calls an algorithm that retrieves a field on the current account. This value, in turn, is compared to a given value. If the comparison results in a True value, the next condition is checked. If the comparison doesn't result in a True value, the **Group is false** (and, the group indicates that if the group is false, the script isn't eligible).

The last criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base script as described below.

- Field to Compare. The person characteristic in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of **True** if the related field value (the customer's date of birth) is greater than 65 years (refer to `SECC-TIMESPN` for an example of this type of algorithm). Note that this time span algorithm can be used to compare days, weeks and months as well.

You'll notice that if a value of **True** is returned by the **True if senior** algorithm, the group is true (and we've set up the group to indicate a true group means the script is eligible).

A Script With Service Type Comparison

Imagine a script that is only eligible if the current customer has gas service and the user belongs to user groups A, B or C. This script would need the following eligibility rules:

- Customer has gas service
- User belongs to user group A, B, or C

These rules require only one eligibility group on the script. It would looks as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Has gas service and user is part of user group A, B or C	**Eligible**	**Ineligible**

The following criteria are required for this group:

Group 1: Has gas service and user is part of user group A, B, or C

Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data
10	Algorithm: check if customer has gas service	= True	**Check next condition**	**Group is false**	**Group is false**

Copyright © 2026, Oracle and/or its affiliates.

20	Algorithm: check if user belongs to user group A, B or C	= True	**Group is true**	**Group is false**	**Group is false**

Both criteria are similar - they call an algorithm that performs a logical comparison. These algorithms are a bit counter intuitive (but understanding them provides you with another way to implement complex eligibility criteria):

The first criterion works as follows:

- Field to Compare. We chose a "field to compare" algorithm that checks if the current account has service agreements that belong to a given set of service types. It returns a value of **True** if the customer has an active service agreement that matches one of the service types in the algorithm. In our example, the "check if customer has gas service" algorithm returns a value of **True** if the customer has at least one active service agreement whose SA type references the gas service type. The "check if customer has electric service" algorithm is almost identical, only the service type differs.
- Comparison Method. We simply compare the value returned by the algorithm to True and indicate the appropriate response.

The second criterion works similarly:

- Field to Compare. We chose a "field to compare" algorithm that checks if the user belongs to any user group in a set of user groups. It returns a value of **True** if the user belongs to at least one user group defined in parameters of the algorithm. Refer to algorithm type `SECF-USRNGRP` for an example of this type of logic.
- Comparison Method. We simply compare the value returned by the algorithm to True and indicate the appropriate response.

> **Note**
>
> **Bottom line.** The "field to compare" algorithm isn't actually returning a specific field's value. Rather, it's returning a value of **True** or **False**. This value is in turn, compared by the "comparison method" and the group is set to true, false or check next accordingly.

The Big Picture Of Server-Based Scripts

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Scripts to better understand the basic concept of scripts.

Server-based scripts allow an implementation to configure backend business processes. The system supports three types of server-based scripts, `Plug-In` scripts, **Service** scripts and **Groovy Library** scripts.

- Plug-in scripts allow an implementation to develop routines that are executed from the system's various plug-in spots. For example, an implementation could configure a plug-in script that is executed every time an adjustment of a given type is frozen.

Copyright © 2026, Oracle and/or its affiliates.

- Service scripts allow an implementation to develop common routines that are invoked from both front-end and back-end services. For example, an implementation could create a service script that terminates an account's automatic payment preferences. This service script could be invoked from a BPA script initiated by an end-user when a customer asks to stop paying automatically, and it could also be executed from a plug-in script if a customer fails to pay their debt on time. Service scripts are typically written using xpath scripting.
- Groovy Library scripts allow an implementation to develop groups of common routines written in the Groovy language which may be invoked from **Groovy Member** step types.

The topics in this section describe background topics relevant to server-based scripts.

- Additional Coding Options For Server Scripts
- Plug-In Scripts
- Service Scripts
- Groovy Library Scripts
- Debugging Server-Based Scripts

Additional Coding Options For Server Scripts

Server based scripts often perform complex functions best supported by coding in languages with more comprehensive command sets than the base script steps. The system supports two common third-party languages for this purpose.

XML Path Language (XPath) is a language for querying and evaluating elements or nodes in an XML document. XPath commands and expressions can be used directly within edit data step types. The script engine version is used to define the applicable XPath version.

Groovy is an object-oriented, dynamic language for the Java platform. The framework supports the use of Groovy within server-based scripts to provide restricted and controlled access to Java-like code, particularly for cloud based implementations. The following topic provides more information on how to incorporate Groovy code into scripts.

- Using Groovy Within Scripts

Using Groovy Within Scripts

Groovy code can be incorporated in scripts using the step type **Groovy Members**. For each script with Groovy code, there will be a single Groovy class created by concatenating all **Groovy Members** steps.

For security, the product and third party Java classes available for scripting in Groovy are restricted. The allowable base classes may be viewed in the Groovy Javadocs viewer via the 'Java Docs' action in the Script portal. The list of allowable third party classes can be viewed via the 'Third Party Groovy Allowlist' action also in the Script portal.

> **Note**
>
> This system supports the use of Groovy for back end processing purposes. It is not intended for user interfaces. Groovy is also not applicable to BPA scripts.

Groovy Script Engine

For scripts invoked from within Java and entirely written in Groovy should use a script engine version of **Groovy**. The system processes this type of script in a similar way to code written in

Copyright © 2026, Oracle and/or its affiliates.

Java. This avoids the need to convert the script to and from XML format and allows the use of code that acts directly on the system objects with consequent performance benefits.

The following script types support the Groovy engine version:

Plug In Scripts

Plug-in Scripts can be configured to use the Groovy engine if they contain only Groovy oriented step types. The system provides an automatically generated superclass that defines the plug-in spot API. Internally, the Groovy code must conform to the system conventions for Java based algorithm types, including the inclusion of an 'invoke' method that is the plug-in entry point, and the definition of 'soft' parameters using annotations.

Groovy Library Scripts

Groovy Library Scripts provide the ability to create groups of common routines written in Groovy that can be called from within other scripts. Scripts of this type must include a single step type of **Groovy Library Interface** in which the publicly available methods in the library are listed. The supporting code for those methods is defined in one or more **Groovy Members** step types within the library script. The methods defined in the library can accept arguments and return values of any type. Scripts of this type use the Groovy engine by default and cannot include scripting step types.

Scripts that need to invoke methods from a Groovy library can use the createLibraryScript method provided by the system to instantiate the library interface. Ensure that the class being instantiated includes the .class filename extension.

Invoking Groovy within the Scripting Engine

If a plug-in script or a service script is configured to use a scripting engine version, it can include a mixture of regular and **Groovy Members** step types. The script step types will define the process to be executed. The **Groovy Members** steps contain code that can be called from **Edit Data** step types within the script using the **invokeGroovy** command. Only Groovy methods that receive no arguments and return void are supported using this command. Refer to the section on edit data steps for more details.

For scripts using this option, the framework provides a superclass containing methods that support common scripting actions such as move commands, string evaluation, and methods to invoke business objects, business services and service scripts. Refer to the Groovy specific Javadocs for details of the supported methods.

Importing Groovy Classes

All the Groovy classes used by your script should be explicitly declared in a **Groovy Imports** step type. When saving a script, if the system finds classes referenced in the script that are not explicitly declared, the system issues a warning and automatically adds import statements for the classes. Script writers should review the imports to confirm that they are the correct classes (in case there is a class with the same name in two different packages).

Backward Compatibility

In previous releases, the system did not require explicit imports and had logic to automatically import all callable classes available to Groovy at run-time. For any existing script that does not explicitly declare its imports, the system continues to fall back on importing all callable classes at runtime. However, the recommendation is to identify and update any such script as this is not the recommended practice.

The system provides a batch program `F1-CAGVY` that reports all scripts that required the automatic import step to successfully compile. In addition, this batch program is useful when the version of Groovy is upgrade to proactively detect possible issues in compilation of existing

Copyright © 2026, Oracle and/or its affiliates.

scripts that may occur due to changes in Groovy. Implementations are encouraged to run this batch program in the short term to identify and fix any scripts missing explicit imports and also for any Groovy version upgrade to review any implementation owned scripts that may be reported.

Complex SQL Statements in Groovy

> **Important**
>
> Whenever you are designing an SQL statement, it is strongly recommended to work with your DBA to ensure the query meets performance standards.

There is a string size limitation in Groovy of 65,535 when using createPreparedStatement. This section highlights valid syntax to use to get around this limitation, without the risk of SQL injection.

Technique 1 - Simple String Concatenation with '+'

The createPreparedStatement method supports concatenating multiple long strings using the + (plus) operator as a connector.

```
def query = createPreparedStatement("SELECT * FROM TABLE_NAME"
+ "WHERE FIELD_A = 'VALUE1'"
+ "AND FIELD_B = 'VALUE2'"), 'My Query');
```

Technique 2 - Special Method in Prepared Statement

The system provides a method concatenateGroovyStrings which accepts varargs of strings:

```
concatenateGroovyStrings(String... args)
```

This method may be used directly in createPreparedStatement as an argument:

```
def query = createPreparedStatement(concatenateGroovyStrings("SELECT* ",
"FROM TABLE_NAME", "My query");
```

Technique 3 - Reference a Variable in Prepared Statement

The system supports defining a variable as the first argument in createPreparedStatement as long as the variable was built using the concatenateGroovyStrings method.

```
def concatStr = concatenateGroovyStrings("SELECT * ", "FROM ", "TABLE_NAME");
def query = createPreparedStatement(concatStr, "My Query");
```

The system validates that the variable used as an argument was created using the concatenateGroovyStrings method. The system also supports a straight reassignment of the variable.

```
def concatStr = concatenateGroovyStrings("SELECT * ", "FROM ", "TABLE_NAME");
def newStr = concatStr;
def query = createPreparedStatement(newStr, "My Query");
```


Copyright © 2026, Oracle and/or its affiliates.

Following on with this one, the prepared statement can combine strings and variables, as long as the variable was created using the concatenateGroovyStrings method.

```
if (restrictByField()){
def fieldClause = concatenateGroovyStrings(" AND FIELD_NAME = 'VALUE'");
}
def query = createPreparedStatement("SELECT * FROM TABLE_NAME WHERE OWNER_FLG
= 'CM'" + fieldClause, "My Query");
```

Handling Elements in an Array

When attempting to build an SQL statement, you may have use cases that require looping through a list and dynamically adding content to the SQL statement from within the list. Additional helper methods are provided for these more complex use cases.

Use Case 1: Include one or more values in an "in" clause.

If you have a dynamic list of values that you want to include in an "in" clause, the following helper method handles generating the bind variables.

```
concatenateSqlInClause(String columnExpr, String bindName, int start, int
count)
```

The columnExpr and bindName should be string literals. Example: you have a list of To Do Types and you want to include that list in an "in" clause.

```
def sql = concatenateGroovyStrings("SELECT TD_ENTRY_ID FROM CI_TD_ENTRY",
                   "WHERE ENTRY_STATUS_FLG <> 'C'",
                   "AND",
         concatenateSqlInClause("TD_TYPE_CD", ":tdType", 1,
tdTypeInputCount),"
         )
```

The result of the above would be the following if there are 5 To Do Types in the list:

```
SELECT TD_ENTRY_ID FROM CI_TD_ENTRY WHERE ENTRY_STATUS_FLG <> 'C'
                   AND TD_TYPE_CD IN
(:tdType1,:tdType2,:tdType3,:tdType4,:tdType5)
```

Use Case 2: Create a Bind Name with an index

The following helper method allows you to create bind names for use in creating an SQL statement:

```
concatenateBindName(String bindName, int index)
```

For example, within a 'for' loop where 'i' is the index name,

```
concatenateBindName(:tdType, i)
```

The results of this method can be added to the SQL statement being prepared using concatenateGroovyStrings.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Use the Script query > Step Information search option to look for base scripts that use the helper method to find examples of this technique.

Use Case 3: Using the index value

There may be use cases where you need the literal value of an index variable. For this use case, the following method is available:

```
literalFor(int i)
literalFor(long l)
```

This may be used to dynamically build the column names, for example when you want to search for records by the prime key value and the number of prime keys is variable. Example:

```
String sql = concatenateGroovyStrings("SELECT * FROM F1_MO_UPD WHERE ")
for (int i = 1; i <= pkCount; i++) {
    sql = concatenateGroovyStrings(sql, "PK_VALUE", literalFor(i), " = ",
concatenateBindName(":pkVal", i))
    if (i < pkCount) {
        sql = concatenateGroovyStrings(sql, " AND ")
    }
```

For an object that has a two-part primary key, the above result is as follows:

```
SELECT * FROM F1_MO_UPD WHERE PK_VALUE1 = :pkVal1
    AND PK_VALUE2 = :pkVal2
```

Plug-In Scripts

> **Note**
>
> This section assumes you are familiar with the notion of plug-in spots (algorithm entities) and plug-ins. See The Big Picture Of Algorithms for more information.

As an alternative to writing a java program for a plug-in spot, the framework supports creating plug-ins using the defined script steps, Xpath commands, Groovy code or a combination of these three options.

The following topics describe basic concepts related to plug-in scripts.

- A Plug-In Script's API
- Setting Up Plug-In Scripts

A Plug-In Script's API

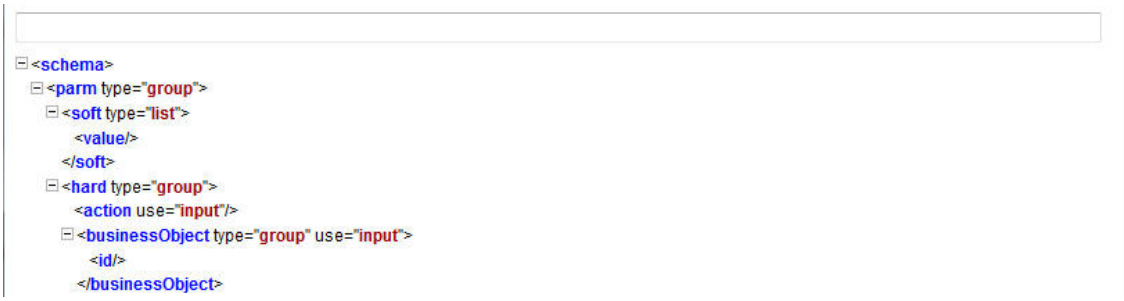
Like program-based plug-ins, plug-in scripts:

- Run on the application server.

Copyright © 2026, Oracle and/or its affiliates.

- Have their API (input / output interface) defined by the plug-in spot (i.e., plug-in scripts do not get to declare their own API).
- Can only be invoked by the "plug-in spot driver".

For plug-ins configured to use a script engine version, the best way to understand the script's API is to use the **View Script Schema** hyperlink to view its parameters data area schema.



Notice the two groups: soft and hard. If you are familiar with plug-in spots, you'll recognize these as the classic soft and hard parameters:

- The **soft** parameters are the values of the parameters defined on the algorithm. Notice they are not named - if you want to reference them in your plug-in script, you must do it by position.
- The **hard** parameters are controlled by the plug-in spot (i.e., the algorithm entity). Notice that this plug-in spot has a single input parameter called " **businessObject/id"**. Also notice the `use=` attribute - this shows that this parameter is input-only (i.e., you can't change it in the plug-in script).

> **Note**
>
> **XPath.** You can click on an element name to see the XPath used to reference the element in your script.

Plug-ins configured to use the **Groovy** engine version do not use an XML interface for the API and instead are processed in the same way as Java algorithms. The framework supplies a dynamically generated superclass that implements the plug-in spot for Groovy objects. Use the **View Script Superclass** hyperlink to view this superclass and the methods to set and get the hard parameters.

> **Note**
>
> For plug-in scripts using the **Groovy** engine version, soft parameters do not appear in the plug-in spot API as defined by the superclass. Plug-ins using only Groovy code define their soft parameters using annotations, in a similar way to Java algorithms, and fetch those values using methods defined in the algorithm code.

Setting Up Plug-In Scripts

The following points describe how to implement a plug-in script:

- Compose your plug-in script, associating it with the appropriate algorithm entity (plug-in spot).

Copyright © 2026, Oracle and/or its affiliates.

- Create a new algorithm type for the respective algorithm entity, referencing your plug-in script as the program to carry out the algorithm type's function. Only plug-in scripts associated with the algorithm entity may be referenced on the algorithm type.
- Set up an algorithm for the respective algorithm type and plug it in where applicable. Refer to Setting Up Algorithm Types for more information.

Service Scripts

BPA scripts run on the client's browser and guide the end-users through business processes. Service scripts run on the application server and perform server-based processing for BPA scripts, zones and more. You may want to think of a service script as a common routine that is set up via scripting (rather than programming).

The following topics describe basic concepts related to service scripts.

- A Service Script's API
- Invoking Service Scripts

A Service Script's API

As with any common routine, a service script must declare its input / output parameters (i.e., its API). A service script's API is defined on its schema.

> **Note**
>
> Refer to Schema Nodes and Attributes for a complete list of the XML nodes and attributes available to you when you construct a schema.

Invoking Service Scripts

Any type of script configured to use a scripting engine version may invoke a service script:

- A BPA script may invoke a service script to perform server-based processing.
- Plug-in scripts may invoke a service script (like a "common routine").
- A service script may call another service script (like a "common routine").

Map zones may be configured to invoke service scripts to obtain the data to be displayed. Refer to Map Zones for more information.

Inbound web services support interaction with service scripts allowing the outside world to interact directly with a service script.

You can also invoke a service script from a Java class.

Groovy Library Scripts

Just as service scripts can define common routines written in scripting language, Groovy library scripts are used to define groups of common components or methods written in Groovy. Groovy library code runs on the application server and performs server-based processing for scripts that utilize Groovy code.

The following topics describe basic concepts related to Groovy library scripts.

- A Groovy Library Script's API

Copyright © 2026, Oracle and/or its affiliates.

- Invoking Groovy Library Methods

A Groovy Library Script's API

A Groovy library script's API is composed of one or more public methods whose code is defined in the script's steps. Those methods are defined in a step type of **Groovy Library Interface**. A Groovy library script must have one (and only one) step of this type.

Invoking Groovy Library Methods

Any type of script that supports the **Groovy Members** step type may invoke common methods defined in Groovy library scripts.

Code within a **Groovy Members** step must create an instance of the Groovy library interface definition to enable the interface methods to be invoked. Refer to the topic Using Groovy Within Scripts for more information.

Debugging Server-Based Scripts

The server can create log entries to help you debug your server scripts.

The logs contain a great deal of information including the contents of the referenced data area for **Move data**, **Invoke business object**, **Invoke business service** and **Invoke service script** steps.

Refer to the Debug Mode topic in the Configuration Tools chapter for details of how to execute the application in debug mode.

Defining Scripts

The script maintenance portal is used to maintain your scripts.

You can access the portal from the **Admin Menu**, then **System**, then **Script**. You are brought to a query portal with options for searching for a specific script. Once a script has been selected you are brought to the maintenance portal to view and maintain the selected record.

Scripts may be of the following types: **BPA Script**, **Plug-In Script**, **Groovy Library Script** and **Service Script**. Refer to The Big Picture Of BPA Scripts and The Big Picture Of Server Based Scripts for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new script, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Script.** This zone provides information about the selected script. Refer to Securing Script Execution for more information.
- **Steps.** Use this zone to review and maintain the script's steps. You may update a single step or maintain all steps as a list. Refer to How To Set Up Each Step Type for an overview of the step types.

The following zones may appear as part of the portal's **Schema** tab page. This tab page only appears for service scripts and BPA Scripts.

Copyright © 2026, Oracle and/or its affiliates.

- **Schema Editor.** This zone allows you to edit the script's schema using the Schema Designer tool. The purpose of the schema is to describe the data elements passed to and from the script.

> **Note**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Tip**
>
> Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **Data Areas** tab page. This tab page does not appear for Groovy Library scripts or plug-in scripts using the **Groovy** engine version.

- **Data Areas.** Use this zone to define the data areas used to pass information to and from the server or any other data area describing your temporary storage. Click on the **View Data Area** action to view the data area's schema in the schema viewer window.

The following zones may appear as part of the portal's **Eligibility** tab page. This tab page only appears for BPA scripts.

- **Script Eligibility.** Use this zone to define a BPA script's eligibility rules. Refer to the guidelines under The Big Picture Of Script Eligibility for more information on how to define this information.

The following zones may appear as part of the portal's **References** tab page.

- **Script References.** This zone list various configuration entities referencing the current script or its schema.

Actions

The page supports the following special actions:

- **View Schema.** Click this link to view the script's expanded schema definition. Doing this opens the schema viewer window. This action is only available for BPA scripts and service scripts.
- **View Script As Text.** Click this link to view the internal scripting commands in a separate window.
- **Test.** Use this action to test your service script of BPA script.
  - For a service script, enter input values and save. The service is performed and output information is presented, but the system issues a roll-back so that the changes are not committed. Note that if your service uses the save-point dispatcher to perform any actions or if it issues a call to an external system, the roll-back does not apply to those actions. If the service issues an error the corresponding message is displayed. Note that the input and output testing user interface is dynamically rendered based on UIHints embedded in the service script's schema. As such, the user interface may be as elaborate or basic as the UIHints in that schema.
  - Testing a BPA script is only applicable to a BPA that does not rely on information passed to it from somewhere else. Because the system cannot detect if this is the

Copyright © 2026, Oracle and/or its affiliates.

case for your BPA, the Test button is always visible. Attempting to use it on a BPA that relies on information passed into it may show an error.

> **Note**
>
> This action is secured by a designated "Test a Service" application service and the "Generic Service Executor" application service and also requires security rights for creating a script, which is the alternative way for testing any service. It is recommended to allow access to this function in non-production environments only.

- **Java Docs.** Use this action to launch the Groovy Javadocs viewer.
- **Third Party Groovy Allowlist.** Use this action to review the allowlist of third party Groovy classes and verify the restricted list of classes available for Groovy coding in the script.
- **View Script Supreclass.** Use this action to view the code of the runtime generated superclass for the related plug-in spot's implementation.. This action appears only for plug-in scripts using an engine version of **Groovy**.
- How To Set Up Each Step Type
- Additional Topics

How To Set Up Each Step Type

The contents of this section describe how to set up each type of step. Note that some step types are only applicable for BPA scripts and some step types are only supported for server based scripts. In addition, some BPA script steps are considered legacy functionality and therefore only supported on fixed pages and not portals. Each section indicates whether there are any script type limitations.

- Conditional Branch Steps
- Display Text Steps
- Edit Data Steps
- Go To Steps
- Height Steps
- Input Data Steps
- Invoke Business Object Steps
- Invoke Business Service Steps
- Invoke Function Steps
- Invoke Map Steps
- Invoke Service Script Steps
- Groovy Imports Steps
- Groovy Library Interface Steps
- Groovy Member Steps
- Label Steps
- Mathematical Operation Steps
- Move Data Steps

Copyright © 2026, Oracle and/or its affiliates.

- Navigate To A Page Steps
- Perform Script Steps
- Press A Button Steps
- Prompt User Steps
- Set Focus To A Field Steps
- Terminate Steps
- Transfer Control Steps

Conditional Branch Steps

Conditional branch steps allow you to conditionally jump to a different step based on logical criteria. For example, you could jump to a different step in a script if the customer is residential as opposed to commercial. In addition, several fields are required for **Conditional Branch** steps:

Compare Field Type and **Compare Field Name** define the first operand in the comparison. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Current To Do Information**. Use this field type when the field being compared resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Data Area**. Use this field type when the field being compared is one that you put into one of the scripts data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the field being compared resides on one of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the field being compared is a global variable.
- **Temporary Storage**. Use this field type when the field being compared is one that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the field being compared resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

Condition defines the comparison criteria:

- Use >, <, =, >=, <=, <> (not equal) to compare the field using standard logical operators. Enter the comparison value using the following fields.
- Use **In** to compare the first field to a list of values. Each value is separated by a comma. For example, if a field value must equal `1`, `3` or `9`, you would enter a comparison value of `1,3,9`.
- Use **Between** to compare the field to a range of values. For example, if a field value must be between `1` and `9`, you would enter a comparison value of `1,9`. Note, the comparison is inclusive of the low and high values.

Comparison Field Type, **Comparison Field Name** and **Comparison Value** define what you're comparing the first operand to. The following points describe each field type:

Copyright © 2026, Oracle and/or its affiliates.

- **Current To Do Information**. Use this field type when the comparison value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Data Area**. Use this field type when the comparison value resides in one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the comparison value resides on one of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the field being compared is a constant value defined in the script. When this field type is used, use **Comparison Value** to define the constant value. Refer to How To Use Constants In Scripts for instructions on how to use constants.
- **Temporary Storage**. Use this field type when the comparison value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the comparison value resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

> **Note**
>
> **Conditional field types.** The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts. Only the field types **Temporary Storage** and **Predefined Value** are applicable when the Condition is **In** or **Between**.

The above fields allow you to perform a comparison that results in a value of **TRUE** or **FALSE**. The remaining fields control the step to which control is passed given the value:

- **If True, Go to** defines the step that is executed if the comparison results in a **TRUE** value.
- **If False, Go to** defines the step that is executed if the comparison results in a **FALSE** value.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Numeric Comparison**. Comparison of two values may be numeric or textual (left-to-right). Numeric comparison takes place only when values on both side of the comparison are recognized as numeric by the system. Otherwise, textual comparison is used. Fields for **Current To Do Information**, **Data Area**, **Page Data Model**, and **User Interface Field** types are explicitly associated with a data type and therefore can be recognized as numeric or not. This is not the case for fields residing in **Temporary Storage** or those set as **Predefined Values**. A **Temporary Storage** field is considered numeric if it either holds a numeric value moved to it from an explicitly defined numeric value (see above) or it is a resultant field of mathematical operation. A **Predefined Value** field is considered numeric if the other field it is compared to is numeric. For example, if a numeric field is compared to a **Predefined Value** the latter is considered numeric as well resulting in numeric value comparison. However, if the two fields are defined as **Predefined Values** the system assumes their values are text strings and therefore applies textual comparison.

Display Text Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Display text steps cause a text string to be displayed in the script area. Steps of this type can be used to provide the user with guidance when manual actions are necessary. In addition, they can be used to provide confirmation of the completion of tasks.

The information you enter in the **Text** field is displayed in the script area when the step is executed.

The text string can contain substitution variables and HTML formatting commands. Also note that for debugging purposes, you can display an entire data area (or a portion thereof) by entering **%+...+%** where **...** is the name of the node whose element(s) should be displayed.

Edit Data Steps

Edit data steps provide a free format region where you can specify commands to control your script processing.

Enter your scripting commands in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for defining the edit data step.

In general, the syntax available within edit data mimics the commands available within the explicit step types. However, there are a few commands that are available only within edit data. For example, the two structured commands: **For**, and **If**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Not all BPA step types are supported using the edit data syntax. Refer to the details below for more information on edit data commands and examples.

The topics in this section provide detail of the syntax supported in the edit data step type.

Comments

You can place comments into your script using the double slash notation `//` in the first two characters of the edit data step. Example:

```
//
// quit with error
//
if ("not(customer/securityEnabled)")
terminate with error (8000, 1001 %1="customer/id" %2='not allowed');
end-if;
```

Temporary Variables

Temporary variables can be declared within all types of scripts. They should be referenced by a leading single dollar sign ('$'). However, temporary variables behave differently in the various script types:

- In BPA Scripts temporary variables remain persistent from one BPA script to another (refer to the Perform Script and Transfer Control statements), which means that you can use temporary variables to communicate between BPA scripts.
- Temporary variables cannot be passed from a BPA script to a service or plug-in script. Only data area elements can be passed between these types of scripts.
- Within service and plug-in scripts, temporary variables remain persistent only for the life of the particular script that declared the variable. This means that temporary variables cannot be passed between plug-in scripts and service scripts, only global variables, context variables and data area elements can be passed between these types of scripts.

Declaring / Initializing / Defaulting Temporary Variables

When using a temporary variable, it should be declared or initialized with an appropriate value before using it. A typical method for declaring a variable is to simply move data to it in a move statement, for example.

```
move "0" to $index;
```

> **NOT_SUPPORTED**
>
> Refer to Move to a Temporary Variable for more information on implicit declaration of a temporary variable within a move statement.

For BPA scripts, as mentioned above, temporary variables may be passed from one BPA script to another. As such, it is common to reference a temporary variable in a BPA that should have been initialized by a previous BPA. However, if there is any reason that a temporary variable did not get initialized by a previous BPA, a reference to it will cause an error. It is good practice,

Copyright © 2026, Oracle and/or its affiliates.

therefore, to use the **default** statement that will initialize temporary variables that are not already created / initialized.

- The following statement will initialize the temporary variable $InputAction – but only if the temporary variable has not yet been initialized:

  ```
  default $InputAction;
  ```

- The following statement will set the value of the temporary variable $InputAction to 'read' – but only if the variable has not yet been initialized:

  ```
  default $InputAction as 'read';
  ```

> **Note**
>
> Scripts should take care not to define variables using a reserved keyword. The following table lists the reserved keywords.

Keyword
add
as
asError
bpa
branch
data
declareBO
declareBS
declareDA
declareMap
declareSS
default
delete
edit
element
else
end-edit
end-for
end-if
error
escape
evaluate
fastAdd
fastUpdate
for
goto
if
in
invokeBO
invokeBS

Copyright © 2026, Oracle and/or its affiliates.

Keyword
invokeMap
invokeSS
label
map
move
navigate
navigateAndReloadDashboard
null
page
performScript
popup
read
readWithoutVersion
replace
suppress
target
terminate
to
transferControl
update
using
warn
with

Context Variables

Context variables are only available within service scripts. The context variable will be available for the duration of the service script and all that it invokes. Therefore, you can use a context variable within a service script to communicate information to a lower level service script or schema. They should be referenced by leading double dollar signs ('$$').

> **Note**
>
> Because context variables are available for lower level scripts, they may sometimes be referred to as global variables or global context variables. But they should not be confused with global variables.

Declaring / Initializing / Defaulting Context Variables

When using a context variable, it should be declared or initialized with an appropriate value before using it. A typical method for declaring a variable is to simply move data to it in a move statement, for example.

```
move 'context variable' to $$contextVariable;
```


Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Move using a Context Variable for more information.

Move Statement

The **move** statement copies a source value to a target. The following table highlights various options supported in the move statement.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move to Element **move "xpath" to "xpath";**	Moves a "tacent" to this Bulle "xpcat" hr to "er" pace. / fo

> **Note**
>
> An XPath expression is surrounded by double quotes.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description

rmattedValue";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Moves a "cement" with X-Path/Concatenate function.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
	erson/lastName)" to "parm/fullName";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Moves a "stem" to the top of the X-Path (subspacing-fluorescence)

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	t,ign.)" to "person/firstName";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Moves a tenant with XPath substring/full-name (Neanmen't

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	ion) " to " person/last Name";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Moves a "stem" to the top of the XPath/substring function."

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	arm/year";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move to Element **move 'literal' to "xpath";** **Note** A literal value is surrounded by single quotes.	Moves a token at using a literal string. account/

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	preferences[type="mail"]/text";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
Move to Element **move 'Boolean' to "xpath";**	Move to element ("accountnumber"). Move to element ("accountnumber"). Move to element ("accountnumber").

Statement	Example Syntax Explanation
	true' to "account/hasDebitBalance";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	end-if;

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Moving a expression, which has a Boolean.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Note that the "filter" on the example below is located

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	<n/a><c/o><u/n><d><e></schema> move "boolean(a

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	count[balance>0])" to "account/hasDe

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	bitBalance";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move to Group **move "xpath" to "xpath";**	Move "a" to "c" element/cust from "e" to "w" then sother.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	This system attaches the equivalent to the elements in a

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	mgs and the mengroup/olvis t"ance sofurnotm/ceus curcfes/che"m

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	at to the "parsons/chem" a. For lists and groups, the de

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Exception
	fault behavior is to move all elements within the s

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	source group/list to the target group/list (eveni

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	f the yare not defined in the target group/list). Th

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	ebusinessserviceF1-MoveByNamemaybeusedform

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	or regranular control of the move (without having t

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	odefineeachindividualmovestatement).

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move using a Temporary Variable	When moving to a temporary location (Temporary Variable, Temporary Variable)

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	[illegible]

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	arsonCe;ildCount;

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	When removing Arcomatic Boilar Vessel "Ptahem/Ariab

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Less than "Value" by double quotes. **move" $variab**

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	le "to “xpath”;

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move using a Context Variable **move "xpath" to $$variable;** **move $$variable to “xpath”;**	Context variables, source or target, are referenced

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	cardinal hour/any double quotes. move from a c

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
	ontext variable.move $ $contextVari

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	able to "MarketMessage/sender";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move using a Dynamic Location **move "xpath" | 'literal' to evaluate("xpath" | $variable);** **move evaluate("xpath" | $variable) to "xpath" | $variable;**	The evaluate statement allows you to move a dynamic location

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	ExampleDescription

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	"evaluate("

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
	schemaLocation") to $Variable;// mo

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
	ve evaluate($Variable) to "schemaLoc

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	ation";

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move escape **move escape("xpath" | $variable) to "xpath" | $variable;**	Moves escape (as only valid location) to "xpath"

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	and a blue-; in/scripts. The case (as a statement) scans

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Use the Location Extraction Function for HTML content and escapes

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	it, i.e. replaces any HTML-like characters with

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Description
	special characters that are escaped from HTML rend

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	ering. By doing so the text would be displayed as pl

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	a inline text when displayed as part of an HTML element.

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
Move null **move null to "xpath";**	Removes a node in a full or a fit or a child to the XML instance

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Document (chrsolugmethe/ssecurialsyrtaxbolfmove')nulg'

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Note that you can specify either an outline or the

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	Use the expression to ring, or upon a Remove, or Use the specified Load

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	resource tag error to the ordered with a handle such as "ffix" or "element" (with "lobee

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Description
	limit (actual from mpr/sec sub-ring. Enabled) ") mov

Copyright © 2026, Oracle and/or its affiliates.

Statement	Example Syntax Explanation
	end null to "customer/*"; end-if;

Go To Statement

The edit data step supports functionality analogous to the Go To step type. The syntax is **goto label;** where the label represents another location within the edit data text field (identified by this label) or represents another step in the script.

Copyright © 2026, Oracle and/or its affiliates.

The following is an example of going to another location in the same step identified by the label **addSpouse**.

```
if ("string(parm/spouse/name) != $BLANK")
  goto addSpouse;
end-if;
addSpouse: invokeBO 'Person' using "parm/spouse" for add;
```

The following is an example of going to a different step within the same script. The step sequence is the reference used as the label.

```
if ("string(parm/spouse/name) != $BLANK")
  goto 110;
end-if;
.
.
.
110: invokeBO 'Person' using "parm/spouse" for add;
```

Conditional Branch Statement

The edit data step supports functionality analogous to the Conditional Branch step type. The syntax is **branch (“xpath”) goto label else label;** where:

- The XPath condition in the **branch** statement must evaluate to a Boolean value of True or False.
- The targets for the **goto** and **else** statements are labels that represent another location within the edit data text field (identified by this label) or represent another step in the script.

The following example uses labels for **addSpouse** and **addAccount**

```
branch ("string(parm/spouse/name) != $BLANK") goto addSpouse else addAccount;
```

If Statement

The **if** statement is similar to the conditional branch statement. Either can be used to structure the logic of your script. This statement may optionally include an **else** statement but it should always end with an `end-if` statement.

> **Note**
>
> This is an example of a statement that is not represented as a separate step type. It is only available within the edit data text.

The syntax is **if (“xpath”) else end-if;**. The XPath condition must evaluate to a Boolean value of True or False. The following are some examples.

Example where the XPath contains a simple logical condition.

```
if ("string(parm/spouse/name) != $BLANK")
  //
  // Create spouse since spouse name present
  goto addSpouse;
else
```


Copyright © 2026, Oracle and/or its affiliates.

```
  //
  // Create account without spouse
  goto addAccount;
end-if;
```

Example where the XPath contains a complex condition.

```
if ("string(parm/spouse/name) != $BLANK and string(parm/hasSpouse) = true or
boolean(parm/requireSpouse)")
  //
  // Create spouse since spouse name present
  goto addSpouse;
end-if;
```

Example of a stacked set of statements used to evaluate multiple possible values of a field.

```
if ("parm/rowCount = 0")
  //
  // no rows found
  goto quit;
end-if;
if ("parm/rowCount = 1")
  //
  // one row found
  goto process;
end-if;
if ("parm/rowCount > 1")
  //
  // more than one row found
  goto quit;
end-if;
quit: terminate;
```

The following XPath shows Boolean based on the existence of the node. In this example, if the node exists in the XML instance document being processed, the statement will evaluate to True. If no element is found, the statement evaluates to false.

> **Note**
>
> When treating XPath nodes as Boolean variables be aware that an empty node evaluates to True. Only a missing node return False.

```
if ("boolean(parm/spouse/name)")
  goto addSpouse;
else
  //
  // Create account without spouse
  goto addAccount;
end-if;

if ("not(parm/spouse/name)")
  //
```


Copyright © 2026, Oracle and/or its affiliates.

```
  // Create account without spouse
  goto addAccount;
else
  goto addSpouse;
end-if;
```

For Statement

The **for** statement creates a list of nodes or values depending on your XPath expression. If you specify a list node then every child node of the list, along with its contents, will be available within the loop. If you specify a child node directly, then a list of values only will be available within the loop.

> **Note**
>
> For more information on creating new entries in a list, please refer to the creating a new list instance example.

> **Note**
>
> This is an example of a statement that is not represented as a separate step type. It is only available within the edit data text.

The syntax is **for ($variable in "xpathList") end-for;**. The XPath condition must evaluate to a Boolean value of True or False.

The following examples are based on this sample schema:

```
<schema>
  <SAList type="list">
    <id/>
    <balance/>
  </SAList>
  <SAContributor type="list">
    <id/>
  </SAContributor>
</schema>
```

Example that specifies the list node in the XPath expression where all child nodes are available for processing.

```
move "0" to $AccountBalance;
move "0" to $index;
for ($SAList in "SAList")
    move "$SAList/balance + $AccountBalance" to $AccountBalance;
    //
    // keep track of each SA contributing to the balance in the SA
Contributor list
    move "1 + $index" to $index;
    move "$SAList/id" to "SAContributor[$index]/id";
end-for;
```


Copyright © 2026, Oracle and/or its affiliates.

Example that specifies a child node within the list node in the XPath expression. Only values of that node are available for processing.

```
move "0" to $AccountBalance;
for ($SABalance in "SAList/balance")
    move "$SABalance + $AccountBalance" to $AccountBalance;
end-for;
```

Example that shows that a filter can be used to limit the rows selected by the **for** loop.

```
move "0" to $AccountDebitBalance;
for ($SAList in "SAList[Balance>0]")
    move "$SAList/balance + $AccountDebitBalance" to $AccountDebitBalance;
end-for;
```

Example that shows the use of a filter when specifying child nodes.

```
move "0" to $AccountCreditBalance;
for ($SABalance in "SAList[Balance<0]/balance")
    move "$SABalance + $AccountCreditBalance"
to $AccountCreditBalance;
end-for;
```

List Processing

This section provides details about processing lists. The examples in this section reference the following schema:

```
<schema>
  <parm type="group">
    <name/>
  </parm>
  <Person type="group">
    <names type="list">
      <type/>
      <name/>
    </names>
  </Person>
</schema>
```

Referencing a List Element. You can move a value to a particular list instance by referencing an identifying node in the list within a filter. The syntax is **move "xpath" to "xpathList[filter]/element";** Example:

```
move "parm/name" to "Person/names[type='main']/name";
```

Creating a New List Instance. A special notation can be used within a move target statement to indicate a new list instance should be created. The "+" indicates to the script processor that a new instance of a list should be initiated for the target element. The syntax is **move "xpath" to "+xpathList";** Example:

```
move "parm/name" to "Person/+names/name";
```


Copyright © 2026, Oracle and/or its affiliates.

Deleting a List Instance. An XML list entry can be deleted from the database by moving an action attribute of 'delete' to the element name. To cause a database delete of a list entry requires an attribute of action="delete" in the target node and a subsequent update BO interaction. The syntax is **move 'delete' to "xpathList@action");** Example:

```
if ("parm/action = 'd'")
    move "0" to $index;
    for ($CCList in "CCList")
        move "1 + $index" to $index;
        if ("$CCList/id = parm/id")
            move 'delete' to "CCList[$index]@action";
            goto update;
        end-if;
    end-for;
end-if;
```

The following shows the resulting XML.

```
<root>
  <CCList>
    <id>9876538976</id>
    <balance>309.98</balance>
  </CCList>
  <CCList action="delete">
    <id>4321125899</id>
    <balance>87.45</balance>
  </CCList>
</root>
```

> **Note**
>
> Deleting a list instance through use of the action attribute is risky if iterative BO interactions are required. The XML document that contains the list instance to be deleted will not be altered after a successful BO interaction, which means the document will still contain the list instance even though it no longer exists. To solve this problem, it is essential to re-read the BO after any BO update where the action attribute of 'delete' has been used.

> **Note**
>
> An alternative to the delete attribute described here, is to use the BO action of replace. Manipulating a list to use the replace action avoids the problem described above concerning stale information in request documents post BO update.

Functions for Processing a List

XPath provides several functions that are useful to process elements of a list including **count**, **sum** and **last**.

Copyright © 2026, Oracle and/or its affiliates.

The following examples are based on this sample XML document:

```
<xml>
  <ft>
    <type>bill</type>
    <date>20100101</date>
    <amt>30.30</amt>
    <cat>tax</cat>
  </ft>
  <ft>
    <type>adj</type>
    <date>20100301</date>
    <amt>20.20</amt>
    <cat>int</cat>
  </ft>
  <ft>
    <type>bill</type>
    <date>20100201</date>
    <amt>10.10</amt>
    <cat>tax</cat>
  </ft>
</xml>
```

The following is an example of a sum. The syntax is **move "sum(xpathList/element)" to $variable;** The example sums the total balance.

```
move "sum(ft/amt)" to $TotalBalance;
```

The following is an example of a sum using a filter to get a subtotal. The example sums the balance of the entries that have the ‘tax’ category.

```
move "sum(ft[cat='tax']/amt)" to $TaxBalance;
```

The following is an example of a count. The syntax is **move "count(xpathList)" to $variable;** The example finds the count of the number of FT entries in the list.

```
move "count(ft)" to $TranCount;
```

The following is an example of ‘last’, which is used to locate the last entry. The syntax is **move "last(xpathList)" to $variable;** The example finds the last amount in the FT list.

```
move "ft[last()]/amt" to $LastAmount;
```

Declare and Invoke Schema Based Objects

You can invoke a business object, business service or service script within the edit data step. To support the dynamic invoke, a dynamic data area name can be declared.

The schema being declared may be a business object (BO) schema, a business service (BS) schema, a service script (SS) schema, data area (DA) schema or a UI map schema. The declare statement will differ based on the type of schema, but the syntax is analogous.

- **declareBO 'BO Name' | $variable | "xpath" as 'DynamicDataArea';**

Copyright © 2026, Oracle and/or its affiliates.

- **declareBS 'BS Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareSS 'SS Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareDA 'DA Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareMap 'Map Name' | $variable | "xpath" as 'DynamicDataArea';**

When invoking a BO, BS or SS, the name of the object can be specified as a literal or it can be a value contained within an element or a variable. For every Invoke, you must supply an XPath reference to a group name.

When invoking a business object, an action must be supplied. The syntax is **invokeBO 'BO Name' | $variable | "xpath" using "xpath" for action;**. The valid actions are as follows:

- **read**. This action reads the current view of the BO data.
- **add**. This action will add the object and read and return the resulting view of the BO.
- **fastAdd**. This action will add the object but does not perform a subsequent 'read' to return the resulting view of the BO. This option is better than add for performance purposes if there is no reason to re-read the record.
- **update**. This action will update the object and read and return the resulting view of the BO. This action executes a 'merge' of the information specified in the invoke statement's request XML document with existing BO data. Using this action allows the script to only indicate the elements that are changing.
- **fastUpdate**. This action will update the object but does not perform a subsequent 'read' to return the resulting view of the BO. This option is better than update for performance purposes if there is no reason to re-read the record.
- **delete**. This action deletes the object.
- **replace**. This action is an alternate to the update action. The replace action completely replaces existing BO data with the information in the request document. Typically, the replace action is used when a BO contains a list because it is easier to simply replace all instances of a list rather than attempt a list merge, which requires special logic to delete a list instance explicitly.

> **Note**
>
> The replace action must be used when using the UI map functionality to Upload a CSV File.

> **Note**
>
> The replace action is currently not supported for any maintenance object that is maintained in a 'fixed' page that uses a list metaphor to show all the records in the page at once. Currency is an example of this type of page.

Examples:

```
invokeBO 'BusinessObject' using "dataArea" for fastAdd;

invokeBO $variableBO using "dataArea" for fastUpdate;

invokeBO "daName/boElement" using "dataArea" for replace;
```


Copyright © 2026, Oracle and/or its affiliates.

The syntax of the invoke statements for both a business service and service script are similar. The BS / SS is specified along with the XPath reference to the group name:

- **invokeBS 'BS Name' | $variable | "xpath" using "xpath";**
- **invokeSS 'SS Name' | $variable | "xpath" using "xpath";**

The examples use the **invokeBS** statement but the statements are similar for the **invokeSS** statement.

```
invokeBS 'BusinessService' using "dataArea";

invokeBS $variableBS using "dataArea";

invokeBS "daName/bsElement" using "dataArea";
```

BO Warnings. Note that for BPA scripting, the **invoke** statements may also indicate how to handle warnings.

Syntax	Description	Examples

`with warn asError`	Indicates that a warning should be treated as an error displayed in the UI map. The text **asError** is optional.	`invokeBO 'BusinessObject' using "dataArea" for add with warn asError;`
		
		`invokeSS 'ServiceScript' using "dataArea" with warn;`

`with warn popup`	Indicates that a warning should be presented in the standard framework popup. In this scenario, the user is presented with standard OK and Cancel buttons. If the user clicks OK, it means that the process should continue. If the user clicks Cancel, the processing should discontinue. This is the recommended setting.	`invokeBS "daName/bsElement" using "dataArea" with warn popup;`

`with warn suppress`	Indicates that a warning should be suppressed. This is the default if no warning syntax is added to the **invoke** statement.	`invokeBS "daName/bsElement" using "dataArea" with warn suppress;`
		
		`invokeSS 'ServiceScript' using "dataArea";`

> **Note**
>
> For service scripts, all objects invoked from the service script will inherit their warning level. Therefore, if the service script is invoked **with warn**, all nested invoke statements will also be invoked **with warn**.

Copyright © 2026, Oracle and/or its affiliates.

For BPA scripts, there should also be logic following the invocation to handle errors and warnings (if **with warn as popup** is used). The system variable `$WARNING` is set to **true** if the user has clicked the Cancel button in the warning popup. If a map was shown, the logic should redisplay the map (where the warning message will be redisplayed.) This allows the user to potentially make changes and save again. If no map was shown prior to the warning, then the logic should terminate.

The system variable `$ERROR` indicates that an error was received. If a map was shown prior to the error, then the logic should redisplay the map where the error will be shown. If no map was shown, then the product provides a BPA Script `F1-HandleErr` that should be used to display the error. The following is an example of typical error handling logic.

```
invokeBO "F1-DetermineBo/output/bo" using "boSchema" for update with warn
popup;
if ("$WARNING")
   if ("map_schema/action = 'DEL'")
     terminate;
   else
     goto maintMap;
   end-if;
end-if;
if ("$ERROR")
  if ("map_schema/action != 'DEL'")
    goto maintMap;
  else
    transferControl 'F1-HandleErr';
  end-if;
end-if;
```

System and Global Variables

The following tables highlight system and global variables available for script writing.

System Variables - All Script Types

The following system variables are available for all script types (service scripts, plug-in scripts, and BPA scripts).

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$BLANK	Reinfres(sentsaingmptym/sp.ouse/name) != $BLANK")

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	goto addSpouse;end-if;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$CURRENT-DATE	Reprovesents the current date. For BPA scripts, this is the b

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	rowserdate. For serverscripts this is the serverdate (an

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	disaffected by the system data overrides log).

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$CURRENT-STD-DTTM	Represents the current date-time expressed in standard Data Terminal

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	time (meaning without any adjustments for summer time/day

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	lightsavingstime).

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$DEVICE-OS	Removes "$DEVICE-OS" device operating system.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$DEVICE-BROWSER	Removes "Internet Explorer" from $DEVICE. $DEVICE-BROWSER

Variable	Description
	wser;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$DEVICE-DISPLAY-TYPE	Reprovesents the user's device-display-type when the Verifi

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	Displays Type: Medium or Small size. Returned values m

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	ayb elikeoraDesktop,oraTabletandoraPhone.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$DEVICE-INFO	Provides the Combination of all three device properties (Description, Inventory, Location)

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	DEVICE-OS, DEVICE-BROWSER and DEVICE-DISPLAY-TYPE) an

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	each property value is separated by semi-colon.

System Variables - BPA Scripts Only

The following system variables are only available / applicable for BPA script types.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$DOUBLE_QUOTE	Represents a double-quote. $DOUBLE_QUOTE" to $tempField;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$SINGLE_QUOTE	Represents an apostrophe. "Use $SINGLE_QUOTE" to $tempField;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$SPACE	Contains a single space value. Field;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$SYSTEM-DATE	Reprovesents $SYSTEM-DATE. Note that this date is affected

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	by the system date overide logic)

System Variables - Server Scripts Only

The following system variables are only available / applicable for service script and plug-in script types.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$ADDITIONAL-IP-INFO	Am HTTP "request" $ADDITIONAL-IP-INFO "additional" $ADDITIONAL-IP-INFO "address/s" "header"

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	Example: Exam Profile Tables Map Add or Result. Based on implementation if there

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	is some information available on the proxy server or load b

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	balancer, such as the originating IP address.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$CURRENT-DTTM	Represents the current date-time. $CURRENT-DTTM-time. $tempDateTime;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$F1-INSTALLATION-TIMEZONE	Represents the time zone of the Oracle Fine Mortgage Instalation

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	Exampleption
	Zone;

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$LANGUAGE	Represents the language code that is used by Typical

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	ly this is the user's default language.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$PROCESS-DATE	Reprocess "Present" $PROCESS-DATE. The process date differs from

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	mthecurrentdatebecausetheprocessdatewillremaincon

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	sistent throughout the duration of the process being exec

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	uted. For example, if a service script stores several busi

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	nessobjects–theprocessdateisinitializedatthestart

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	of the service script executed for each business object w

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	will have the same processed default ted. The current date

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	, especially the current date time, will reflect the actual

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	time of processing.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$PROCESS-DTTM	Represents the Process Date-time. Note that the Process Date-time is not the Data Time.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	date and time is initialized at the start of a particular pro

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	cessandwillnotreflectthe exactdateandtimeofanupdat

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	e.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$REQUESTING-IP-ADDRESS	Represents the IP address from the HTTP "request". Not/that

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	Exampl ifthere/suyestemIpuAddthrou"ghaproxyserverorloadbala

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	ncer, this IP address is the IP address of the proxy or loa

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	dbalancer, not the IP address of the end user. Refer to the

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$ADDITIONAL-IP-INFO	variable for ip information.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
$USER	Represents the User ID of the user executing the script.

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	a tneedtoknowtheuserIDofthecurrentuser, refertoUsef

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
	Multi Tips and Tricks.

Global Variables

BPA scripts and service scripts have access to the values defined in Global Context.

When a BPA script is launched from the user interface, these variables will be automatically initialized. They may be referenced with a single dollar sign in front of the field name. For example if PER_ID is a supported global variable, then $PER_ID can be referenced within the BPA script:

```
move "$PER_ID" to "schema/customerId";
```

For service scripts, global variables may only be referenced if the service script has been invoked directly from a BPA script or a zone on a portal. When a service script is invoked from a BPA script or portal zone, it will have access to the suite of global context variables populated in the UI session. For service scripting, the global fields must be prefixed by two dollar signs (instead of one like in BPA scripting). For example if PER_ID is a supported global context variable, then $$PER_ID can be referenced within the service script.

```
move $$PER_ID to "schema/customerId";
```

> **Note**
>
> As described in Context Variables, a service script may declare context variables that use the same two dollar sign syntax.

Copyright © 2026, Oracle and/or its affiliates.

Perform Script and Transfer Control Statements

The edit data step supports functionality analogous to the Perform script step type and the Transfer Control step type. These are both applicable only to BPA scripts.

Syntax	Valid Values	Comments
`performScript`	'BPA Script Name'	Script to perform is explicitly provided.
	$Variable	Script to perform is found in a variable.
	"XPath"	Script to perform is found in an element, referenced by its XPath.
`transferControl`	Analogous to the **performScript** statement	

> **Note**
>
> When the script named in the **performScript** statement has finished, control will be returned to the calling BPA script. When the script named in the **transferControl** statement has finished, you will not be returned to the calling script, complete control will be granted to the transferred to script.

Navigate Statement

The edit data step supports functionality analogous to the Navigate to a page step type. This is applicable only to BPA scripts.

Syntax	Valid Values	Comments
`navigate`	'Navigation Code'	Navigation option is explicitly provided.
	$Variable	Navigation option is found in a variable.
	"XPath"	Navigation option is found in an element, referenced by its XPath.

In addition, the edit data step supports the ability to indicate that the sidebar should be refreshed when navigating. This is only applicable to BPA scripts.

Syntax	Valid Values
`navigateAndReloadDashboard`	Analogous to the **navigate** statement

Declare BO with BO Group

This statement is specific to BPA scripts that plan to use the base script Main BO Maintenance Processing (`F1_MainProc`) for its Generate Edit Map statements. This script expects that the data used to display in the map is within a **boGroup** tag.

Syntax	Valid Values	Comments
`declareBOWithBOGroup`	'BO Name'	BO is explicitly provided.
	$Variable	BO is found in a variable.
	"XPath"	BO is found in an element, referenced by its XPath.

The following table highlights additional syntax for this statement.

Copyright © 2026, Oracle and/or its affiliates.

Syntax	Valid Values
as	'Dynamic Schema Name'

Examples:

```
declareBOWithBOGroup 'BusinessObject' as 'newMapSchema';

declareBOWithBOGroup $variableBO as 'newMapSchema';

declareBOWithBOGroup "daName/boElement" as 'newMapSchema';
```

Invoke Map Statement

The edit data step supports functionality analogous to the Invoke map step type. This is applicable only to BPA scripts.

Syntax	Valid Values	Comments
invokeMap	'Map Name'	UI Map is explicitly provided.
	$Variable	UI Map is found in a variable.
	"XPath"	UI Map is found in an element, referenced by its XPath.

The following table highlights additional syntax for this statement.

Syntax	Valid Values	Comments
using	"Data Area group name"	Indicates the data area to be passed to and from the server when rendering the HTML form associated with the map.
target	bpa	
	page	
	popup	

Refer to the Invoke map step type for more information about the **target** values.

If the UI map is configured to return a value, then it can be evaluated using the $MAP-VALUE variable.

```
invokeMap 'UI Map' using "dataArea";

invokeMap $variableMap using "dataArea";

invokeMap "daName/mapElement" using "dataArea" target bpa;

// $MAP-VALUE is a variable returned by the invoked map.
if ("$MAP-VALUE='continue' ")
    goto 300;
else
    terminate;
end if;
```


Copyright © 2026, Oracle and/or its affiliates.

Generate Edit Map Statements

The 'generate edit map' statements are used to dynamically generate and launch a UI edit map based on a schema definition. The schema used may be a BO schema, a BS schema, an SS schema or a DA schema. This is applicable only to BPA scripts. The generate statement will differ based on the type of schema, but the syntax is analogous.

Syntax
`generateBOEditMap`
`generateBSEditMap`
`generateSSEditMap`
`generateDAEditMap`

The BO code / BS code / SS code / DA code may be specified using a literal (surrounded by single quotes), as a temporary variable or by referencing an XPath schema location (surrounded by double quotes).

The following table highlights additional syntax for this statement.

Syntax	Valid Values	Comments
`using`	"Data Area group name"	Indicates the data area to be passed to and from the server when rendering the HTML form associated with the map.
`target`	`bpa`	
	`page`	In Redwood view mode this option presents the form in the large drawer panel.
	`popup`	
	`sdrawer`	Presents the form in the small drawer panel.
	`mdrawer`	Presents the form in the medium drawer panel.
	`ldrawer`	Presents the form in the large drawer panel.

The target values indicate where the generated map should be displayed as described in the Invoke map step type. If the UI map is configured to return a value, then it can be evaluated using the `$MAP-VALUE` variable.

The examples use the **generateBOEditMap** but the statements are similar for the other schema types.

```
generateBOEditMap 'BO Name' using "dataArea";

generateBOEditMap $variableMap using "dataArea";

generateBOEditMap "daName/mapElement" using "dataArea" target bpa;

generateBOEditMap "daName/mapElement" using "dataArea" target sdrawer;

generateBOEditMap "daName/mapElement" using "dataArea" target $drawerSize;
```


Copyright © 2026, Oracle and/or its affiliates.

```
// $MAP-VALUE is a variable returned by the invoked map.
if ("$MAP-VALUE='continue' ")
    goto 300;
else
    terminate;
end if;
```

Terminate Statement

The edit data step supports functionality analogous to the Terminate step type.

The following is an example of a simple **terminate** step that will stop the script.

```
if ("not(parm/spouse/name)")
  terminate;
else
  goto addSpouse;
end-if;
```

The **terminate with error** statement is only available in a service script.

> **Note**
>
> For issuing an error from a BPA script refer to Useful Tips and Tricks.

Syntax	Attributes	Comments
`terminate with error (x, y %n= element= )`	'x' represents the message category	Required.
	'y' represents the message number	Required.
	%n="Element XPath" or %n='literal'	Specify the substitution parameters supported by the message using either literal values or XPath references. When the value to substitute is an element within a list, use the XPath of the element within the list qualified by the list occurrence.
	element='Element XPath' or element=$variable	Optionally specify an element name within a UI map to highlight as part of the error. When the element in error is within a list, use the XPath of the element within the list qualified by the list occurrence.

Example of a simple field:

```
if ("string(customer/lastName) = $BLANK")
  terminate with error (8000, 1001 %1="customer/lastName" %2='Last name
required' element='customer/lastName');
end-if;
```


Copyright © 2026, Oracle and/or its affiliates.

Example of terminate where the element to mark is in a list:

```
for ($list in "parm/hard/newBusinessObject/listName")
  if //** check some condition for elementName
      terminate with error (11000, 11000 %1="$list/elementName"
element='$list/elementName');
  end-if;
end-for;
```

> **NOT_SUPPORTED**
>
> For more information on presenting errors in a UI map, please refer to Display Errors.

Invoking Groovy Code

If you have a plug-in script or a service script that mixes XPath scripting and Groovy scripting, the edit data step supports the ability to invoke Groovy code using the syntax **invokeGroovy 'method';** where 'method' is the name of a method defined in a **Groovy Members** step within the script. Only methods that do not receive arguments and return void can be invoked in this way. However, the method invoked from the edit data step can be supported by additional Groovy code in other **Groovy Members** step types.

Example Edit Data step:

```
invokeGroovy 'invoke';
```

Example Groovy Members step:

```
void invoke() {
  initParms()
  readBO()
  initConfig()
  retrieve()
  updateBO()
}
```

> **Note**
>
> As mentioned in Using Groovy Within Scripts, if this is a plug-in script and the code is only Groovy, you should use the **Groovy** script engine version rather than the above technique. This is considered more efficient as it avoids the conversion of the data to and from the XML interface.

Debugging a BPA Script

If a BPA script has height greater than zero, then selected nodes of the script's data area can be displayed at runtime. The XML data is displayed during script execution within the BPA script's display area. Specify the XPath of an XML node from any of the BPA script's data areas, between the paired characters: '%+' and '+%'.

For example, the entire contents of the schema group node named 'input', and the specific contents of the schema element named 'output/status' will be displayed in the BPA script's

Copyright © 2026, Oracle and/or its affiliates.

display area. The debug text must be entered into the BPA script's text area and not within the script's edit data field. Debug text can be declared for any explicit step of the script.

```
display input: %+input+% , and output status: %+output/status+%
```

Useful Tips and Tricks

This section provides some suggestions about how to implement some common logic.

Finding the Logged In User in a BPA

For server based scripts, the $USER variable is populated with the current logged in user. This variable is not available for BPA scripts. The BPA can call the service script `F1-GetUser` if this information is needed.

Issuing an Error from a BPA script

For server based scripts, you use the **terminate with error** statement to return an error message to display to the user. The BPA can display that error by transferring control to `F1-HandleErr` as described in the Declare and Invoke Schema Based Objects. This is not supported in a BPA script. A technique that is available to a BPA for issuing an error is to call the business service `F1-RethrowError` passing the error details. This business service populates the appropriate system error fields needed to issue the error. Then transfer control to `F1-HandleErr`.

```
  if ("string($assignedToUser) != string(F1-GetUser/user)")
    declareBS 'F1-RethrowError' as 'errorMsg';
    move '11010' to "errorMsg/messageCategory";
    move '11511' to "errorMsg/messageNumber";
    move "$toDoEntryId" to "errorMsg/messageParameters/+parameters/
parameterValue";
    invokeBS 'F1-RethrowError' using "errorMsg";
    transferControl 'F1-HandleErr';
  end-if;
```

Go To Steps

Go to steps allow you to jump to a step other than the next step. Additional fields required for **Go To** steps:

Next Step defines the step to which the script should jump.

Height Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Copyright © 2026, Oracle and/or its affiliates.

Height steps are used to change the height of the script area to be larger or smaller than the standard size.

The following additional fields are required for **Height** steps:

Script Window Height defines the number of **Pixels** or the **Percentage** (according to the **Height Unit**) that the script window height should be adjusted. The percentage indicates the percentage of the visible screen area that the script area uses. For example, a percentage value of `100` means that the script area will use the entire area.

> **Note**
>
> **Standard Number of Pixels**. The default number of pixels used by the script area is `75`.

> **Note**
>
> **Adjust script height in the first step**. If you want to adjust the height of the script area, it is recommendation to define the **height** step type as your first step. Otherwise, the script area will open using the standard height and then readjust, causing the screen to redisplay.

> **Note**
>
> **Hide script area**. You could use this type of step to set the height to `0` to hide the script area altogether. This is useful if the script does not require any prompting to the user. For example, perhaps you define a script to take a user to a page and with certain data pre-populated and that is all.

> **Note**
>
> **Automatically close script area**. If you want the script area to close when a script is completed, you could define the final step type with a height of 0.

Input Data Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Copyright © 2026, Oracle and/or its affiliates.

Input data steps cause the user to be prompted to populate an input field in the script area. The input value can be saved in a field on a page or in temporary storage. A **Continue** button always appears adjacent to the input field. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Input Data** steps:

Destination Field Type and **Destination Field Name** define where the input field will be saved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model**. Use this field type to put the input field into a field that resides on any of the tab pages in the object display area (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage**. Use this field type to put the input field into temporary storage. Use **Field Name** to name the field in temporary storage. Refer to How To Name Temporary Storage Fields for more information.
- **User Interface Field**. Use this field type to put the input field into a field that resides on the currently displayed tab page. Note, if you want to execute underlying default logic, you must populate a **User Interface Field**. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

Invoke Business Object Steps

Invoke business object steps allow you to interact with a business object in order to obtain or maintain its information.

The following additional fields are required for **Invoke business object** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts. Refer to BO warnings for more information about handling warnings when invoking a BO.

Group Name references the data area to be passed to and from the server when communicating with the **Business Object**. Indicate the **Action** to be performed on the object when invoked. Valid values are **Add, Delete, Fast Add (No Read), Fast Update (No Read), Read, Replace, Update**.

> **NOT_SUPPORTED**
>
> Refer to BO Actions for more information about the various actions.

Copyright © 2026, Oracle and/or its affiliates.

The business object call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a business object is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

Invoke Business Service Steps

Invoke business service steps allow you to interact with a business service.

The following additional fields are required for **Invoke business service** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

Group Name references the data area to be passed to and from the server when the **Business Service** is invoked.

The business service call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a business service is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use a step that invokes one of the above configuration tool objects in a script rather than defining a function.

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Invoke function steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class.

> **NOT_SUPPORTED**
>
> You must set up a function before it can be referenced in a script. Refer to Maintaining Functions for the details.

The following additional fields are required for **Invoke Function** steps:

Function defines the name of the function. The function's **Long Description** is displayed below.

When a function is invoked, it will either be successful or return an error. The next two fields control the step to which control is passed given the outcome of the function call:

- **If Success, Go to** defines the step that is executed if the function is successful.
- **If Error, Go to** defines the step that is executed if the function returns on error. Refer to How To Use Constants In Scripts for a list of the global variables that are populated when a function returns an error.

> **Note**
>
> **Error technique.** If a function returns an error, we recommend that you invoke a step that transfers control to a script that displays the error message information and stops (note, the error information is held in global variables). You would invoke this script via a **Transfer Control**.

The **Send Fields** grid defines the fields whose values are sent to the function and whose field value source is not **Defined On The Function**. For example, if the function receives an account ID, you must define the name of the field in the script that holds the account ID.

- **Field** contains a brief description of the field sent to the function.

Copyright © 2026, Oracle and/or its affiliates.

- **Source Field Type** and **Mapped Field / Value** define the field sent to the function. Refer to the description of Source Field under How To Set Up Move Data Steps for a description of each field type.
- **Comments** contain information about the field (this is defined on the function).

The **Receive Fields** grid defines the fields that hold the values returned from the function. For example, if the function returns an account's customer class and credit rating, you must set up two fields in this grid.

- **Field** contains a brief description of the field returned from the function.
- **Destination Field Type** and **Mapped Field** define the field returned from the function. Refer to the description of Destination Field under How To Set Up Move Data Steps for a description of each field type.
- **Comments** contain information about how the field (this is defined on the function).

Invoke Map Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Invoke map steps are used to invoke a UI Map to display, capture and update data using an HTML form. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Invoke map** steps:

Group Name references the data area to be passed to and from the server when rendering the HTML form associated with the **Map**.

Use **Target Area** to designate where the map will be presented.

- Select **BPA Zone** if the map should be presented within the script area.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

- Select **Page Area** if the map should be presented in the object display area, i.e. the frame typically used to house a maintenance page. In Redwood view mode this option presents the form in the large drawer panel.
- Select **Pop-up Window** if the map should be launched in a separate window.
- Select **Small, Medium, Large Drawer** if the map should be launched in a drawer panel.

The **Returned Values** grid contains a row for every button defined on the map.

- **Returned Value** is the value returned when the user clicks the button.

Copyright © 2026, Oracle and/or its affiliates.

- **Use as Default** can only be turned on for one entry in the grid. If this is turned on, this value's Next Script Step will be executed if the returned value does not match any other entry in the grid. For example, if the user closes a pop-up (rather than clicking a button), the default value will be used.
- **Next Script Step** defines the step to execute if the user clicks the button.

Invoke Service Script Steps

Invoke service script steps allow you to execute a service script.

The following additional fields are required for **Invoke service script** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

Group Name references the data area to be passed to and from the server when the **Service Script** is invoked.

The service script call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a service script is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

Groovy Imports Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

Groovy Imports step types provide a free format text area where you list the Groovy classes to be imported for use by the code in **Groovy Member** steps within the script.

Enter a description of the imports step in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter the list of classes to import in the **Edit Data Text** field using the syntax **import 'class';** where 'class' is the fully qualified package name of the Groovy class. Click the adjacent icon to open a window providing more space for editing the text.

Copyright © 2026, Oracle and/or its affiliates.

When saving a script, if the system has been configured accordingly, when the system detects references to a Groovy class that is not explicitly listed in this step, the system warns the user and adds an explicit import for that class. (If no Groovy Import step type exists for the script, one will be added.) Users should review the imported class to ensure it is the expected class, in the case of multiple classes with a similar name.

> **Note**
>
> For security, the classes that may be imported are restricted to those allowed by the Framework. The warning and automatic update to the script is controlled by a property setting. Refer to the topic Using Groovy Within Scripts for more information.

Groovy Library Interface Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

Groovy Library Interface steps are only applicable to **Groovy Library Scripts**. They provide a free format text area where you can list the Groovy methods defined within the script that are available for use by other scripts.

Enter a description of the interface step in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter the list of interface methods in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for editing the list.

> **Note**
>
> Every **Groovy Library Script** must have one and only one step of type **Groovy Library Interface**. The supporting code for the available methods is defined using one or more **Groovy Member** steps in the same script. Refer to the topic Using Groovy Within Scripts for more information.

Groovy Member Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

Groovy Member steps provide a free format text area where you can enter Groovy code.

Enter a description of the code block in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter your code in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for editing the code.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> While it is possible to set up multiple steps of type **Groovy Member** the system treats these steps as a single class for compilation and execution purposes. Refer to the topic Using Groovy Within Scripts for more information.

Useful APIs

The following provides information about APIs that may be useful for Groovy programmers.

Lookup object instantiation

This API allows programmers to instantiate a lookup for subsequent use in Groovy code when there is no Java class generated for the lookup (as is often the case with custom lookups). Example:

```
Lookup customLookup = LookupHelper.getLookupInstance("ALG_ENTITY_FLG",
"CMAL");
```

Label Steps

Label steps allow you to describe what the next step(s) are doing. Steps of this type are helpful to the script administrators when reviewing or modifying the steps in a script, especially when a script has many steps. When designing a script, the label steps enable you to provide a heading for common steps that belong together. The script tree displays steps of this type in a different color (green) so that they stand out from other steps.

There are no additional fields for **Label** steps.

Mathematical Operation Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Mathematical operation steps allow you to perform arithmetic on fields. You can also use this type of step to add and subtract days from dates. For example, you could calculate a date 7 days in the future and then use this value as the customer's next credit review date. The following additional fields are required for **Mathematical Operation** steps:

Base Field Type and **Base Field Name** define the field on which the mathematical operation will be performed. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model**. Use this field type when the field resides on any of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage**. Use this field type when the field resides in temporary storage. You must initialize the temporary storage field with a Move Data step before performing mathematical operations on the field. Refer to How To Set Up Move Data Steps for more information.

Copyright © 2026, Oracle and/or its affiliates.

- **User Interface Field**. Use this field type when the field resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

Math Operation controls the math function to be applied to the **Base Field**. You can specify +, -, /, and *. Note, if the base field is a date, you can only use + or -.

Math Field Type, **Math Field Name** and **Math Field Value** define the field that contains the value to be added, subtracted, divided, or multiplied. The following points describe each field type:

- **Current To Do Information**. Use this field type when the value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the value resides on any of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the value is a constant. When this field type is used, use **Source Field Value** to define the constant value. Refer to How To Use Constants In Scripts for more information. Note, if you are performing arithmetic on a date, the field value must contain the number and type of **days**/ **months**/ **years**. For example, if you want to add 2 years to a date, the source field value would be **2 years**.
- **Temporary Storage**. Use this field type when the value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the value resides in a field on the current tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

Move Data Steps

Move data steps allow you to move data (from a source to a destination). The following additional fields are required for **Move data** steps:

Source Field Type, **Source Field Name** and **Source Field Value** define what you're moving. The following points describe each field type:

- **Context Variable**. Use this field type in a plug-in or service script if the source value is a variable initiated in a higher level script. This is only applicable to Service Scripts and Plug-in Scripts.
- **Current To Do Information**. Use this field type when the source value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**. This is only applicable to BPA Scripts.
- **Data Area**. Use this field type when the field being compared is one that you put into one of the script's data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Global Context**. Use this field type when the source value is a global context variable. This is only applicable to BPA Scripts.
- **Page Data Model**. Use this field type when the source value resides on any of the tab pages in the object display area (i.e., the source field doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed).

Copyright © 2026, Oracle and/or its affiliates.

Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

- **Portal Context**. Use this field type when the source value is a variable in the portal context. This is only applicable to BPA Scripts.
- **Predefined Value**. Use this field type when the source value is a constant value defined in the script. When this field type is used, use **Source Field Value** to define the constant value. Refer to How To Use Constants In Scripts for instructions on how to use constants.

> **Note**
>
> **Concatenating fields together.** You can also use **Predefined Value** if you want to concatenate two fields together. For example, let's say you have a script that merges two persons into a single person. You might want this script to change the name of the person being merged out of existence to include the ID of the person remaining. In this example, you could enter a **Source Field Value** of `%ONAME` **merged into person %PERID** (where **ONAME** is a field in temporary storage that contains the name of the person being merged out of existence and **PERID** contains the ID of the person being kept). Refer to How To Substitute Variables In Text for a description of how you can substitute field values to compose the field value.

- **Temporary Storage**. Use this field type when the source value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the source value resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

Destination Field Type and **Destination Field Name** define where the source field will be moved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Context Variable**. Use this field type in your plug-in or service script if you use a variable to communicate information to a lower level service script or schema. This is not applicable to BPA Scripts.
- **Data Area**. Use this field type when the destination field resides on one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the destination field resides on any of the tab pages in the object display area (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.
- **Portal Context**. Use this field type when the destination to be updated is in the current portal context. This is only applicable to BPA Scripts.
- **Temporary Storage**. Use this field type when the destination field resides in temporary storage. Use **Field Name** to name the field in temporary storage. Use **Field Name** to name the field in temporary storage. Refer to How To Name Temporary Storage Fields for more information.

Copyright © 2026, Oracle and/or its affiliates.

- **User Interface Field**. Use this field type when the destination field resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

> **Note**
>
> **Conditional field types.** The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts.

Navigate To A Page Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Navigate to a page steps cause a new page (or tab within the existing page) to be displayed in the object display area. Steps of this type are a precursor to doing anything on the page. The following additional field is required for **Navigate to a page** steps:

Navigation Option defines the transaction, tab, access mode (add or change) and any context fields that are passed to the transaction in change mode. For example, if you want a script to navigate to Person - Characteristics for the current person being displayed in the sidebar, you must set up an appropriate navigation option. Refer to Defining Navigation Options for more information.

> **Note**
>
> **Navigating to a page in update mode.** Before you can navigate to a page in change mode, the page data model must contain the values to use for the navigation option's context fields. If necessary, you can move values into the page data model using a Move Data step first. For example, before you can navigate to a page in change mode with an account ID in context, you may need to move the desired account ID into the ACCT_ID field in the page data model. The actual field name(s) to use are listed as context fields on the navigation option.

Perform Script Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Perform script steps cause another BPA script to be performed. After the performed script completes, control is returned to the next step in the original script. You might want to think of the scripts referred to on steps of this type as "subroutines". This functionality allows you to encapsulate common logic in reusable BPA scripts that can be called from other BPA scripts. This simplifies maintenance over the long term.

The following additional field is required for **Perform script** steps:

Copyright © 2026, Oracle and/or its affiliates.

Subscript is the name of the script that is performed.

Press A Button Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Press a button steps cause a button or link text to be 'pressed' in the object display area, the application toolbar or the page title area. For example, you could use this type of step to add a new row to a person's characteristic (and then you could use a **Move Data** step to populate the newly added row with a given char type and value). The following additional fields are required for **Press a button** steps:

Button Name is the name of the button to be pressed. This button must reside on the currently displayed tab page (or in the application toolbar or page actions toolbar). Refer to How To Find The Name Of A Button for more information.

Prompt User Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the user using the script area are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Prompt user steps cause the user to be presented with a menu of options. The options can be presented using either buttons or in the contents of a drop down. You can also use steps of this type to pause a script while the user checks something out (and when the user is ready to continue with the script, they are instructed to click a prompt button). The following additional fields are required for **Prompt User** steps:

Prompt Type controls if the prompt shown in the script area is in the form of `Button(s)` or a **Dropdown**. Note, if you use a **Dropdown**, a Continue button appears adjacent to the dropdown in the script area when the step executes. The user clicks the Continue button when they are ready for the script to continue.

The **Prompt Values** grid contains a row for every value that can be selected by a user. Note, if you use a **Prompt Type** of `Button(s)`, a separate button is displayed in the script area for each entry in this grid.

Copyright © 2026, Oracle and/or its affiliates.

- **Prompt Text** is the verbiage to appear on the button or in the dropdown entry. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons or dropdown entries.
- **Use As Default** can only be turned on for one entry in the grid. If this is turned on for a dropdown entry, this value is defaulted in the grid. If this is turned on for a button, this button becomes the default (and the user should just have to press `Enter` (or `space`) rather than click on it).
- **Next Script Step** defines the step to execute if the user clicks the button or selects the dropdown value.

Set Focus To A Field Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the underlying page's fields and buttons are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Set focus to a field steps cause the cursor to be placed in a specific field on a page. A **Continue** button always appears in the script area when this type of step executes. The user may click the **Continue** button when they are ready for the script to continue. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Set focus to a field** steps:

Destination Field Name defines the field on which focus should be placed. This field must reside on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

Terminate Steps

Terminate steps cause a server-based script to end processing successfully or issue an error.

The following additional fields are required for **Terminate** steps:

Copyright © 2026, Oracle and/or its affiliates.

Error indicates whether an error should be thrown or not. If error, **Error Data Text** must be specified, indicating the error message and any message substitution parameters. Refer to Edit Data Syntax the actual syntax of initiating an error message.

> **Note**
>
> The ability to terminate a step in error is only supported for server-based scripts.

Transfer Control Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

Transfer control steps cause the current BPA script to terminate and the control to pass to another BPA script. You might want to construct a BPA script with steps of this type when the script has several potential logic paths and you want to segregate each logic path into a separate BPA script (for ease of maintenance).

The following additional fields are required for **Transfer control** steps:

Subscript is the name of the script to which control is transferred.

Additional Topics

The contents of this section provide additional information about steps.

- How To Find The Name Of User Interface Fields
- How To Find The Name Of Page Data Model Fields
- How To Find The Name Of A Button
- How To Substitute Variables In Text
- How To Use HTML Tags And Spans In Text Strings and Prompts
- How To Use Constants In Scripts
- How To Use Global Variables
- How To Name Temporary Storage Fields
- How To Work With Dates
- How To Use To Do Fields
- How To Reference Fields In Data Areas
- Script Engine Version 2 and Above Notes

How To Find The Name Of User Interface Fields

This section is only relevant if you are using the legacy capability of BPA scripts to bring a user to a base fixed page and populate information into fields on that page.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Guiding scripts that interact with and surf over fixed pages are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

Follow these steps to find the name of a field that resides on a fixed page:

- Navigate to the page in question.
- Right click in the body of the page.
- Select **Inspect** from the pop-up menu to display the HTML.
- In the search look for the text "Widget Info". Note that if there are multiple sections on the page, you will find multiple "Widget Info" results. Based on the details shown, you are able to confirm which section you are looking for. For example, the following is an example from the Algorithm - Main page:

```
* Widget Info:
*     Widget_ID , Element Type - label info - label
*     ALG_CD, IT - CI_ALG$ALG_CD -  Algorithm Code
*     IM_ALG_CD, IM - $SEARCH_FOR_ALG_LBL -  Search for Algorithm
*     OWNER_FLG, IS - $OWNER_FLG -  Owner
*     DESCR50, IT - $DESCR50 -  Description
*     IM_Main_algtypCont, IM - $F1_SHOW_ALG_CTX -  Show Algorithm Context Menu
*     ALG_TYPE_CD, IT - $ALG_TYPE_CD -  Algorithm Type
*     IM_ALG_TYPE_CD, IM - $SEARCH_ALGTYPE_LBL -  Search for Algorithm Type
*     ALG_TYPE_DESCR50, PL - $DESCR50 -  Description
*     ALG_TYPE_LONG_DESC, IA - $DESCRLONG -  Detailed Description
*     IM_VerScroll_lftArrow, IM - $PREVIOUS_ALG_LBL -  Previous Version
*     ALG_VER$recordCount, SN - $OF_LBL -  Of
*     IM_VerScroll_rtArrow, IM - $NEXT_ALG_LBL -  Next Version
*     IM_ADD, IM - $ADD_ALG_LBL -  Add Version
*     IM_DEL, IM - $DELETE_ALG_LBL -  Delete  Version
*     ALG_VER$EFFDT, IT - $EFFDT -  Effective Date
*     L_OWNER_LBL, LP - $OWNER_FLG -  Owner
*     ALG_VER$OWNER_FLG, IS - $OWNER_FLG -  Owner
*     resultGrid, GD
```

The field names that you'll reference in your scripts are defined on the left side of the HTML (e.g., ALG_CD, DESCR50, ALG_TYPE_CD, etc.).

The names of fields that reside in scrolls are in a slightly different format. The following is an example of the HTML for the version scroll that appears on Algorithm - Main. Notice that the fields in the scroll are prefixed with the name of the scroll plus a $ sign. For example, the effective date is called ALG_VER$EFFDT.

```
*     IM_VerScroll_lftArrow, IM - $PREVIOUS_ALG_LBL -  Previous Version
*     ALG_VER$recordCount, SN - $OF_LBL -  Of
*     IM_VerScroll_rtArrow, IM - $NEXT_ALG_LBL -  Next Version
*     IM_ADD, IM - $ADD_ALG_LBL -  Add Version
*     IM_DEL, IM - $DELETE_ALG_LBL -  Delete  Version
*     ALG_VER$EFFDT, IT - $EFFDT -  Effective Date
*     L_OWNER_LBL, LP - $OWNER_FLG -  Owner
*     ALG_VER$OWNER_FLG, IS - $OWNER_FLG -  Owner
```

The names of fields that reside in grids are in a slightly different format. The following is an example of the HTML for the algorithm parameters grid that appears on Algorithm - Main. Notice that the fields in the grid are prefixed with the name of the grid plus a `:x$`. For example,

Copyright © 2026, Oracle and/or its affiliates.

the algorithm parameter label is called `ALG_PARM:x$ALG_PARM_LBL`. When you reference such a field in your script, you have the following choices:

- Substitute `x` with the row in the grid (and keep in mind, the first row in a grid is row `0` (zero); this means the second row is row `1`).
- If you want to reference the "current row" (that is, the row in which the cursor will be placed), you can keep the `x` notation (`x` means the "current row").

```
* Widget Info:
*     Widget_ID , Element Type - label info - label
*     ALG_PARM:x$ALG_PARM_LBL, IL - CI_ALG_TYPE_PRM_L$ALG_PARM_LBL -  Parameter
*     ALG_PARM:x$SEQNO, IL - $SEQNO -  Sequence
*     ALG_PARM:x$ALG_PARM_VAL, IA - $ALG_PARM_VAL -  Value
*
```

How To Find The Name Of Page Data Model Fields

You find the name of a **Page Data Model** field in the same way described under How To Find The Name Of User Interface Fields. The only restriction is that you cannot refer to hidden / derived fields. However, you can refer to any of the object's fields regardless of the tab page on which they appear. For example, if you position the object display area to the Main tab of the Account transaction, you can reference fields that reside on all of the tab pages.

> ⚠ **Caution**
>
> If you populate a **Page Data Model** field, none of the underlying default logic takes place. For example, if you populate a customer contact's contact type, none of the characteristics associated with the customer contact type are defaulted onto the customer contact. If you want the underlying defaulting to take place, you must populate a **User Interface Field**.

How To Find The Name Of A Button

This section is only relevant if you are using the legacy capability of BPA scripts to bring a user to a base fixed page and perform functionality by executing existing buttons on that page.

> ⚠ **Caution**
>
> Guiding scripts that interact with and surf over fixed pages are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

If you want a **Press a button** step to press a button or click a link in the application toolbar, use one of the following names:

Button Name
`IM_GOBACK`
`IM_HISTORY`
`IM_GOFORWARD`
`IM_menuButton`

Copyright © 2026, Oracle and/or its affiliates.

IM_USER_HOME
IM_MY_PREF
IM_helpButton

If you want a **Press a button** step to press a button in the page actions toolbar, use one of the following names:

Button Name
IM_SAVE IM_REFRESH IM_CLEAR IM_COPY
IM_DELETE
IM_ScrollBack
IM_ScrollForward

The following buttons are also supported:

Button Name	Comments
IM_TO_DO	
IM_CurrentTo Do	This navigates to the To Do Entry page for the user's current To Do. Refer to Working On A To Do for more information.
IM_MINIMIZE_DASHBOARD	Pressing this will collapse the sidebar. Note that when a script is finished, it will return the sidebar to the state it was when the script was launched.
IM_MAXIMIZE_DASHBOARD	Pressing this will expand the sidebar. Note that when a script is finished, it will return the sidebar to the state it was when the script was launched.

Follow these steps to find the name of other buttons that reside in the object display area of a fixed page:

- Navigate to the page in question.
- Right click in the body of the page
- Select **Inspect** from the pop-up menu.
- In the search look for the text "Widget Info". Note that if there are multiple sections on the page, you will find multiple "Widget Info" results. Based on the details shown, you are able to confirm which section you are looking for.
- Iconized buttons (e.g., search buttons) are represented as HTML images and their field names are prefixed with **IM**. The following is an example of the HTML on the Algorithm - Main page (notice the **IM** fields for the iconized buttons).

```
 * Widget Info:
 *     Widget_ID , Element Type - label info - label
```


Copyright © 2026, Oracle and/or its affiliates.

```
    *     ALG_CD, IT - CI_ALG$ALG_CD -  Algorithm Code
    *     IM_ALG_CD, IM - $SEARCH_FOR_ALG_LBL -  Search for Algorithm
    *     OWNER_FLG, IS - $OWNER_FLG -  Owner
    *     DESCR50, IT - $DESCR50 -  Description
    *     IM_Main_algtypCont, IM - $F1_SHOW_ALG_CTX -  Show Algorithm Context
Menu
    *     ALG_TYPE_CD, IT - $ALG_TYPE_CD -  Algorithm Type
    *     IM_ALG_TYPE_CD, IM - $SEARCH_ALGTYPE_LBL -  Search for Algorithm
Type
```

- Transaction-specific actions buttons are represented as switches. The following is an example of the HTML on the Batch Job Submission - Main page (notice the **SW** fields for the buttons). Note, if you want to **Set focus** to such a field, you would move a **Predefined Value** of **TRUE** to the switch.

```
    *     ACTION_Q_SW, BU - $ACTION_Q_SW -  Duplicate & Queue
    *     ACTION_CANCEL_SW, BU - $CANCEL_SW -  Cancel
```

How To Substitute Variables In Text

You can substitute field values into a step's text string. You do this by prefixing the field name whose value should be substituted in the string with a %. For example, the message, "On %COMPLETION_DTTM this bill was completed, it's ending balance was %ENDING_BALANCE" contains two substitution variables (the bill's completion date / time and the bill's ending balance).

To substitute the value of an element from a data area you need to reference its XPath location as follows: %=XPath=%. If you want to substitute the whole XML node, not just the value, you need to reference it as follows %+XPath+%.

Only fields linked to the current To Do and fields that reside in temporary storage and global variables can be substituted into a text string.

> **Note**
>
> You can substitute fields that reside in the User Interface or Page Data Model by first moving them into temporary storage (using a **Move data** step).

You can also substitute field values into the verbiage displayed in prompts using the same technique.

How To Use HTML Tags And Spans In Text Strings and Prompts

You can use HTML tags in a step's text string. For example, the word "Continue" will be italicized in the following text string "PressContinue after you've selected the customer" (the and are the HTML tags used to indicate that the surrounded text should be italicized).

The following are other useful HTML tags:

-
 causes a line break in a text string. If you use

 a blank line will appear.
- **** text causes the surrounded text to be colored as specified (in this case, red). You can also use hex codes rather than the color name.

Copyright © 2026, Oracle and/or its affiliates.

Please refer to an HTML reference manual or website for more examples.

> **Note**
>
> Refer to Color Contrast for information about the use of the HTML color 'red' and its impact on accessibility.

How To Use Constants In Scripts

Some steps can reference fields called **Predefined Values**. For example, if you want to compare an input value to the letter "Y", the letter Y would be defined as a Predefined Value's field value.

Special constants are used for fields defined as switches. When you move **TRUE** to a switch, it turns it on. When you move **FALSE** to a switch, it turns it off.

You can use a global variable as a Predefined Value. For example, if you wanted to move the current date to a field, you'd indicate you wanted to move a Predefined Value named %CURRENT_DATE.

How To Use Global Variables

Some explicit steps can reference fields called **Predefined Values**. In addition to referencing an ad hoc constant value (e.g., the letter Y), you can also reference a global variable in such a field value. A global variable is used when you want to reference system data.

Note that when using the Edit Data step type, the variable available are slightly different. Refer to Edit Data Syntax for details.

The following global variables exist for BPA scripts:

Variable Name	Comments
%PARM-<name>	This is the value of a parameter of that name passed in to the application when launched via the standard system URL. Refer to Launching A Script When Starting the System for more information on these parameters.
%PARM-NOT-SET	This is to be used to compare against **%PARM-**< > parameters to check if the parameter has been set or not when the application was launched. A parameter that has not been set would be considered equal to this global variable. It is recommended to compare parameters against this global variable before using them for the first time.
%BLANK	A constant that contains a blank value (no value).
%SPACE	A constant that contains a single space value.
%CURRENT-DATE	The current date as known by the browser, not the server.
%SYSTEM-DATE	The server date. Note that this date is affected by the system date override logic)
%SAVE-REQUIRED	A flag that contains an indication of whether the data on a page has been changed (and thus requires saving). You may want to interrogate this flag to force a user to save their work before executing subsequent steps. This flag will have a value of **TRUE** or **FALSE**.

Copyright © 2026, Oracle and/or its affiliates.

Variable Name	Comments
%NEWLINE	A constant that contains a new line character (carriage return). Upon substitution, a line break is inserted in the resultant text.

> **Note**
>
> This constant does not have the desired effect when the resultant text is HTML. For example, a step's text and prompt strings. This is because HTML ignores special characters such as new lines. Refer to How To Use HTML Tags And Spans In Text to learn how to cause a line break in an HTML text.

To refer to a global context variable, use %FIELD_NAME. For example, if the field SP_ID is in the global context, you may reference %SP_ID to reference the ID of the service point currently in context. In addition, the following special values are supported:

Variable Name	Comments
%CONTEXT-PERSONID	A constant that contains the ID of the current person.
%CONTEXT-ACCOUNTID	A constant that contains the ID of the current account.
%CONTEXT-PREMISEID	A constant that contains the ID of the current premise.

In addition, if the script is invoking something else via one of the various “Invoke” step types and an error is returned, the following global variables contain information about the error:

Variable Name	Comments
%ERRMSG-CATEGORY %ERRMSG-NUMBER	The unique identifier of the error message number.
%ERRMSG-TEXT	The brief description of the error.
%ERRMSG-LONG	The complete description of the error.

How To Name Temporary Storage Fields

Input Data and **Move Data** steps can create fields in temporary storage. You specify the name of the temporary storage field in the step's **Field Name**. The name of the field must not begin with % and must not be named the same as the global variables. Besides this restriction, you can use any **Field Name** that's acceptable to JavaScript (i.e., you can name a field in temporary storage almost anything). Keep in mind that field names are case-sensitive.

How To Work With Dates

Before we discuss how to work with dates in your scripts, we need to point out that there are two types of date fields: date-only and date-time. Date-only fields only contain a date. Date-time fields contain both a date and a time. The following topics describe how to work with dates on the various step types.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If you're working with a field that resides on the database (as opposed to a temporary storage field), the database field name will tell you what type of date it is: date-only fields are suffixed with **DT**, and date-time fields are suffixed with **DTTM**.

- Move Data
- Mathematical Operation
- Input Data

Move Data

If you intend to use a **Move data** step to populate a *date-time* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD-HH.MM.SS or YYYY-MM-DD. If the field is in the format YYYY-MM-DD, the time of 12:00 am will be defaulted.
- If the destination field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called EXPIRE_DTTM:
  - First, you populate the date portion of the field. To do this, you'd move a date (this value can be in any valid date format that a user is allowed to enter) to a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix `_FWDDTM_P1` to the field name.
  - If you want to populate the time, you'd move the time (again, the field value can be in any format that a user could use to enter a time) to a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix `_FWDDTM_P2` to the field name.

If you intend to use a **Move data** step to populate a *date-only* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD.
- If the destination field resides in the *user interface*, the source field can be in any valid date format that a user is allowed to enter.

> **Note**
>
> **%CURRENT-DATE.** Keep in mind that the global variable%CURRENT-DATE contains the current date and you can move this to either a page data model, user interface, or temporary storage field. If you move %CURRENT-DATE to a temporary storage fields, it is held in the format YYYY-MM-DD.

Mathematical Operation

If you intend to use a **Mathematical operation** step to calculate a date, you can reference both date-only and date-time fields. This is because mathematical operations are only performed against the date portion of date-time fields.

Copyright © 2026, Oracle and/or its affiliates.

Mathematical operations are limited to adding or subtracting days, months and years to / from a date.

> **Note**
>
> A useful technique to perform date arithmetic using the current date is to move the global variable%CURRENT-DATE to a temporary storage field and then perform the math on this field.

Input Data

If you intend to use an **Input data** step on a *date-time* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD-HH.MM.SS (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called EXPIRE_DTTM:
  - First, you populate the date portion of the field. To do this, you'd input the date (this value can be in any valid date format that a user is allowed to enter) in a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix `_FWDDTM_P1` to the field name.
  - If you want to populate the time, you'd input the time (again, the field value can be in any format that a user could use to enter a time) in a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix `_FWDDTM_P2` to the field name.

If you intend to use an **Input data** step to populate a *date-only* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, the user can enter any valid date format.

How To Use To Do Fields

As described under Executing A Script When A To Do Entry Is Selected, you can set up the system to automatically launch a script when a user selects a To Do entry. These types of scripts invariably need to access data that resides on the selected To Do entry. The following points describe the type of information that resides on To Do entries:

- **Sort keys**. These values define the various ways a To Do list's entries may be sorted. For example, when you look at the bill segment error To Do List, you have the option of sorting the entries in error number order, account name order, or in customer class order. There is a sort key value for each of these options.
- **Message parameters**. These values are used when the system finds *%n* notation within the message text. The *%n* notation causes field values to be substituted into a message before it's displayed. For example, the message text **The %1 non-cash deposit for %2 expires on %3** will have the values of three fields merged into it before it is displayed to the user (%1 is the type of non-cash deposit, %2 is the name of the customer, and %3 is the expiration date of the non-cash deposit). Each of these three values is stored as a separate message parameter on the To Do entry.

Copyright © 2026, Oracle and/or its affiliates.

- **Drill keys**. These values are the keys passed to the page if a user drilled down on the entry (and the system wasn't set up to launch a script). For example, a To Do entry that has been set up to display an account on the account maintenance page has a drill key of the respective account ID.
- **To Do ID**. Every To Do entry has a unique identifier referred to as its To Do ID.

You can access this information in the following types of steps:

- **Move Data** steps can move any of the above to any data area. For example, you might want to move a To Do entry's drill key to the page data model so it can be used to navigate to a specific page.
- **Conditional Branch** steps can perform conditional logic based on any of the above. For example, you can perform conditional logic based on a To Do entry's message number (note, message numbers are frequently held in sort keys).
- **Mathematical Operation** steps can use the above in mathematical operations.

A To Do entry's sort key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of `SORTKEY[index]`. Note, you can find an entry's potential sort keys by displaying the entry's To Do type and navigating to the Sort Keys tab. If you want to reference the first sort key, use an index value of `1`. If you want to use the second sort key, use an index value of 2 (and so on).

A To Do entry's drill key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of `DRILLKEY[index]`. Note, you can find an entry's potential drill keys by displaying the entry's To Do type and navigating to the Drill Keys tab. If you want to use the first drill key, use an index value of `1`. If you want to use the second drill key, use an index value of 2 (and so on).

A To Do entry's message parameters are accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of `MSGPARM[index]`. Note, because a To Do type can have an unlimited number of messages and each message can have different parameters, finding an entry's message parameters requires some digging. The easiest way to determine these values is to display the To Do entry on To Do maintenance. On this page, you will find the entry's message category/number adjacent to the description. Once you know these values, display the message category/number on Message Maintenance. You'll find the message typically contains one or more %n notations (one for each message parameter). For example, the message text **The %1 non-cash deposit for %2 expires on %3** has three message parameters. You then need to deduce what each of the message parameters are. You do this by comparing the message on the To Do entry with the base message (it should be fairly intuitive as to what each message parameter is). If we continue using our example, `%1` is the non-cash deposit type, `%2` is the account name, and `%3` is the expiration date. You can access these in your scripts by using appropriate index value in `MSGPARM[index]`.

A To Do entry's unique ID is accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of `TD_ENTRY_ID`.

In addition, any of the above fields can be substituted into a text string or prompt. Simply prefix the To Do field name with a % as you would fields in temporary storage. For example, assume you want your script to display the following text in the script area: "ABC Supply does not have a bill cycle" (where ABC Supply is the account's name). If the first sort key linked to the To Do entry contains the account's name, you'd enter a text string of **%SORTKEY[1] does not have a bill cycle**.

How To Reference Fields In Data Areas

Various step types involve referencing field elements residing in the script's data areas. To reference an element in a data area you need to provide its absolute XPath notation starting

Copyright © 2026, Oracle and/or its affiliates.

from the data area name. For example, use "F1-BatchControl/batchControl" to reference a top-level "batchControl" element in a script data area called "F1–BatchControl".

You don't have to type in long XPath notations. Use the **View Script Schema** hyperlink provided on the Script portal to launch the script's data areas schema.

Figure Schema Viewer



Doing this opens the schema viewer window where you can:

- Click on the field element you want to reference in your script step. The system automatically populates the text box on the top with the element's absolute XPath notation.
- Copy the element's XPath notation from the text box to your script.

You can also use the **View Data Area**, **View Service Script Data Area**, or **View Plug-In Script Data Area** links on the Data Areas tab page on the Script portal to the same effect. These open up the schema viewer for a specific data area respectively.

Script Engine Version 2 and Above Notes

Scripting using the engine version 2 or above requires some extra syntax to take advantage of XPath 2 functionality. In general, any variable declared will be assumed to be a string. This means, that if you intend to construct a mathematical statement then it is necessary to explicitly declare the data type of variables as integers, numbers, or dates.

> **Note**
>
> Unless otherwise noted, all XPath examples in this topic are for the Version 1 engine – which means XPath 1. Statements that function using XPath 1 will not necessarily work for XPath 2. This is especially true when executing math, see below for examples.

Date and Time Arithmetic

XPath date/time and interval data types support arithmetic operations ('+', '-', '*' etc.) and functions, which can be used for time calculations in the same way as '1 + xs:integer(value)' is used for numeric calculations.

Compare time duration:

```
if ("(xs:dateTime(fn:current-dateTime()) - xs:dateTime($updateDateTimeX))
   ge xs:dayTimeDuration(concat('PT', BO/hoursBetweenStatisticsUpdate, 'H'))")
   goto 60;
end-if;
```


Copyright © 2026, Oracle and/or its affiliates.

Compare one date to another:

```
if ("xs:date(parm/endDate) < xs:date(parm/startDate)")
   terminate with error (11108, 11507 element='endDate');
end-if;
```

Compare a date against today's date:

```
if ("xs:date(parm/startDate) <= xs:date($CURRENT-DATE)")
   terminate with error (11108, 11507 element='endDate');
end-if;
```

Calculate the end of month:

```
// covert to ISO
move "concat($year,'-',$mon2,'-01T00:00:00')" to $monthStart;

// calculate
move "xs:dateTime($monthStart) + xs:yearMonthDuration('P1M') -
xs:dayTimeDuration('P0DT1S')"
  to $monthEnd;

// convert from ISO to OUAF
move
"concat($year,'-',$mon2,'-',substring(string($monthEnd),9,2),'-23.59.59')"
to $endDateTime;
```

> **Note**
>
> XPath date/time/interval formats use the ISO standard, which needs to be converted to/from formats supported in the framework.

Comparing Date/Times in String Format

Any ISO-like string format for date/time preserves the YYYY MM DD HH MM SS sequence, which is zero-padded. Regardless of separators, this format will remain appropriate for comparison operations. In particular, date/time values in the framework format “YYYY-MM-DD.HH.MM.SS” can be used with “=”, “!=”, as well as “>”, “>=”, “<”, “<=” operators.

```
// retrieve framework date/time value
invokeBS 'CM-MAXMSRMT' using "CM-MAXMSRMT";
move "string(cm-MAXMSRMT/results[1]/measurementDateTime)" to $lastMsmtDT;

// construct another date/time
move "concat($year,'-01-01-00.00.00')" to $startDateTime;

// compare using string operators
if ("$lastMsmtDT >= $startDateTime")
    move "substring($lastMsmtDT,1,4)" to $latestMsrmtYear;
```


Copyright © 2026, Oracle and/or its affiliates.

Converting Date/Times Between Framework and ISO

Conversion of date/time from framework format to ISO is only necessary for date/time arithmetic. Comparisons can be done with the framework format directly. The only difference between the framework format and ISO date/time formats is in the separators:

Framework: “YYYY-MM-DD.HH.MM.SS”

ISO: “YYYY-MM-DDTHH:MM:SS”

Example of converting from the framework format to ISO:

```
move "concat(substring($ouafDT, 1, 10), 'T', translate(substring($ouafDT,
12),'.',':'))" to $isoDT;
```

Example of converting from ISO to the framework format:

```
move "concat(substring($isoDT, 1, 10), '.', translate(substring($isoDT,
12),':','.'))" to $ouafDT;
```

Round Money With a Dynamic Currency Scale

Because different currencies support a different number of decimals, the framework provides an API for rounding a monetary amount based on a given currency.

```
move "parm/amount" to $qnty;
move "currency/decimals" to $decimals;
move "fn:round(xs:decimal($qnty) * math:exp10(xs:double($decimals)))
  div math:exp10(xs:double($decimals))" to "parm/roundedAmount";
```

Looping through Sequences

In XPath 2 it is possible to organize a for-loop over a sequence of integers, not only a node list.

This example shows a loop over a range of months. This is a sequence-forming construct in XPath. The XPath node list, which we are familiar with, is just another type of sequence.

```
for ($month in "1 to 12")
```

This example shows a loop over a give range of years in descending order:

```
for ($year in "fn:reverse(parm/startYear to parm/endYear)")
     move "concat($year,'-01-01-00.00.00')" to $startDateTime;
     move "concat($year,'-12-31-23.59.59')" to $endDateTime;
     ...
```

This example shows a loop through a node list using ‘index’, so that other node lists can be accessed:

```
for ($idx in "1 to count(parm/touData/touList)")
     move "parm/touData/touList[$idx]" to $tou;   // access any list with
this index
```


Copyright © 2026, Oracle and/or its affiliates.

The above syntax can be used as an elegant alternative to maintaining indices separately, for example instead of the following:

```
move “0” to $idx;
for ($item in "parm/touData/touList")
     move “1 + xs:integer($idx)” to $idx;
```

String Padding and Decimal Formatting

This is used with specific input formats or output formatting. It is applicable to zero, space and other types of padding.

This example shows prefixing for date/time components, for example producing “2010-01-02” instead of “2010-1-2”.

```
move "substring(concat('0',string($month)), string-length(string($month)),
2)" to $mon2;
```

This example shows suffixing for adding decimal zero-padded alignment, for example producing “12.30” and “4.00” instead of “12.3” and “4”. The example performs 3 tasks: rounding to 2 decimals, inserting a period if necessary, and zero padding.

```
// round and zero-pad to 2 decimals
move "$item/amount" to $qty;
move "fn:round(xs:double($qty) * 100) div 100" to $qty;
move "string($qty)" to $qty;
move "concat(substring-
before(concat($qty,'.'),'.'),'.',substring(concat(substring-
after($qty,'.'),'00'),1,2))" to $qty;
```

Ternary Operation

This makes a choice between values based on a condition, so that it could be used in a single expression instead of an if/else block. It is know in C/C++ as ‘cond ? value1 : value2’ or in BASIC as ‘IFF(cond, value1, value2)’. In XPath the syntax is: "if (cond) then value1 else value2". Note this is not the top-level scripting if-statement block.

In XPath this is an expression, which can be combined with other expressions. In scripting it can be used as:

```
move "if (string(D1-UnitOfMeasure/measuresPeakQuantity) = 'D1MP') then 'D1MX'
else 'D1SM' " to $func;
```

Pipeline Processing

In scripting, it is not easy to create a simple reusable piece of code as there are no local functions, and a separate script call is a coding overhead and requires packing/unpacking parameters. To avoid copying and pasting the same code block between similar script stages, consider ‘pipelining’, which is breaking the overall process into separate top-level steps, some of which could be shared between alternating paths. This is common for parameter preparation and output formatting. An intermediate result between stages can be stored in a “parm” substructure.

Copyright © 2026, Oracle and/or its affiliates.

Instead of this code:	Consider this alternative:

`if ("type = A")`
`  prepare params ...`
`  call services for A ...`
`  format output ...`
`end-if;`
`if ("type = B")`
`  prepare params ...`
`  call services for B ...`
`  format output ...`
`end-if;`
	`prepare params ...`
	
	`if ("type = A")`
	`  call services for A ...`
	`end-if;`
	`if ("type = B")`
	`  call services for B ...`
	`end-if;`
	
	`format output ...`

XPath 2 Functions

Script engine versions 2 and above support XQuery 1.0 Functions and Operators, and the XQuery 1.0 standard itself with some minor limitations. Below are the URLs to both specifications. The first link has the functions/operators available to use from XQuery.

- http://www.w3.org/TR/xpath-functions/
- http://www.w3.org/TR/xquery/

The following can only access local file systems. (For other protocols like http they will return an empty sequence):

- fn:doc
- fn:collection

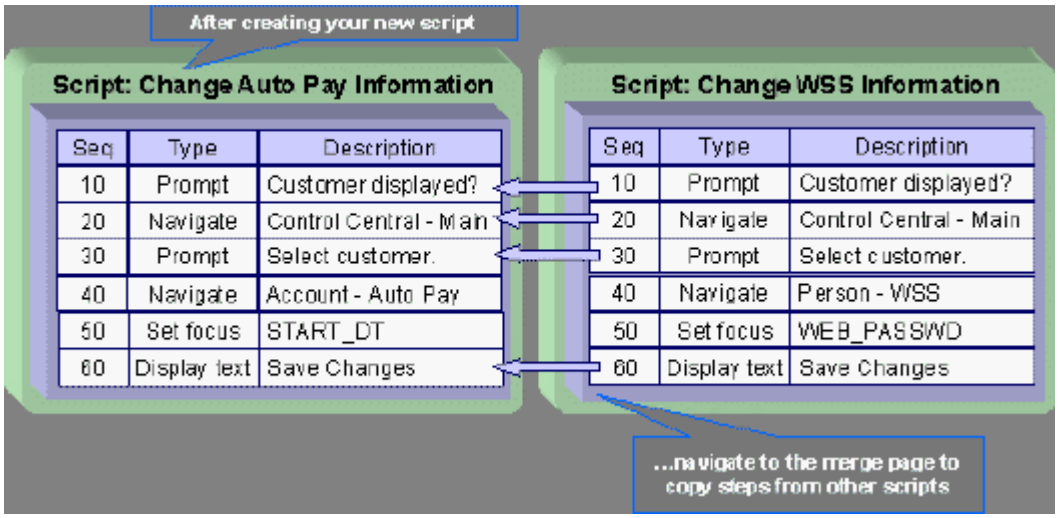
Merging Scripts

Use the Script Merge page to modify an existing script by copying steps from other scripts. The following points summarize the many diverse functions available on the Script Merge transaction:

- You can use this transaction to renumber steps (assign them new sequence numbers).
- You can use this transaction to move a step to a different position within a script. When a step is moved, all references to the step are changed to reflect the new sequence number.
- You can use this transaction to delete a step.
- You can use this transaction to copy steps from other scripts. For example:
  - You may want to create a script that is similar to an existing script. Rather than copying all the information from the existing script and then removing the inapplicable steps, this page may be used to selectively copy steps from the existing script to the new script.
  - You may have scripts that are very similar, but still unique. You can use this transaction to build large scripts from smaller scripts. In this scenario, you may choose to create special 'mini' scripts, one for each of the various options that may make a script unique. Then, you could use the script merge page to select and merge the mini scripts that are applicable for a main script.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The target script must exist prior to using this page.** If you are creating a new script, you must first create the Script and then navigate to the merge page to copy step information.



> **Note**
>
> **Duplicate versus Merge**. The Script page itself has duplication capability. You would duplicate a script if you want to a) create a new script and b) populate it with *all* the steps from an existing script.

- Script Merge

Script Merge

Open **Admin**, and then **System**, and then **Script Merge** to open this page.

Description of Page

For **Original Script**, select the target script for merging steps.

For **Merge From Script**, select the template script from which to copy the steps.

> **Note**
>
> You may only copy steps from one Merge From script at a time. If you want to copy steps from more than one script, select the first Merge From script, copy the desired steps, save the original script, and then select the next Merge From script.

The left portion of the page displays any existing steps for the **Original Script**. The right portion of the page displays the existing steps for the **Merge From Script**.

You can use the **Copy All** button to copy all the steps from the **Merge From** script to the **Original** script. If you use **Copy All**, the steps are added to the end of the original script.

Copyright © 2026, Oracle and/or its affiliates.

Each time you save the changes, the system renumbers the steps in the original script using the **Start From Sequence Number** and **Increment By**.

Merge Type indicates **Original** for steps that have already been saved in the original script or **Merge** for steps that have been merged, but not yet saved. The **Sequence**, **Step Type** and **Description** for each step are displayed.

The topics that follow describe how to perform common maintenance tasks:

Resequencing Steps

If you need to resequence the steps:

- Use the up and down arrows in the Original Script grid to reorder the steps.
- Make any desired changes to the **Start From Sequence Number** or **Increment By**.
- Click Save.

The steps are given new sequence numbers according to their order in the grid.

Removing a Step from Script

If you want to remove a record linked to the Original script, click the delete button to the left of the record.

For example, to remove the **Reset existing bundle XML** step, click the  icon.

		Merge Type	Sequence	Step Type	Description
		Original	10	Edit data	Edit data - Check that the BO is an Export Bundle
		Original	20	Edit data	Edit data - Read the Bundle
		Original	30	Edit data	Edit data - Reset existing bundle XML
		Original	40	Edit data	Edit data - Create new bundle XML

After removal, the grid displays:

		Merge Type	Sequence	Step Type	Description
		Original	10	Edit data	Edit data - Check that the BO is an Export Bundle
		Original	20	Edit data	Edit data - Read the Bundle
		Original	40	Edit data	Edit data - Create new bundle XML

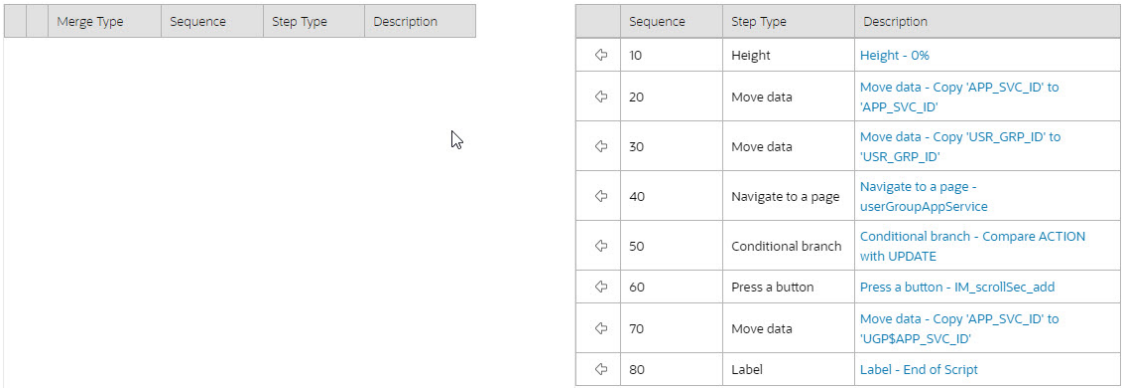
> **Note**
>
> You cannot delete a step that is referenced by other steps unless you also delete the referencing steps, such as **Go to step** or **Prompt** type steps. The system informs you of any missing referenced steps when you attempt to save the original script.

Copyright © 2026, Oracle and/or its affiliates.

Adding a Step to a Script

You can move any of the steps from the Merge From script to the Original Script by clicking the left arrow adjacent to the desired step. Once a record is moved it disappears from the Merge From information and appears in the Original information with the word **Merge** in the Merge Type column.

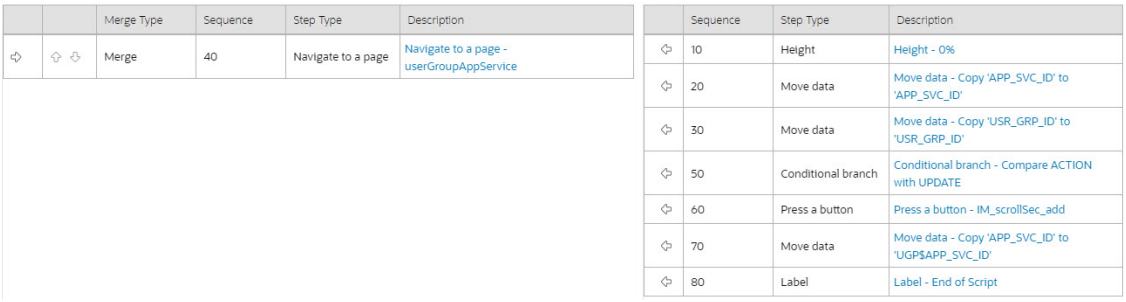
For example, to copy the **Navigate to a page** step, click the left arrow.



	Merge Type	Sequence	Step Type	Description

	Sequence	Step Type	Description
⇦	10	Height	Height - 0%
⇦	20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
⇦	30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
⇦	40	Navigate to a page	Navigate to a page - userGroupAppService
⇦	50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
⇦	60	Press a button	Press a button - IM_scrollSec_add
⇦	70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID'
⇦	80	Label	Label - End of Script

The step is moved to the left portion of the page.

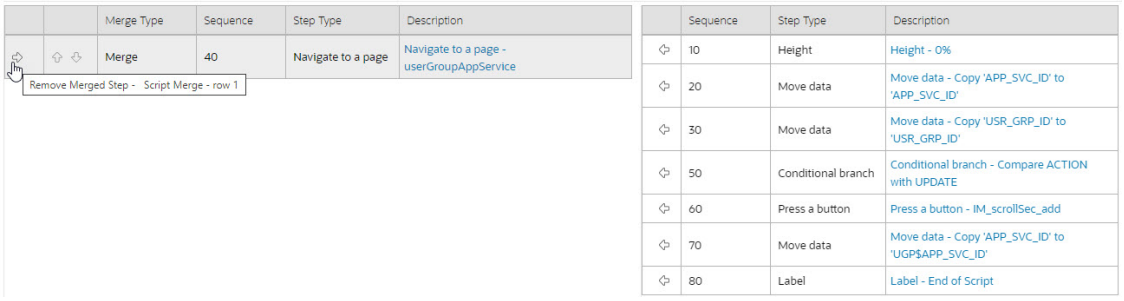


		Merge Type	Sequence	Step Type	Description
⇨	⇧ ⇩	Merge	40	Navigate to a page	Navigate to a page - userGroupAppService

	Sequence	Step Type	Description
⇦	10	Height	Height - 0%
⇦	20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
⇦	30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
⇦	50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
⇦	60	Press a button	Press a button - IM_scrollSec_add
⇦	70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID'
⇦	80	Label	Label - End of Script

> **Note**
>
> If you add a step, such as **Go to step** or **Prompt** type steps, that references other steps, you must also add the referenced steps. The step references are updated to use the new sequence numbers when you save the original script. The system informs you of any referenced steps that haven't been added when you attempt to save the original script.

Removing an Uncommitted Step from a Script



		Merge Type	Sequence	Step Type	Description
⇨	⇧ ⇩	Merge	40	Navigate to a page	Navigate to a page - userGroupAppService

Remove Merged Step - Script Merge - row 1

	Sequence	Step Type	Description
⇦	10	Height	Height - 0%
⇦	20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
⇦	30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
⇦	50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
⇦	60	Press a button	Press a button - IM_scrollSec_add
⇦	70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID'
⇦	80	Label	Label - End of Script

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Functions

> **Note**
>
> Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use one of the above configuration tool objects in a script rather than defining a function. The documentation has not been updated throughout this section to highlight where BS, SS or BO could be used to perform the equivalent logic.

Invoke function steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class. Doing this is much more efficient than the alternative of transferring to the account page and retrieving the customer class from the Main page.

An **Invoke function** step retrieves or updates the relevant data by executing a service (on the server). These types of steps do not refer to the service directly. Rather, they reference a "function" and the function, in turn, references the service.

> **Note**
>
> **Functions are abstractions of services.** A function is nothing more than meta-data defining the name of a service and how to send data to it and retrieve data from it. Functions allow you to define a scriptwriter's interface to services. They also allow you to simplify a scriptwriter's set up burden as functions can handle the movement of data into and out of the service's XML document.

The topics in this section describe how to set up a function.

> **Note**
>
> **You can retrieve data from all base-package objects.** If you know the name of the base-package "page" service used to inquire upon an object, you can retrieve the value of any of its fields for use in your scripts. To do this, set up a function that sends the unique identifier of the object to the service and retrieves the desired fields from it.

- Function - Main
- Function - Send Fields
- Function - Receive Fields

Function - Main

Use this page to define basic information about a function. Open this page using **Admin**, and then **System**, and then **Function.**

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Enter a unique **Function** code and **Description** for the function.

Use the **Long Description** to describe, in detail, what the function does.

Define the **Internal Service** that the function invokes.

> **Note**
>
> Only page services can be invoked.

The tree summarizes the following:

- The fields sent to the service. You can use the hyperlink to transfer to the **Send Fields** tab with the corresponding field displayed.
- The fields received from the service. You can use the hyperlink to transfer to the **Receive Fields** tab with the corresponding field displayed.
- Scripts that reference the function. You can use the hyperlink to transfer to the script page.

Function - Send Fields

Use this page to add or update the fields sent to the service. Open this page using **Admin**, and then **System**, and then **Function** and then navigate to the **Send Fields** tab.

> **Note**
>
> **Displaying a specific field.** Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

> **Note**
>
> **You're defining the service's input fields.** On this tab, you define which fields are populated in the XML document that is sent to the service. Essentially, these are the service's input fields.

Description of Page

Use **Sequence** to define the order of the **Send Fields**.

Enter a unique **Function Field Name** and **Description** for each field sent to the application service. Feel free to enter **Comments** to describe how the field is used by the service.

Use **Field Value Source** to define the source of the field value in the XML document sent to the service:

- If the field's value is the same every time the function is invoked, select **Defined On The Function**. Fields of this type typically are used to support "hard-coded" input values (so that the scriptwriter doesn't have to populate the field every time they invoke the function). Enter the "hard-coded" **Field Value** in the adjacent field.
- If the field's value is supplied by the script, select **Supplied By The Invoker**. For example, if the function retrieves an account's customer class, the script would need to supply the

Copyright © 2026, Oracle and/or its affiliates.

value of the account ID (because a different account ID is passed each time the function is invoked). Turn on **Required** if the invoker must supply the field's value (it's possible to have optional input fields).

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to populate the field's value in the XML document sent to the service.

Function - Receive Fields

Use this page to add or update the fields received from the service. Open this page using **Admin**, and then **System**, and then **Function** and then navigate to the **Receive Fields** tab.

> **Note**
>
> **Displaying a specific field.** Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

> **Note**
>
> **You're defining the application service's output fields.** On this tab, you define which fields are populated in the XML document that is received from the service. Essentially, these are the service's output fields.

Description of Page

Use **Sequence** to define the order of the **Receive Fields**.

Enter a unique **Function Field Name** and **Description** for each field received from the service. Feel free to enter **Comments** to describe the potential values returned from the service.

Turn on **Required** if the invoker must use the field.

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to retrieve the field's value from the XML document received from the service.

> **Note**
>
> **Fields in multiple lists.** If a field within the list is unique across the service, the generic list reference is sufficient for the XML population logic. However, if the field you are trying to reference is in multiple lists, the XPath must include the list name. For example, instead of `/pageBody/list/listBody/field[@name='FIELD_NAME']`, the XPath Population Logic must read `/pageBody/list[@name='LIST_NAME']/listBody/field[@name='FIELD_NAME']`.

Maintaining Managed Content

The Managed Content object is used to store various types of text content. Some content is used internally such as Allowlists. Other content may be used by implementations such as XSL

Copyright © 2026, Oracle and/or its affiliates.

files used for communicating messages with an external system, JavaScript include files, and CSS files. These files may then be maintained in the same manner as the HTML in UI Maps.

Open this page using **Admin**, and then **System**, and then **Managed Content**.

Use the **Managed Content Query** portal to search for an existing managed content entry. Once the managed content is selected, you are brought to the maintenance page to view and maintain the selected record.

Most fields on the page are self explanatory.

Use **Managed Content Type** to indicate the type of content.

- **Allowlist** is used to capture a list of supported values for given functionality. For example, the product provides allowlists for SQL functions and valid HTML tags.
- **XSLTC** is used to reference XSL files. These may be used internally for user interface referencing. Additionally, the XSLs may be referenced for sending messages to external system using outbound message and for receiving messages in SOAP as inbound messages.

> **Note**
>
> The value F1X (XSLT) is no longer supported and will be deprecated in a future release. Other managed content values are not currently supported by any current framework functionality and are reserved for future use.

Use the **Schema** tab to view and maintain the contents of the managed context text.

Defining Entity Tags

This portal is used to maintain an entity tag along with the list of entities associated with it. Entity tagging supports grouping of configuration, demo and test related entities so they can better describe a logical function. Each group is represented by a unique tag name and entities may be associated with multiple tags.

An entity tag may explicitly restrict the type of entities it can be associated with by referencing a specific maintenance object as part of the entity tag record.

You can access the portal from the**Admin Menu**, and then **Implementation Tools**, and then **Entity Tag**. You are brought to a query portal with options for searching for a specific tag name. Once an entity tag has been selected you are brought to the maintenance portal to view and maintain the selected record.

A Tags sidebar zone allows you to review, add and remove tags for the current entity you review. For configuration type of entities, you may use this portal to tag multiple entities at once.

The following zones may appear as part of the portal's **Main** tab page

- **Entity Tag.** This zone provides information about the selected entity tag.
- **Tagged Entities.** This zone lists entities associated with the selected entity tag. You may also use the zone to remove entities from the list.
- **Tag Entities.** Use this zone to add configuration entities to the selected tag.

Copyright © 2026, Oracle and/or its affiliates.

Advanced Schema Topics

The topics in this section describe some advanced information related to schemas used for business objects, business services, service scripts and UI maps.

- Schema Nodes and Attributes
- Schema Designer
- Schema Viewer

Schema Nodes and Attributes

For business object definition, the purpose of the schema is to create a link between the schema and a maintenance object. For business service definition you are specifying the link between the schema and a service (either a general service, search service, or a maintenance object service). For service script definition, you are defining the API for passing information to and from the script. The following documentation is a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Note**
>
> The definition of a schema may include user interface related attributes that represent a separate aspect of the schema for presentation purposes. These attribtutes belong to a separate name space known as UI Hints and as such are not part of the schema's data structure. Refer to Understanding UI Hints for more information.

The Four Element Types

A schema element can be one of four different structure types. Note that there are two classes of element types: the structural nodes group and list, and the data containing nodes of field and raw.

Mnemonic	Valid Values	Description	Examples
`type=`	`"field"`	The field type is the default type for any element not explicitly labeled as something other than a field. Therefore, you virtually never have to explicitly label an element as a field. Note that a field element, unlike group or list, will contain information in its nodes - rather than other nodes.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"group"	The group element is typically a structural element of the schema only, in which case it has no mapping. Note that when grouping several elements that are all used to map an XML structure of a CLOB / XML field of a business object driven record, the mapping may be at the group level.	Example where a group is used to create a structure <schema> <input type="group" <userId/> </input> <output type="group"> <firstName/> <lastName/> </output> </schema> Example where the group includes the mapping: <parameters type="group" mapXML="BO_DATA_AREA" mdField= "F1_TODOSUMEMAIL_PARM_ LBL"/> <numberOfDays mdField="F1_NBR_DAYS" required="true"/> <frequency mdField="F1_FREQUENCY" />

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"list"	The list node is structural node like the group node. The only difference is that the list structure has the ability to repeat multiple times in an XML document.	Example of a schema with a list: <schema> <statesList type="list"> <state isPrimeKey="true"/> <description/> </statesList> </schema> Example of a schema with a list: <xml> <statesList> <state>AK</state> <description>Alaska</description> </statesList> <statesList> <state>AL</state> <description>Alabama</description> </statesList> ... </xml>
	"raw"	The raw data type is used to capture a chunk of raw text that doesn't have any inherent structure associated with it.	<sendDetail type="raw" /> Example of an XML instance for the above schema: <sendDetail> <messageInfo> <senderAddress>123 W. Main St, Ontario, CA </senderAddress> <corpZone>3A</corpZone> </messageInfo> </sendDetail>

Copyright © 2026, Oracle and/or its affiliates.

The Data Type of a Field Element

Of the four different element types, only a field can have a data type.

Mnemonic	Valid Values	Description	Examples
dataType=	"string"	By default, a field element is a string. Therefore, there is no requirement to specify the string data type.	<schema> <custName dataType="string"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

	`"number"`	Defines an element that is a number.
		
		Note
		UI hints include a setting to SuppressAutomaticNumberFormatting.
		
		Note
		Use **currencyRef** attribute for auto-display of currency symb
			Examples
			
			`<schema>`
			`  <count dataType="number"/>`
			`</schema>`
			
			`<schema>`
			`  <taxRate dataType="number"`
			`   currencyRef="currency"/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		ol that is associated with the referenced currency code. The currency decimal positions are ignored by this formatting allowing you to display a currency symbol	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		for a unit rate with many decimals.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

	`"money"`
	Optional additional attributes
	`currencyRef=`
	"element name"
		Defines an element that represents a monetary amount.
		The currency reference is optional and if left blank the installation currency will be used. Automatic formatting and validation to be applied based on the currency. For example, the currency symbol will be shown when auto-rendering. In addition, the number of decimal places must not exceed the valid number defined for the currency.
		Note
		Refer to Referencing Other Elements for supported syntax for referring to other elements.
			`<schema>`
			`  <currency default="USD"`
			`    suppress="true"/>`
			`  <balance dataType="money"`
			`   currencyRef="currency"/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"lookup" Required additional attribute lookup= "field name"	Defines an element that has valid values defined using a lookup. The lookup field is required.	<schema> <status dataType="lookup" lookup="STATUS_FLG"/> </schema>
	"lookupBO" Required additional attribute lookupBO= "bo name"	Defines an element that has valid values defined using an extendable lookup. The extendable lookup's business object is required.	<schema> <category dataType="lookupBO" lookupBO="CM-BusinessCategory"/> </schema>
	"boolean"	Defines an element that has values of "Y" and "N".	<schema> <allowsEdit dataType="boolean"/> </schema>
	"date"	Defines an element that represents a date.	<schema> <startDate dataType="date"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

	`"dateTime"`	Defines an element that represents a date and time.
		
		Note
		Refer to Standard Time Considerations for additional configuration available for date / time fields that represent standard time.
			`<schema>`
			`  <startDateTime`
			`  dataType="dateTime"/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"time"	Defines an element that represents a time.	<schema> <startTime dataType="time"/> </schema>
	"uri"	Defines an element captures a URI. Elements defined with this type enable the support for URI Allowlist and Substitution as described in Referencing URIs .	<schema> <exportDirectory dataType="uri"/> </schema>

Referencing Other Elements

There are several attributes that allow for a reference to another element in the same schema. The supported syntax of the XPath reference is the same in every case. This section provides examples using the default reference attribute (**defaultRef**).

Reference a sibling element:

```
<schema>
   <id mapField="ACCT_ID" required="true"/>
   <altId defaultRef="id" required="true"/>
</schema>
```

Reference an element in a higher group:

```
<schema>
   <id mapField="ACCT_ID" required="true"/>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId defaultRef="../id" required="true"/>
   </msgInfo>
</schema>
```

Reference an element in a lower group:

```
<schema>
   <id mapField="ACCT_ID" defaultRef="msgInfo/altId" required="true"/>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId required="true"/>
   </msgInfo>
</schema>
```

Reference an element in another group:

```
<schema>
  <acctInfo type="group">
    <id mapField="ACCT_ID" required="true"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
  </acctInfo>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId defaultRef="../acctInfo/altId" required="true"/>
   </msgInfo>
</schema>
```

Legal vs. Standard Time Considerations

Most date / time fields represent "legal" time such that if a time zone changes their clocks for winter and summer time, the date / time field captures the current observed time. However, some date / time fields should always be captured in standard time to avoid confusion / ambiguity. A good example is a date and time related to detailed interval data. Refer to Designing Time Zones for more information.

When defining an element with `dataType="dateTime"`, you may use `legalTime="true"` to indicate that data captured in the element always represents legal time in the 'base' time zone, or use `stdTime="true"` to indicate that data is stored in standard time in the 'base' time zone. The 'base' time zone is specified on the Installation options.

> **Note**
>
> If an element is mapped to a table / field with a Standard Time Type of **Physical**, then `stdTime="true"` is implied. Refer to Fields for more information.

Example:

```
<schema>
  <startTimeS dataType="dateTime" stdTime="true"/>
  <startTimeL dataType="dateTime" legalTime="true"/>
</schema>
```

If the time zone that represents the date / time field is not the installation time zone, use the optional setting `stdTimeRef`="XPath to time zone element" to indicate that the element represents standard time and indicates the time zone to use. Use `legalTimeRef`= when the element is stored in legal time. Refer to Referencing Other Elements for supported syntax for referring to other elements.

Example:

```
<schema>
  <alternateTimeZone fkRef="F1-TZONE" suppress="true"/>
  <startDateTimeS dataType="dateTime" stdTimeRef="alternateTimeZone"/>
  <startDateTimeL dataType="dateTime" legalTimeRef="alternateTimeZone"/>
</schema>
```

> **Note**
>
> If an element is mapped to a table / field with a Standard Time Type of **Referenced**, then `stdTimeRef`="XPath" is implied. Refer to Fields for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When schema elements are captured in standard time the UI map supports HTML notation to automatically display the data applying a daylight savings time / summer time correction. Refer to the HTML attribute oraType="dateTime;" for more information.

There may be cases where the date / time is captured in one time zone, but should be displayed using a different time zone. In this case, the attribute `displayRef`="XPath" may be used in addition to the appropriate attribute that identifies the time zone that the data is capture in. Refer to Referencing Other Elements for supported syntax.

```
<schema>
  <displayTimeZone fkRef="F1-TZONE" suppress="true"/>
  <startDateTimeS dataType="dateTime" stdTime="true"
displayRef="displayTimeZone"/>
  <startDateTimeL dataType="dateTime" legalTime="true"
displayRef="displayTimeZone"/>
</schema>
```

The Mapping Attributes

When constructing your schema, you can choose from one of the following mapping attributes.

Mnemonic	Valid Values	Description	Examples

`mapField=`	"field name"	In the case of a business object, the **mapField** attribute is used to identify the database column the element is related to. For business service schemas, the `mapField=` attribute is used to link a schema element with a service element.	`<schema>`
			`  <factId`
			
			`mapField="FACT_ID"/>`
			`</schema>`

`mapChild=`	"table name"	The **mapChild** attribute is used only for business object mapping. It is used in two ways:
		• First, to create a list in the business object that corresponds to a child table of an MO.
		• Second, you can use **mapChild** to identify the child table a flattened field lives in. For more information on flattening, refer to flattening section below.
			Example of a list within a child table in a BO:
			
			`<persons type="list"`
			
			`mapChild="CI_ACCT_PER"`
			`  <personId`
			`mapField="PER_ID"/>`
			`</persons>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
mapList=	"list name"	The **mapList** attribute is used only for business service mapping. It is used in two ways: • First, to create a list in the business service that corresponds to a list in the service. • Second, you can use **mapList** to identify the list that a flattened field lives in. For more information on flattening, refer to flattening section below.	Example of a list within a business service: <selectList type="list" mapList="DE" <value mapField="COL_VALUE"> <row mapList="DE_VAL"> <SEQNO is="2"/> </row> </value> </selectList>
mapXML=	"field name"	The **mapXML** attribute is typically used to map XML structures into a large character / XML field of the service. Note that when you use **mapXML** to map either a list or group structure (type="list" or type="group") you don't have to map all the child elements within the structure. It is also possible to map list elements to a large character field associated with the list child.	<enrollmentRequest type="group" mapXML="CASE_CLOB"> <messageID/> <sender/> </enrollmentRequest> <enrollmentKey mapXML="CASE_CLOB"/> <enrollmentInfo type="list" mapChild="CI_CASE_CHILD"> <sequence mapField="CHILD_SEQ"/> <name mapXML="CASE_CHILD_CLOB"/> <value **mapXML="CASE_CHILD_CLOB"**/> </enrollmentInfo>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`isPrimeKey=`	`"true"`	You must specify a primary key for a list defined within a mapped XML element (`type="list" mapXML="CLOB"`). The primary key is used by the framework to determine whether a list element add, update or delete is required during a business object update.
		
		Note
		You do not need to specify the prime key for a business object list mapped to maintenance object list. Whe
			`<questionnaire type="list" mapXML="CASE_CLOB">`
			`  <question isPrimeKey="true"/>`
			`  <answer/>`
			`</questionnaire>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		n a ph ysi cal list is ma pp ed, the pri me ke y is de riv ed fro m exi sti ng ph ysi cal me ta- dat a.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
orderBy=	"XPath asc|desc , XPath asc|desc "	By default, a list defined within a mapped XML element (type="list" mapXML="CLOB") is sorted by the first element of the list. A different sort order may be specified using the **orderBy** attribute. The attribute value is a comma separated list of field XPaths (relative to the list element) with an optional sort order (ascending is the default). **Note** This is only available for lists mapped as XML within business objects.	<questionnaire type="list" orderBy="page/section, page/sequence" mapXML="CASE_CLOB"> <question isPrimeKey="true"/> <answer/> <page type="group"> <section/> <sequence/> </page> </questionnaire>

Descriptive Attributes

The following attributes can be used to describe a schema element and provide additional configuration related to the element. Typically, these attributes are useful for field elements only.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`<!-- comment -->`		Use this to add a comment to a schema by using special open and close characters: <!-- and -->.	`<schema>`
			`  <!-- This schema is used to capture business information only, please refer to the 'HUMAN' BO for person information -->`
			`</schema>`

`description=`	"text"	The description of an element may be used to provide an internal description of the element to help a reader of the schema to understand the business reason for the element.	`<schema>`
			`  <active type="boolean" description="active account" label="Active"/>`
			`</schema>`

`label=`	"text"	The label of an element is meant to be a short bit of verbiage that would typically precede the element in a user interface.	`<schema>`
			`  <active type="boolean" description="active account"` **`label="Active"`**`/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`required=`	`"true"`	Used to require the existence of an element during object interaction.
		
		Note
		For included schemas, the required="true" attribute is not processed on business object and business service schemas when
			`<schema>`
			`  <logDate`
			`mapField="LOG_DT"`
			`default="%CurrentDate"`
			
			`required="true"`
			`private="true"`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		they are included within a service script schema. This is to support the ability of the service script to populate required elements before an embedded busin	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		ess object or business service is invoked from the service script.	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`mdField=`	"field code"	The meta-data field attribute is used to associate an element with a field's metadata. The field defines data type, as well as its label and help text. If you link an element with a meta-data field, you don't need to specify any of these attributes.	`<schema>`
			`  <active mdField="CM_ACTIVE_SW" />`
			`</schema>`

Note

For a business object schema, the mapField attribute is used to derive the data type and label. An mdField attrib

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		ute may be provided to override these attributes, if needed. If the element is mapped to an XML column, the mdField is needed to provide the appr op	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		riate data type and label.	
`fkRef=`	"FK Reference Code"	If the element is a foreign key, defining its FK Reference will enable framework validation of the element during schema interaction and automatically provide descriptions and navigation capability.	`<schema> <person fkRef="PER" mapField="CHAR_VAL_FK1"> <row mapChild="CI_SA_CHAR"> <CHAR_TYPE_CD is="PER"/> </row> </person> </schema>`
`private=`	`"true"`	Marking an element as private will prevent it from being exposed in schema interaction. **Note** A private element requires a default.	`<schema> <type mdField="SA_TYPE_CD" default="E1" private="true"/> </schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
suppress=	"true"	This setting prevents an element from appearing in automatically generated user interfaces. **Note** This attribute can be specified on a group, in which case all elements of the group will be suppressed.	<schema> <ls mdField="LIFE_SUPPORT_FLG" default="N" suppress="true"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		Note Refer to Defaulting and System Variables for advice related to default values for suppressed elements.	

	`"blank"`	This setting means that automatic UI rendering in display mode will hide the element if its value is blank. The element will still be modifiable on the input map whether blank or not.	`<schema>`
			`  <email mdField="EMAIL" suppress="blank"/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

	`"input"`	This setting means that automatic UI rendering will suppress the element for input, although it may still be displayed if it is not blank.
		
		Note
		Elements marked as `suppress="input"` will behave as with `suppress="blank"`. If the value is blank, no value will
			`<schema>`
			`  <email mdField="EMAIL" suppress="input"/>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		be displayed on either the display or input map. If the element's value is present, then the element will be displayed on both the display and inpu	

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

t map.

Note

Refer to Defaulting and System Variables for advice related to default values for suppressed elements.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
noAudit=	"true"	This setting prevents an element from appearing as a changed element in the business object's audit plug-in spot. If specified on a group or list node it applies to the whole node. You cannot specify it on the schema root node, only on schema elements. **Note** This attribute is only applicable to business object schemas.	<schema> ... <version mapField="VERSION_NBR" noAudit="true"/> ... <workFields type="group" noAudit="true" <lastProcessedId/> <lastProcessedTime/> </workFields> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`storeEmptyNodes=`	`"true"`	By default, empty nodes are removed from a business object instance when saved. This setting allows a group or a list element to explicitly keep its empty nodes.
		This attribute may be useful in situations where business object data is exchanged with an external system and there is a need to distinguish between an empty value for an element that participates in the synchronization and an element that does not participate and therefore omitted from the message altogether.
		Note
		This is only available for group and list elements.
			`<schema>`
			`  <message type="group" storeEmptyNodes="true">`
			`  ...`
			`  </message>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

`emitEmptyGroups=`	`"true"`	By default, empty nodes are not included in the output of a business object or a business service call. This attribute is set at the top level schema node, allowing empty nodes of all groups and list elements to be included in the output.
		This attribute may be useful in the same situations that may require the use of the **storeEmptyNodes=** attribute.
		Note
		This is only available for business object and business service schemas.
			`<schema emitEmptyGroups="true" >`
			`  <message type="group">`
			`  ...`
			`  </message>`
			`</schema>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
emitEmptyElements=	"true"	This attribute is similar to the **emitEmptyGroups**= attribute but may also be specified at the field level. The attribute is applicable to all schema types.	<schema> <messageNumber type="field" emitEmptyElements="true"> ... </message> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
adheresToDA=	“comma separated list of data area names.”	This attribute is applicable to the schema, group and list nodes. It claims that this sub-schema node is intended to "look like" the schema of one of the listed data areas, for polymorphism reasons in general. **Note** The recommended approach for enforcing the structure of a shared data area is by including it in the schema. Thi	<schema> <messageNumber type="group" adheresToDA="DataAreaName"> ... </message> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
		s attribute should be limited to use cases where the ability to include the data area is not possible.	

Schema Constants

There are some product owned schemas where the design warrants a value to be defaulted in the schema, but where the value is implementation specific and therefore cannot be defined by the product. For these situations, the product may use a technique called a schema constant. The design of the schema will include a declared constant. At implementation time, a configuration task will include defining the appropriate value for the constant.

For example, imagine the product delivers an algorithm that will create an outbound message when a certain condition occurs. The outbound message type to use must be configured by the implementation. To use a schema constant to define the outbound message type, the base product will configure the following:

- An option type lookup value for the lookup `F1CN_OPT_TYP_FLG` is defined. For example, `M202` - Activity Completion Outbound Message Type with a Java Value Name of **outmsgCompletion**

Copyright © 2026, Oracle and/or its affiliates.

- The base schema that is used to create the "complete activity" outbound message references the schema constant using the Java Value Name of the option type's lookup value

```
...
<outboundMessageType mapField="OUTMSG_TYPE_CD"
default="%Constant(outmsgCompletion)"/>
...
```

At implementation time, the administrative users must configure the appropriate outbound message type for "activity completion". Then, navigate to Feature Configuration, choose the **Schema Constants** feature type, choose the option type **Activity Completion Outbound Message Type** and enter the newly created outbound message type in the option value.

Schema constants may also be used in the flattening syntax to define the row elements required for flattening.

Defaulting and System Variables

The default attribute can be used to default values into field elements as well as the row elements required for flattening . You can default a field to a constant or to one of several system variables.

When an element is required, the default value is applied at the server level when adding or changing the record where no value is provided for this element.

When an element is optional, a default value may be configured to provide a suggested value to the user when populating the record on the user interface in Add mode. Users should be able to remove the default value for this type of element and the system will not attempt to populate it at save time.

> **Note**
>
> The standard for our 'list' maintenance is to show an empty row for a new list that a user may or may not choose to populate. As such, UI default values are not shown for list elements.
>
> It is not recommended to configure a default attribute for an optional element that is not editable or visible on the user interface in Add mode (for example an element with `suppress="true"` or `suppress="input"` or a list element whose default value is not shown per the previous point). The default for this type of element is still applied but the user doesn't have the ability to reset the value. This may result in inconsistencies as defaulting does not occur for optional fields added by a use case other than the UI interaction.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
default=	"value"	Use this attribute to default an element to a specified value. The values that are valid depend on the **dataType** setting.	<schema> <perType mapField="PER_OR_BUS_FLG" default="P" required="true"/> </schema>
			<schema> <frequency dataType="number" default="1" required="true"/> </schema>
	"%CurrentDate"	Used to default the element to the current date. This is only applicable to date elements.	<schema> <logDate mapField="LOG_DT" default="%CurrentDate" required="true"/> </schema>
	"%CurrentDateTime"	Used to default the element to the current date / time. This is only applicable to date / time elements.	<schema> <logDateTime mapField="LOG_DTTM" default="%CurrentDateTime" required="true"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"%StandardDateTime"	Used to default the standard date and time. The standard date and time is identical to the current date and time, unless daylight savings time / summer time is in effect for the base time zone. This may be used with the **stdTime** attribute. **Note** Refer to Standard Time Considerations for more information.	<schema> <startDateTime mapField="START_DTTM" default="%StandardDateTime" required="true"/> </schema>
	"%ProcessDate"	You can default the process date. The process date differs from the current date because the process date will remain constant throughout the duration of the process being executed. The current date will reflect the actual date of processing. This is similar to the batch business date that is a standard batch parameter.	<schema> <billDate mapField="BILL_DT" default="%ProcessDate" required="true"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"%ProcessDateTime"	This is similar to "%ProcessDate" but for date / time fields.	<schema> <calcDateTime mapField="CALC_DTTM" default="%ProcessDateTime" required="true"/> </schema>
	"%CurrentUser"	Used to default the element to the current user.	<schema> <logUser mapField="LOG_USER" default="%CurrentUser" required="true"/> </schema>
	"%CurrentUserTimeZone"	Used to default the element to the current user's time zone. If the current user's time zone is not found, the installation time zone is used.	<schema> <timeZone default="%CurrentUserTimeZone" required="true"/> </schema>
	"%CurrentUserLanguage"	Used to default the element to the current user's language.	<schema> <custLanguage mapField="CUST_LANG" default="%CurrentUserLanguage" required="true"/> </schema>
	"%InstallationCurrency"	Used to default the currency from the installation record.	<schema> <currency mapField="CURRENCY_CODE" **default="%InstallationCurrency"** required="true"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"%InstallationCountry"	You can default the country from installation record.	<schema> <country mapField="COUNTRY" default="%Installation Country" required="true"/> </schema>
	"%InstallationLanguage"	You can default the language from the installation record.	<schema> <language mapField="LANGUAGE" **default="%Installation Language"** required="true"/> </schema>
	"%Constant( )"	You can default an element value using a schema constant.	The following is an example of a schema constant used as a default value, where the Java Value Name of the Lookup Value is 'customerLanguage'. <language mapField="CUSTOMER_LANG" default="%Constant(customerLanguage)" required="true"/>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"%Context( )"	You can default a value contained in a context variable.	An example of a context variable used as a default value: `<source mapField="PER_ID" default="%Context(source)" required="true"/>`

Warning

Context variables must be initialized within a server script before the schema context default can be applied. Refer to Conte

Note

When defining a context variable in scripting, it should be prefixed with $$. When referring to the varia

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
			xt Variables for more information. ble in the **%Context()** syntax, the prefix is not included.
`defaultRef=`	"XPath"	Use this attribute to default the value of one element to the value of another one.	Refer to Referencing Other Elements for supported syntax for referring to other elements.

The Flattening Nodes and Attributes

The term "flattening" is used to describe the act of defining one or more single elements for a schema that are actually part of a list within the maintenance object. Flattening is possible if there are other attributes of the list that can be defined to uniquely describe the element or elements. A common use case for flattening is a characteristic. Rather than defining the characteristics of an object using a collection where the user must choose the characteristic type and then define the value, the characteristics are defined as actual elements with the appropriate label already displayed. This technique enables the designer of the schema and the user interface to display each separate characteristic in the logical place in the user interface rather than all lumped together.

> **Note**
>
> A flattened element represents a unique row in the database. This row is inserted when the flattened values are created. The row is updated when any of the flattened values are changed. The row is deleted when all the flattened values are removed. The behavior of effective dated elements is slightly different - please see Flattening an Effective Dated List.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The flattening feature can also be used to define a list, see Flattened List.

Identifying the List or Child Table

When flattening a child table, the row node is required to identify the list / child table that the element comes from. Within the row node, at least one element must be defined with an `is=` definition that ensures that the element is a unique row in the database. It may also define elements or fields in the row that are suppressed and are populated using default value configuration.

- For a business object, the row node defines the child table the flattened field belongs to.

  The syntax is **<row mapChild="table name">**. This example is for the list of persons for an account in the customer care and billing product. One person may be marked as the "main" person. This illustrates how to define an explicit element for the main person ID to simplify references to that field. It is part of the CI_ACCT_PER child table. What makes it unique is that the MAIN_CUST_SW is **true** (and only one row may have that value)

  ```
  <custId mapField="PER_ID" mdField="CM-MainCust">
    <row mapChild="CI_ACCT_PER">
      <MAIN_CUST_SW is="true" />
      <ACCT_REL_TYPE_CD default="MAIN" />
    </row>
  </custId>
  ```

  > **Note**
  >
  > The above example illustrates that the row node may also define elements within the list that are suppressed and assigned a default value. This syntax is never used to identify a particular row. Note that a default value can either be a literal string, or a system variable.

- For a business service, the row node identifies the list name the flattened field belongs to.

  The syntax is **<row mapList="list name">**. This example shows two entries from a list being flattened to a field value and description.

  ```
  <selectList type="list" mapList="DE">
      <value mapField="COL_VALUE">
      <row mapList="DE_VAL">
            <SEQNO is="2"/>
      </row>
      </value>
    <description mapField="COL_VALUE">
      <row mapList="DE_VAL">
          <SEQNO is="3"/>
      </row>
    </description>
  </selectList>
  ```

Uniquely Identifying the Flattened Field or List

The `is=` syntax within a row or rowFilter element is used to uniquely identify the row.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
is=	"value"	Use this attribute to reference a value directly.	<tdTypeCd mapField="CHAR_VAL_FK1"> <row mapChild="F1_EXT_LOOKUP_VAL_CHAR"> <CHAR_TYPE_CD **is="CM-TD-TYPE"**/> </row> </tdTypeCd>
	"%Constant( )"	You can configure a flattened element using a schema constant. During a service interaction the value of the schema constant will be used to identify the flattened element in its child row.	An example of a schema constant used in flattening syntax where the Java Value Name of the Lookup field value is 'cmRate'. <unitRate mapField="CHAR_VAL" dataType="number"> <row mapChild= "F1_EXT_LOOKUP_VAL_CHAR"> <CHAR_TYPE is **="%Constant(cmRate)"**/> </row> </unitRate>
	"%n"	The %n substitution value is used to reference a relative list instance. A relative list instance is typically used to configure a flattened element for a child table keyed by sequence number. The value of *n* should be a positive integer value. During a business object read interaction the relative list instance (position) specified by the integer will be returned.	An example with a relative list instance - where the **first** instance of the row is returned. <schema> <mainPhone mapField="PHONE"> <row mapChild="CI_PER_PHONE"> <PHONE_TYPE_CD is="%Constant(mainPhoneType)"/> <SEQ_NUM**is="%1"**/> </row> </mainPhone> </schema>

NOT_SUPPORTED

Additional values for `is=` are used when Flattening an Effective Dated List. Refer to that section for more information.

Flattening a Pre-defined Characteristic Type

Copyright © 2026, Oracle and/or its affiliates.

If the flattened field is in a characteristic collection and the characteristic type is a predefined characteristic, automatic UI rendering will generate a dropdown for the list of valid values. For example, the schema below will generate a dropdown for the Usage element showing the valid values of the Status Reason Usage (F1-SRUSG) characteristic type.

```
<usage mdField="STATUS_RSN_USAGE" mapField="CHAR_VAL">
  <row mapChild="F1_BUS_OBJ_STATUS_RSN_CHAR">
      <CHAR_TYPE_CD is="F1-SRUSG"/>
      <SEQ_NUM is="1"/>
  </row>
</usage>
```

Defining Multiple Elements from the List

When attempting to include multiple columns from the same list, the system provide shorthand notation for copying the flattening rules defined on another element so that the flattening rules do not need to be repeated. To do this, the row node includes the **rowRef** attribute and it indicates the other element name that defines the mapping information. The following example illustrates flattening the fields Customer ID and Receives Copy of Bill from the same list of Persons for an Account (where the MAIN_CUST_SW is **true**).

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
      <MAIN_CUST_SW is="true" />
      <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<copyBill mapField="RECEIVE_COPY_SW" rowRef="custId"/>
```

Note that the above notation also illustrates that the **rowRef** may be defined directly in the element's attribute definition.

> **Note**
>
> Refer to Referencing Other Elements for supported syntax for referring to other elements.

Flattening Two Layers Deep

If your maintenance object or service has nested lists two layers deep, the system supports flattening an element within a flattened element. This technique also uses the **rowRef** attribute. The flattening of the second level refers to the flattened element of the first level. The following example illustrates flattening a characteristic into an element called legalContact for the "main" customer. Notice that the legalContact element has two row nodes: one to refer to the flattening information for its parent record and one to define its child table

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
    <MAIN_CUST_SW is="true" />
    <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<legalContact mapField="CHAR_VAL_FK1">
  <row rowRef="custId">
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <row mapChild="CI_ACCT_PER_CHAR" >
       <CHAR_TYPE_CD is="LEGAL" />
    </row>
  </row>
</legalContact>
```

Note that the above notation also illustrates that the **rowRef** may be defined as an attribute of a row node rather than directly in the element's attribute definition.

Defining a Flattened List

There are times that a list or child table supports multiple values of the same "type" and these should be presented as a list. To continue with the example above, the list of persons for an account may identify one person as the "main" person. This person has been flattened to a single element (with the account relationship type defaulted and suppressed). To maintain the other persons related to an account, you can define a list where each row captures the Person Id and the Account Relationship Type.

Rather than a row node, the flattened list is configured with a **rowFilter** element. The following schema illustrates the described example. The list node defines the child table. The **rowFilter** includes the criteria that identify the rows within the table to include. The elements of the list are defined within the list node outside the **rowFilter** element.

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
    <MAIN_CUST_SW is="true" />
    <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<miscPersons type="list" mapChild="CI_ACCT_PER">
  <rowFilter>
    <MAIN_CUST_SW is="false" />
  </rowFilter>
  <relType mapField="ACCT_REL_TYPE_CD"/>
  <personId mapField="PER_ID"/>
</custId>
```

Note that the system will validate that if a schema contains flattened single elements and flattened lists from the same child table, the criteria that defines what makes them unique must be analogous.

Flattening an Effective Dated List

There are some lists in the application that are effective dated (and still others that have effective date and time). For example, there are some effective dated characteristic collections. In these collections, the design is to capture a single value for a characteristic type that may change over time. It is not meant to support multiple characteristic values in effect at the same time.

There are some maintenance objects that have effective dated characteristics and no sequence based characteristics. For those entities, implementations may want to use the characteristic to define an additional field that doesn't require effective dates.

When considering designing an element for your schema that represents a value in an effective dated collection, consider if the value is one that is used in processes, like calculating a bill, such that it is important to capture changes to the value over time.

Copyright © 2026, Oracle and/or its affiliates.

- If it is not important to track the changes to the value, the recommendation is to set the effective date to a fixed value in the schema. (Examples below).
- If it is important to track the changes to the value, then you need to consider how you want to design the schema.
  - One option is to define a flattened list as described above. Both the value and the effective date are elements in this list. In this use case, you manage the list just like any list. You can add or remove values.
  - Another option is to define a flattened single value in the schema. With this option, when retrieving the record, the latest effective value is returned. In this design, the important question is how is the reference date defined.

> **Note**
>
> Refer to Referencing Other Elements for supported syntax for referring to other elements.

Flattened Elements Where Changes Are Not Tracked

The following is an example of flattening an element in an effective dated collection where the value does not need to be effective dated. The recommendation in this case is to simply hard-code the date in the `is=` attribute.

```
<schema>
  <externalReference mapField="ADHOC_CHAR_VAL" mdField="CM_EXT_REF"
dataType="string">
    <row mapChild="CI_SA_CHAR">
      <CHAR_TYPE is="CM_EXT_REF"/>
      <EFFDT is="2000-01-01"/>
    </row>
  </externalReference>
</schema>
```

Flattened Elements Where Changes Are Effective Dated

If you want the BO schema to flatten an effective dated value that should keep track of changes over time, the flattened value will represent the latest effective entry. In this case, the recommendation is to explicitly define the date or date/time field as an element. Then there is syntax for the `is=` attribute to indicate that the explicit date field is the date to use for the value.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
is=	"%effectiveDate( )"	Use this configuration to indicate that the date to use is the value of another element. **Note** Refer to Referencing Other Elements for supported syntax for referring to other elements.	<schema> <price mapField="CHAR_VAL" dataType="number" required="true"> <row mapChild="CI_SA_CHAR"> <CHAR_TYPE is="PRICE"/> <EFFDT is="%effectiveDate(priceEdate)" /> </row> </price> <priceEdate mapField="EFFDT" rowRef="price"/> </schema>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"%effectiveDateTime()"	Use this configuration to indicate that the date / time to use is the value of another element. **Note** Refer to Referencing Other Elements for supported syntax for referring to other elements.	<schema> <price mapField="CHAR_VAL" dataType="number"> <row mapChild="RATE_CHAR"> <CHAR_TYPE is="PRICE"/> <EFFDTTM is="%effectiveDateTime(priceEdatetime)"/> </row> </price> <priceEdatetime mapField="EFFDTTM" rowRef="price"/> </schema>

You should consider if the explicit date or date/time element should be displayed on the user interface when the flattened field is shown or if that should be suppressed. Designers should also consider the user dialogue for updating the value. It may be more intuitive for the flattened field to be display only and to have a separate dialogue for updating the effective dated value.

For this option, the suggestion is that the maintenance page for the object includes a separate zone that shows the history of the effective dated values over time. The latest value may be included in that zone as well.

Note that for backward compatibility, the system also supports the following options. However, they are not recommended. They implicitly defined the reference date as the current date at all times. The options supported adding new effective dated rows when the value changed. But the technique also caused new effective dated rows to appear when other elements of the same schema were updated. The new rows would have the current date or date/time with the same row value because the reference date or date/time is the current date, which is a

Copyright © 2026, Oracle and/or its affiliates.

change. The product does not recommend managing an effective dated characteristic without defining the date or date/time as an explicit element.

Mnemonic	Valid Values	Description	Examples
is=	"%effectiveDate"	This configuration is not recommended	<schema> <price mapField="CHAR_VAL" dataType="number"> <row mapChild="CI_SA_CHAR"> <CHAR_TYPE is="PRICE"/> <EFFDT is="%effectiveDate"/> </row> </price> </schema>
	"%effectiveDateTime"	This configuration is not recommended	<schema> <price mapField="CHAR_VAL" dataType="number"> <row mapChild="RATE_CHAR"> <CHAR_TYPE is="PRICE"/> <EFFDT is="%effectiveDateTime" /> </row> </price> </schema>

Search Zone

A UI Map schema element can be configured to enable an automatic search dialog when the schema is included within a maintenance UI map.

> **Note**
>
> Please note that an fkRef can be configured with a search zone. If a schema element has an fkRef but no explicit search attributes (as described here) then the fkRef search information will be used in the UI map. In other words, if the schema element already has an fkRef, then these explicit search attributes in the schema are only used to override the fkRef search information.

> **Note**
>
> Refer to the UI Map Attributes and Functions for more information on search zone configuration.

search="search zone"

Copyright © 2026, Oracle and/or its affiliates.

The search attribute can be used within a UI map schema and is used to automatically generate the oraSearch UI map attribute. The search zone is an explorer zone configured as a search.

```
<person fkRef="PER" search="C1_PSRCH"/>
```

searchField="search field:element|'literal';"

The searchField attribute can only be used in conjunction with the search attribute. The searchField attribute is used to build the oraSearchField UI map attribute. The searchField value is used to populate a search zone filter with an initial value when the search zone is launched. The initial value can be a literal also. The searchField value is used to match to the filter mnemonic also named searchField.

Search field: element|'literal'. The search field represents the search zone filter to populate on launch. The element is the map's schema element used to populate the filter. The element is optional, if left blank, it will default to the element that this attribute is specified on. The searchField also takes 'literal' as input value

> **Note**
>
> Multiple filters can be populated within the search zone at launch, but multiple search field pairs must be constructed within the attribute value. The value specified here will be used to directly build the HTML attribute oraSearchField within the UI map where this schema is specified.

> **Note**
>
> Note that the element reference is *relative*. Refer to Referencing Other Elements for supported syntax for referring to other elements.

```
<person fkRef="PER" search="C1_PSRCH" searchField="PERSON;
PER_TYPE:personType;"/>
```

searchOut="search field:element;"

The searchOut attribute can only be used in conjunction with the search attribute. The searchOut attribute is used to build the oraSearchOut UI map attribute. The searchOut value is used to capture a selected value from the search zone and move it to a UI map element. The searchOut value specified should match the ELEMENT_NAME mnemonic within the search result zone parameter.

Search field: element. The search field represents the search zone result brought back to the UI map. The element is the map's schema element to be populated. The element is optional, if left blank, it will default to the element that this attribute is specified on.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Multiple elements can be populated as a result of search zone selection, but multiple search field pairs must be constructed within the attribute value. The value specified here will be used to directly build the HTML attribute oraSearchOut within the UI map where this schema is specified.

> **Note**
>
> Note that the element reference is *relative*. Refer to Referencing Other Elements for supported syntax for referring to other elements.

```
<person fkRef="PER" search="C1_PSRCH" searchField="PER_ID"
searchOut="PER_ID;PRIMARY_PHONE:personPhone;"/>
```

Extend Security for Service Script

Application service security will be enforced when either a business object or a service script is invoked from a BPA script or a UI map, but not from a service script. If you want security to be enforced when the business object or a service script is invoked from a service script, you must add the following attribute to the service script's schema.

appSecurity="true"

The appSecurity attribute is only available for service script schemas. If specified, any business object or service script directly invoked by the service script will have their application service evaluated for access.

```
<schema appSecurity="true">
  ...
</schema>
```

Overriding Action for a Business Service

If you want to invoke a business service with an action other than 'read', you need to specify the action attribute on the primary node business service schema.

pageAction="add, change, delete"

The action attribute is used to override the default action of read on a business service schema. Valid values are:

- add
- update (only allowed for maintenance object service)
- change (not allowed for maintenance object service)
- delete
- read (this is the default action if no pageAction specified)

Example:

```
<schema pageAction="change">
  <parm type="group">
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <ele1/>
    <ele2/>
  </parm>
</schema>
```

Specifying searchBy for a Search Service

If you want to invoke a search service then you must explicitly specify the searchBy attribute appropriate for the elements mapped in the schema.

searchBy="MAIN"

The value values of the searchBy attribute can be found by viewing the XML schema linked to the business service, use the View XML URL. Typical values are:

- MAIN
- ALT
- ALT2
- ALT3
- etc.

```
<schema searchBy="MAIN">
  <AccountID mapField="ACCT_ID"/>
  <Results type="list">
    <AccountID mapField="ACCT_ID"/>
  </Results>
</schema>
```

Including Other Schemas

There are no limitations on your ability to include a schema into another schema - all types can be included in all other types. Nested includes are also allowed - and at present there is no limitation on the depth of the nesting.

Including a schema requires two parts:

1. The include node
2. The name attribute

The following table highlights the supported include statements.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description	Examples
<includeBO name=" "/>	Including a business object schema into another schema is allowed. Note that the mapping rules of a business object or business service schema may or may not make sense in the context of the parent schema. Include other schemas at your own risk. However, a very useful aspect of XML processing is that the framework ignores non-pertinent attributes. In other words, it will not hurt to have mapping attributes included into a script schema.	<schema> <cust type="group"> <includeBO name="C1-Person"/> </cust> <spouse type="group"> <includeBO name="C1-Person"/> </spouse> </schema>
<includeBS name=" "/>	Used to include a business service schema.	<schema> <includeBS name="F1-ReadMOLog"/> </schema>
<includeDA name=" "/>	Used to include a data area schema.	<schema> <includeDA name="F1CommonSchemaFieldData"/> </schema>
<includeMP name=" "/>	Used to include a UI map schema.	<schema> <includeMP name="F1-DisplayRecordActions"/> </schema>
<includeSS name=" "/>	Used to include a service script schema.	<schema> <includeSS name="F1-ActShowZn"/> </schema>

Compatibility Attributes

These attributes were added as part of upgrades from previous versions of the Framework.

fwRel="2"

This attribute has been added to schemas created in Framework 2 as part of a Framework 4 upgrade. New schemas will not need this attribute. It is not advisable to modify this attribute without understanding the following behavior differences as improper changes could result in errors:

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Schemas created in Framework 2 with the fwRel="2" attribute will store any XML mapped fields under groups as top-level XML elements in the mapXML field. This means that if two or more fields, in different group structures, were to have the same field name, their storage would conflict with one another resulting in errors. The new behavior, without the fwRel="2" attribute, will preserve the group structure and avoid the conflicts.

```
<schema fwRel="2"
    ...
</schema>
```

Schema Designer

The Schema Designer is a user-friendly interface for performing the following common schema editing tasks:

- Displaying existing schemas.
- Creating schema elements.
- Moving elements within a schema.
- Adding attribute values.

The designer provides two view modes:

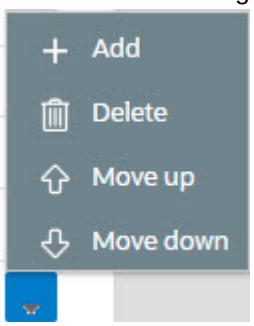
- **Text** mode shows the schema elements and their attributes written in the proper syntax and allows for direct text entry.
- **Tree** mode is a view showing the elements in a tree format with many of the key attributes of each element in tabular form. While in this mode, the **Node Display** controls allow the user to choose between displaying the elements by their internal identifiers or their associated screen labels while viewing or editing the schema.

> **Note**
>
> Schemas that are not owned by the current installation owner are protected in both view modes.

The following sections provide more information about functionality available in the Tree mode.

- If the schema definition refers to another schema using an 'include' statement, a triangle is visible to the left of the tree area. One can expand that schema by clicking the triangle.
- Detailed information about an element's definition can be displayed and if applicable, updated from the**Tree** view by clicking the edit icon for that element, which appears on the right.  Note that the element attributes are only editable if the element is defined in this schema. When viewing the attributes of an element that is part of an 'include' of another schema, the attributes are display only.
- Context-sensitive embedded help is provided for fields and controls in the edit pane by clicking the Help icon 

Copyright © 2026, Oracle and/or its affiliates.

- There is a menu dropdown icon visible to the left of the tree for any element that is defined in the current schema being viewed. It is not visible for elements included from another schema. New elements may be added in the **Tree** mode. Clicking that dropdown displays options to Add, Delete or Move an element. 

Note that not all options are visible, based on what is allowed for the element adjacent to the menu. For example, on the main "schema" node, only the Add option is visible as deleting or moving the "schema" node is not applicable.

When adding a new element, you are prompted first for the element position, which is either a sibling node to the current element or a child node of that element. For either option, the new element is added below the one adjacent to where the menu is clicked. You are then prompted for the element type.

The following lists the possible element types. Most are self explanatory and represent standard schema options.

- Characteristic List. This is a special type of Flattened List element that is used to map list elements that include a sequence and a characteristic value for a given characteristic type. This element is only applicable for maintenance objects that have one or more characteristic child tables where the primary key is the maintenance object's key, characteristic type and sequence. Effective dated characteristic collections are not supported. The user defines the list name, the characteristic value element name and the characteristic type. The system will configure the remaining flattening information accordingly.
- Characteristic. This is a special type of Flattened Field element that is used to map a single element to the characteristic for a given characteristic type. This element is only applicable for maintenance objects that have one or more characteristic child tables where the primary key is the maintenance object's key, characteristic type and sequence. Effective dated characteristic collections are not supported. The user defines the element name and the characteristic type. The system will configure the remaining flattening information accordingly.
- Comment. This adds a comment to the schema.
- Embedded HTML. This is specific to a schema enabled for UI Hints. It is used to include a UI map fragment.
- Field
- Flattened Field
- Flattened List
- Group
- Include BO Schema
- Include BS Schema
- Include DA Schema
- Include Map Schema

Copyright © 2026, Oracle and/or its affiliates.

- Include SS Schema
- Input Map Title. This is specific to a schema enabled for UI Hints. It is used to define a title element for the map.
- List
- Nested Flattened Field. This is a flattened field from a child table.
- Raw Element. This element is used to capture text as is. It is typically used to capture an XML structure without any details of the definition of the individual nodes.
- Section. This is specific to a schema enabled for UI Hints. It is used to define a section within the map.
- Simple Field. This is a special type of Field element that is used to define an element that is mapped to a column that supports data defined in an XML structure. (This is either a column with the character large object data type (CLOB) or the XML data type).

The **Schema Designer** is available by choosing the **Schema** tab on the Business Object, Data Area, UI Map, Business Service, and Script pages.

Schema Viewer

The schema viewer shows a tree-view presentation of a schema in its expanded form.

> **Note**
>
> If the schema includes presentation instructions and attributes they are not included in this view. Refer to Understanding UI Hints for more information.

The schema illustrates the structure to be used when communicating with the schema's associated object. The following takes place when a schema is expanded:

- If the schema definition includes references to other schemas, these references are replaced with the corresponding schema definitions.
- Also, if the schema definition contains **private** elements, they are omitted from this view.

Clicking on any node on the tree populates the text box on the top with the node's absolute XPath expression. You will find this feature very useful when writing scripts interacting with schema-based objects. Scripting often involves referencing elements in a schema-based XML document using their absolute XPath expression. You can use this feature on the schema viewer to obtain the XPath expression for an element and copy it over to your script.

Business Event Log

Business Event Log may be viewed as a tool designed to capture any type of business event worth noting. You configure business objects to represent the various types of events your application calls for. The following type of details may be captured for each event:

- The business object representing the type of event.
- The date and time the event took place and who initiated it.
- The business entity for which this event is logged.
- Standard application message to describe the event.

Copyright © 2026, Oracle and/or its affiliates.

- Additional context information that is available at the time of the event and varies for each type of event. The Business Event Log maintenance object supports a standard characteristics collection as well as an XML storage (CLOB) field. The event's business object determines where each piece of information resides. Refer to Business Objects for more information.

One common type of event may be the audit of changes made to sensitive data, for example, tracking an address change. Whenever an entity associated with a business object is added, changed, or deleted the system summarizes the list of changes that took place in that transaction and hands them over to **Audit** business object algorithms to process. You may design such an algorithm to audit the changes as business event logs. Refer to a business object may define business rules for more information.

You can also allow users to initiate business event logs to capture important notes about a business entity by exposing a BPA Script to invoke the event's corresponding business object.

Bottom line is that any process can create a business event log by invoking the business object representing the appropriate type of event.

Miscellaneous Topics

The following sections describe miscellaneous system wide topics.

- Javadocs Viewer
- Module Configuration
- Global Context Overview
- System Data Naming Convention
- Accessibility Considerations
- Referencing URIs
- Caching Overview
- Expression Parser
- Debug Mode
- Prepare Issue Details
- System Override Date
- Additional Search Options

Javadocs Viewer

The system provides a Javadocs viewer and a Groovy Javadocs viewer, which are interactive tools that allows you to browse Java documentation files (Javadocs) for Java classes that execute on the application server. For information about how to use Javadocs and navigate within the viewer, refer to Javadocs help.

Javadocs

To launch the Javadocs viewer, open open **Admin**, and then **Implementation Tools**, and then **Javadocs**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Proprietary Java Classes. A small number of Java classes have been suppressed due to their proprietary nature.

Additionally, when viewing a record that references a Java program in the algorithm type page, the batch control page or the FK reference page, an icon adjacent to Java program allows you to view the Javadocs for that program.

Groovy Javadocs

The Groovy Javadocs viewer includes the Java classes that are accessible to Groovy code within scripts. Launch this viewer using the 'View Groovy Javadocs' action on the script page.

Module Configuration

The system provides the ability to simplify the user interface based on functionality areas practiced by your organization.

Menu items and other user interface elements are associated with function modules. By default, all function modules are accessible. If a function module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information on how to turn off a module.

If a function module is made non-accessible, i.e. turned off, its related elements are suppressed from the user interface. In addition the system may validate that related functionality is not accessed. This also means that turning off the wrong module may cause any of the following to occur:

- Menu items may not appear. Refer to Menu Item Suppression to better understand how menu item suppression works.
- Entire menus may not appear. Refer to Menu Suppression to better understand how menu suppression works.
- Tabs on pages may not appear.
- Fields may not appear.
- The system may return an error message when you attempt to use a function (indicating the function is turned off).

To correct the above situation, simply remove the module from the turned off list thus making it accessible again.

Your module configuration setup is displayed on the installations record.

- Menu Item Suppression
- Menu Suppression
- Turn Off A Function Module

Menu Item Suppression

The following points describe how your module configuration can suppress menu items.

- Menu items that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules. If your module configuration has

Copyright © 2026, Oracle and/or its affiliates.

turned off all of the menu item's modules, the menu item is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu item is not suppressed.

- If a menu line doesn't contain any accessible items, the menu line is suppressed.
- If all lines on the Admin Menu are suppressed, the menu itself is suppressed. This is not expected to ever occur for the Main menu as the user preferences is on that menu and the expectation is that all users have access to their preferences.
- Menu items that are suppressed are not visible in the Menu Item Search results.

Menu Suppression

In addition to the above Menu Item Suppression logic, the following points describe how your module configuration can suppress an entire menu.

- Menus that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules.
- If your module configuration has turned off all of the menu's modules, the entire menu is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu is not suppressed.

Turn Off A Function Module

The base package is provided with a **Module Configuration** Feature Configuration that allows your organization to turn off base package function modules.

To turn off any of the base package function modules add a **Turned Off** option to this feature configuration referencing that module. Refer to the `MODULE_FLG` lookup field for the complete list of the application's function modules.

Any module not referenced on this feature configuration is considered turned on, i.e. accessible. To turn on a module, simply remove its corresponding **Turned Off** option from this feature configuration.

You may view your module configuration setup on the installation options page.

> **Note**
>
> **Only one**. The system expects only one **Module Configuration** feature configuration to be defined.

Global Context Overview

The framework web application provides each product the ability to nominate certain fields to act as a "global context" within the web application. For example, in Oracle Utilities Customer Care and Billing, the global context fields include Account ID, Person ID and Premise ID. The values of these fields may be populated as a result of searching or displaying objects that use these fields in their keys. If you navigate to the Bill page and display a bill, the global context is refreshed with the Account ID associated with that bill. The global context for Person ID and Premise ID are refreshed with data associated with that account.

The fields designated as global context for the product are defined using the lookup `F1_UI_CTXT_FLDS_FLG`.

Copyright © 2026, Oracle and/or its affiliates.

Changing the values of the global context typically cause data displayed in zones on the sidebar to be refreshed to show information relevant to the current values of these global context fields.

When the value of one of the global context fields changes, an algorithm plugged into the installation record is responsible for populating the remaining global context values accordingly. Refer to your specific product for more information about the base algorithm that is provided for that product.

System Data Naming Convention

There are several maintenance objects in the system that include owner flag in one or more of its tables. We refer to the data in these tables as "system data". Some examples of system data tables include Algorithm Type, Batch Control, Business Object and Script. Implementations may introduce records to the same tables. The owner flag for records created by an implementation is set to **CM** (for customer modification), however the owner flag is not part of the primary key for any of the system data tables. As a result, the base product provides the following guidelines for defining the primary key in system data tables to avoid any naming conflict.

- Base Product System Data
- Implementation System Data

Base Product System Data

For any table that includes the owner flag, the base product will follow a naming convention for any new data that is owned by the base product. The primary key for records introduced by the product is prefixed with `xn-` where **xn** is the value of the owner flag. For example, if a new background process is introduced to the framework product, the batch code name is prefixed with `F1-`.

> **Note**
>
> There are some cases where the hyphen is not included. For example, portal codes omit the hyphen.

For most system data, the remainder of the primary key is all in capital case. An exception is schema oriented records. For business objects, business services, scripts, data areas and UI maps, the product follows the general rule of using CapitalCase after the product owner prefix. For example, `F1-AddToDoEntry` is the name of a base product business service.

> **Note**
>
> **Data Explorer Business Services.** For business services used to invoke a data explorer zone, it is recommended to name the Business Service the same name as the related zone rather than defining a different CapitalCase name for the business service.

Please note that this standard is followed for all new records introduced by the base product. However, there are base product entries in many of these system data tables that were introduced before the naming convention was adopted. That data does not follow the naming convention described above.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Schema naming conventions.** A context sensitive "Schema Tips" zone is associated with any page where a schema may be defined. The zone provides recommended naming conventions for elements within a schema along with a complete list of the XML nodes and attributes available to you when you construct a schema.

Implementation System Data

When new system data is introduced for your implementation you must consider the naming convention for the primary key. The product recommends prefixing records with **CM**, which is the value of the owner flag in your environment. This is consistent with the base product naming convention. This convention allows your implementation to use the CM packaging tool in the Software Development Kit as delivered. The extract file provided with the tool selects system data records with an owner flag of **CM** and with a **CM** prefix.

> **Note**
>
> If you choose not to follow the CM naming convention for your records and you want to use the CM packaging tool, your implementation must customize the extract file to define the appropriate selection criteria for the records to be included in the package. Refer to the Software Development Kit documentation for more information.

Also note that owner flag may be introduced to an existing table in a new release. When this happens, the CM packaging tool is also updated to include these new system data tables. Your implementation will have existing records in those tables that probably do not follow any naming convention. After an upgrade to such a release, if you want to include this data in the CM packaging tool, you must customize the extract file for the tables in question.

Accessibility Considerations

This section provides information about topics related to accessibility that should be considered when extending the system via configuration tools.

Color Contrast

The HTML color `red` (RGB code #FF0000) when used for text on a white background does not comply with the accessibility color contrast guidelines. When it is desired to use red for text for emphasis in a zone, map or in a text string, the product recommends the shade of red with the RGB code #E0292F.

Note that in addition, the product provides a special CSS class **textColorRedOnWhite** which may be used when HTML references are used for applying the red color. For example, UI map, zone help text and scripting.

Identifying Rows in Grids

When navigating between rows in a grid on the user interface, screen readers will read out the row number and the element name (or names) with an attribute of 'scope=row'. By default, the system assigns this attribute to the non-suppressed primary key element (or elements) or to the first element in the grid if no primary key is defined. This may not be the information that best identifies the row. For example, a sequence number may be less meaningful than the heading of the next element in the list, such as 'Parameter Name'. The product provides a UI

Copyright © 2026, Oracle and/or its affiliates.

hint to allow you to specify one or more elements that should be used as the row header for accessibility.

Refer to UI Hint syntax for more details.

Referencing URIs

There are some configuration objects that require a reference to a URI, including file path URIs. The following sections highlight some functionality supported by the product with respect to defining / accessing URIs.

> **Note**
>
> In order for the functionality described below to occur, specific APIs must be used by the underlying code related to the fields that capture or process the file path or URL information. If you find that there is a URI-related field that does not provide the functionality described here, please contact customer support.

> **Note**
>
> For schema elements in a business object that reference a Field with the **URI** data type or define the element with the **URI** data type configured in the schema will automatically use the appropriate API that validates the value that may reference a substitution variable or must be checked against the allowlist if applicable.

- Validation Against an Allowlist
- URI Substitution
- External File Storage

Validation Against an Allowlist

Based on a property setting, your implementation may be configured to define a allowlist of URIs. If this setting is enabled, the system will issue an error if the URI is not defined in the allowlist. Consult your system administrator to verify if this setting is enabled or not for your implementation.

> **NOT_SUPPORTED**
>
> Refer to URI Substitution for information about how defining substitution tokens for URIs using the properties file technique automatically adds the defined URI to the allowlist.

URI Substitution

The system supports the ability to define substitution variables (sometimes referred to as tokens) for both URL values and file path values that reference native file storage locations. For URLs, the system supports defining variables in a Substitution Variable properties file. For native file storage paths, there are two options: the Substitution Variable properties file or via a File Storage configuration extendable lookup. More details on both these options are found below.

Copyright © 2026, Oracle and/or its affiliates.

- Substitution variables properties file. A substitution variable for all or part of the URI definition may be configured in a properties file. This allows the system administrators to define the proper URI locations in a properties file whereas the configuration users only need to know the variable name. For example, when defining a location for an extract file in an extract batch job, instead of typing a file path of `h:\oracle\serverName\1.0.0.0\batch\extract\`, the batch user can enter `@FILE_EXTRACT@`, assuming there is an entry in the substitution variables file with a name of `FILE_EXTRACT`, and a value of `h:\oracle\serverName\1.0.0.0\batch\extract\`. Another example is that the batch user could enter `@BATCH_FILES@\extract\`, assuming that the URI variable for `BATCH_FILES` is defined as `h:\oracle\serverName\1.0.0.0\batch\`.

  > **Note**
  >
  > The product automatically populates the value of SPLOUTPUT in the properties file so that this may be used in URI configuration. In addition, the product may supply some pre-defined variable names for other common references. As part of this, the 'advanced' menu in the system installation steps may prompt for installers to define the values of these pre-defined variables, if desired. Installations may opt to define additional substitution variables for various URI references. Refer to the *System Administration Guide* for more information.

- File Storage extendable lookup. Specifically for file paths that reference the native file system, the system also supports the ability to define a path using the **File Storage Configuration** extendable lookup. The following points highlight the steps to take for this option.
  - Navigate to Extendable Lookup and search for the **File Storage Configuration** lookup.
  - Click **Add** to create a new entry. Define a lookup value name. This will be used when configuring a file path that uses this value. Choose the File Adapter value of **Native File Storage**. Enter the desired file path value. The values defined here may in turn refer to values defined in the Substitution Variables properties file. For example, the file path can reference @SPLOUTPUT@.
  - To reference this value in system configuration, use the syntax `file-storage://XXXX`, where XXXX is the extendable lookup value. Using the same example from above, if you define an extendable lookup value of CM-FileExtract with a file path of `h:\oracle\serverName\1.0.0.0\batch\extract\`, then when configuring the extract file path for an extract batch job, enter `file-storage://CM-FileExtract`. Another example is that if the extendable lookup value's file path is configured as `h:\oracle\serverName\1.0.0.0\batch\`, then the user configuring the file path on the batch extract can enter `file-storage://CM-FileExtract/extract`

When should you define substitution variables in the properties file and when you define them in the extendable lookup? The following points highlight differences between the options that may help this decision.

- Substitution variables for URLs are only supported via the properties file.
- The properties file can typically only be modified by a system administrator. If there are values that are set at installation time and don't change, then defining the values in the properties file may be beneficial.
- When defining additional environments, such as test environments or production, the values in the extendable lookup may be copied using CMA. Ideally the values are defined such that they are the same between various regions.

Copyright © 2026, Oracle and/or its affiliates.

- When attaching files while Sending Email, if the files are located in native file storage, the file path must be declared in an entry in the File Storage Configuration extendable lookup.

External File Storage

The system supports using Oracle Cloud Object Storage for managing files in cloud implementations and can be configured to read files from or write files to this external location. Refer to Oracle Cloud Object Storage documentation for more information about obtaining an account and defining appropriate file locations or "buckets".

Once your cloud storage information is defined, the following points highlight the configuration steps required in the application.

- Define a signature key ring. When the system tries to communicate with cloud object storage, it must provide a signature key so that cloud storage can confirm that the request is from a trusted source.
  - Navigate to Key Ring in add mode and select the **RSA Signature Key Pair** business object.
  - Define a key ring code, which will be used in the File Storage configuration (below) along with a description.
  - Once the key ring is added, click **Generate Key** to generate a private / public key pair.
  - Click View Public Key to launch a pop-up that displays the public key, allowing the user to copy the key.
- At this point, the user should navigate to cloud object storage and register the public key. Refer to Cloud Services Administration Guide for more information on this step. Once this is done, the key ring is now ready to be configured in file storage configuration.
- Next, confirm that the region where your object storage is defined is configured in the system and if not, define the value.
  - Navigate to Extendable Lookup and search for the **OCI Region** lookup. Review the list of regions already provided.
  - If your region is not listed, click the Add button. Be sure that the Region Key and the Region Identifier match the values defined in the OCI region documentation. The region key is a three digit identifier. For example, the one for Sydney region in Australia is SYD. The region identifier follows a pattern of a two digit realm id followed by the city name, followed by a single digit, all separated by hyphens. For example, the one for the Sydney region is "ap-sydney-1".
- Use a File Storage extendable lookup to define the location and connection information so that you may reference this location in system configuration. The following points highlight the steps to take for this option.
  - Navigate to Extendable Lookup and search for the **File Storage Configuration** lookup.
  - Click **Add** to create a new entry. Define a lookup value name. Note that the lookup value should not have a slash or backslash in its name. This will be used when configuring a file path that uses this value. Choose the File Adapter value of **Oracle Cloud Object Storage**. Provide the following information that identifies the cloud storage options: User, Tenancy, Compartment, Namespace and Region. For the Key Ring, choose the value defined above.
- To reference this value in system configuration, use the syntax `file-storage://XXXX/...`, where XXXX is the extendable lookup value and any additional path information that is appropriate. For example, if you define an extendable lookup value of

Copyright © 2026, Oracle and/or its affiliates.

`CM-CloudStorage` and you have a bucket defined in Cloud Storage for processUpload, when configuring the file path for the upload batch job, enter `file-storage://CM-CloudStorage/processUpload`.

- The product supports referencing virtual folders for a bucket. Simply add the virtual folder name(s) after the bucket name. For example, if there is a bucket called 'financials' and a virtual folder for 'paymentUpload', the file path may have this value: `file-storage://CM-CloudStorage/financials/paymentUpload`. For extract processes, if the file path references a virtual folder that does not exist for the referenced bucket, the virtual folder is created as part of the extract process.
- You may optionally use a Bucket Name Prefix on the file storage configuration to support defining different Bucket Names for different environments. For example, imagine you want to have a separate "processUpload" bucket for a development region, a test region and a production region. In the Cloud Storage configuration you can define separate buckets for "DEV_processUpload", "TEST_processUpload" and "PROD_processUpload". If you needed to define the full bucket on each file path reference on various batch jobs, migrating the batch job configuration from one environment to another would require a manual step to change all the bucket references. Instead, the system supports defining a Bucket Name Prefix, which if populated will be prepended to bucket reference before connecting to Cloud Storage. This means that the prefix is defined once and all the references throughout the system can keep the common part of the Bucket name (such as "processUpload"). Then only the File Storage configuration needs to differ between the different regions.
- If your implementation integrates with an external reporting tool that can write the report output to specific delivery channels (buckets) check the Reporting Configuration switch. Define one or more Bucket Names for the buckets that the reporting tool may use for delivery channel options.

> **Note**
>
> Refer to the Key Ring documentation for information about practicing key rotation.

Caching Overview

A great deal of information in the system changes infrequently. In order to avoid accessing the database every time this type of information is required by an end-user or a batch process, the system maintains a cache of static information on the web server and in the batch thread pool worker. These are referred to as the "application caches". Some examples of application caches include

- System messages
- Field label and other field information
- Security Information

The framework product provides many specific caches for commonly used (and infrequently changed) data. In addition, specific edge applications may introduce additional caches as appropriate.

Information may also be cached on each user's browser.

The following topics highlight information about refreshing the various caches.

Copyright © 2026, Oracle and/or its affiliates.

Server Cache

The server cache refers to data that is cached on the web server. An important use of this cache is for users' online access to the application. The caches aid in better performance while navigating throughout the system, allowing for data to be accessed from the cache rather than by always accessing the database. Besides user access to the web server cache, other functionality deployed to the web server uses caches in a similar way. For example, web services are deployed to the web server and access their own version of the cache.

The contents of the cache are cleared whenever the web server is restarted. This means that fresh values are retrieved from the database once users and web services start using the application again.

The product also supplies a flush command that one can issue in the browser's URL to immediately clear the contents of the cache. The command `flushAll.jsp` flushes every cache (with an exception noted below).

For example, assume the following:

- the web server and port on which you work is called `OU-Production:7500`
- you add a new record to a control table and you want it to be available on the appropriate transactions immediately

You would issue the following command in your browser's address bar: `http://OU-Production:7500/flushAll.jsp`. Notice that the command replaces the typical cis.jsp that appears after the port number.

If your system has been configured correctly, the **flushAll** command will submit a request to do a "global" flush of caches (including the web services cache and the thread pool worker cache). This functionality uses a JMS Topic to publish the flush request. Refer to the *Server Administration Guide* for details on how to configure the JMS topic.

> **Important**
>
> For a cloud / production environment, the flushAll command will not flush the script cache. This is because flushing the script cache requires an entire recompilation of all the scripts. You don't want to inadvertently cause script recompilation in production when a script hasn't been changed. When a script has actually been updated in production, the script cache can be explicitly flushed by adding "?cacheNames=ScriptCache" to the URL. For example: `http://OU-Production:7500/flushAll.jsp?cacheNames=ScriptCache`

Also note that the system provides a business service `F1-FlushAllService`, which is equivalent to the flushAll command (except that it does not restart your browser session). This may be useful for any processes that are built to orchestrate updates to configuration data where you want to ensure that the cache is immediately flushed. This business service is also exposed as a REST web service.

> **Note**
>
> This business service does not flush the script cache in a cloud production environment. You need to use the browser command noted above to flush the script cache in this case.

Copyright © 2026, Oracle and/or its affiliates.

Batch Cache

When submitting a batch job, the batch component uses a Hibernate data cache to cache administrative data that doesn't change very often. The tables whose records are included in this cache are configured using the Caching Regime value of **Cached for Batch**. This is often referred to as the "L2 cache". Refer to Tables for more information. When starting a thread pool worker, data in tables marked as cached is loaded and cached for as long as that thread pool is running.

In addition batch jobs may also access application caches when applicable. When starting a thread pool worker, application data that is cached is loaded and cached for as long as that thread pool is running.

If there are any batch jobs that update any tables that are marked as **Cached for Batch**, you must run that batch job using a special thread pool that is configured with the caching disabled. Contact your system administrator to confirm the thread pool that is configured this way.

If there is a change in cached data that should be available for the next batch job, the following points highlight how the cache can be refreshed:

- By default the system is configured to automatically refresh the Hibernate cache every 60 seconds. However, an implementation may override the configuration to either change the number of seconds between intervals or to disable the automatic caching altogether. Application caches used by the batch jobs are not impacted by this refresh.
- Restart the thread pool workers.
- Run the `F1-FLUSH` (Flush all Caches) background process. This background process will flush the application data cached for all thread pool workers for all thread pools.
- If your the region has configured the thread pool workers to "listen" to requests for global flush as described in the Server Cache section, the thread pool worker caches are also refreshed when a **flushAll** command is issued.

> **Note**
>
> The Flush all Caches background process does not flush the script cache in a cloud production environment. You need to use the browser command noted above to flush the script cache in this case.

Client Cache

In addition to the web server's cache, information is also cached on each user's browser. After clearing the cache that's maintained on the web server, you must also clear the cache that's maintained on your client's browser. To do this, follow the following steps (note that depending on the browser you use, the steps may vary slightly):

- Select **Tools** on your browser's menu bar
- Select **Internet Options...** on the menu that appears.
- Click the **Delete Files** button on the pop-up that appears.
- Turn on **Delete all offline content** on the subsequent pop-up that appears and then click **OK**.
- And then enter the standard URL to re-invoke the system.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Automatic refresh of the browser's cache.** Each user's cache is automatically refreshed based on the **maxAge** parameter defined in the web.xml document on your web server. We recommend that you set this parameter to `1` second on development / test environments and `28800` seconds (8 hours) on production environments. Please speak to system support if you need to change this value.

Expression Parser

The product provides support for defining expressions that may be of a mathematical or logical/boolean nature. The expression may include variables and functions.

The data explorer column parameter is an example of where this may be used. That parameter supports the definition of a formula. Edge applications may include support for a formula or expression using this parser as well. For example, several application include a type of 'rule' object (calculation rule, form rule or usage rule) that is used for validation or calculation that may support applying a formula.

The following tables highlight what is supported in the expressions that use this parser.

Category	Supported in Expression	Description
Data types	Number	
	String	
	Boolean	
	List	
Literals	Numbers	
	Strings surrounded with either single quote or double quote. **Note** 'Escaping' special characters is not currently supported.	
	Boolean values: **true** and **false**.	
Operations	+	Plus
	—	Minus
	/	Division
	*	Multiplication
	^ or **	Power
	%	Modulus
Logical operations	=	Equal
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to

Copyright © 2026, Oracle and/or its affiliates.

Category	Supported in Expression	Description
	!= or <>	Not equal to

This table identifies the functions that are supported. Note that several of the functions are applicable to a list of values. Note that although the functions are listed in lower case, the column parameter syntax in data explorer indicates referencing the functions as all capital letters. The system converts the data explorer column formula to lowercase before being evaluated.

Function	Parameter	Results	Comments
`size( )`	List element	Number of elements in the list.	
`isEmpty( )`	List element	Returns true if the list is empty.	
`sum( )`	List element of type 'number'	Returns the sum of the numbers in the list.	
`avg( )`	List element of type 'number'	Returns the average of the numbers in the list.	
	One or more numbers separated by commas	Returns the average of the number arguments.	
`max( )`	List element	Returns the largest value in the list.	
	One or more comparable elements.	Returns the largest value of the number arguments.	
`min( )`	List element	Returns the smallest value in the list.	
	One or more comparable elements.	Returns the smallest value of the number arguments.	
`abs( )`	Number	Returns the absolute value.	
`ceiling( )`	Number	Rounds the number to the ceiling.	
`exp10( )`	Number	Raises 10 to the number power.	
`acos( )`	Number	Returns the arc cosine of the number in radians.	The result will lose precision, as it uses the system's double float based functions.
`asin( )`	Number	Returns the arc sine of the number radians.	The result will lose precision, as it uses the system's double float based functions.
`atan( )`	Number	Returns the arc tangent of the number radians.	The result will lose precision, as it uses the system's double float based functions.
`cos( )`	Radian	Returns the cosine of the radian angle input.	The result will lose precision, as it uses the system's double float based functions.
`exp( )`	Number	Raises e to the number power.	The result will lose precision, as it uses the system's double float based functions.
`log10( )`	Number	Takes the log, base 10, of the number.	The result will lose precision, as it uses the system's double float based functions.

Copyright © 2026, Oracle and/or its affiliates.

Function	Parameter	Results	Comments
`log( )`	Number	Takes the natural log (base e) of the number.	The result will lose precision, as it uses the system's double float based functions.
`sin( )`	Radian	Returns the sine of the radian angle input.	The result will lose precision, as it uses the system's double float based functions.
`sqrt( )`	Number	Returns the square root of the number.	The result will lose precision, as it uses the system's double float based functions.
`tan( )`	Radian	Returns the tangent of the radian angle input.	The result will lose precision, as it uses the system's double float based functions.
`floor( )`	Number	Rounds the number to the floor.	
`round( )`	Number	Assumes a scale of 0. The default rounding mode of “round half up” is applied.	
	Number, Scale	The default rounding mode of “round half up” is applied.	
	Number, Scale, Mode	The mode must be set to one of the following: • `“ROUND_CEILING”` • `“ROUND_DOWN”` • `“ROUND_FLOOR”` • `“ROUND_HALF_DOWN”` • `“ROUND_HALF_UP”` • `“ROUND_HALF_EVEN”` • `“ROUND_UP”` • `“ROUND_UNNECESSARY”`	
`negate( )`	Number	Returns the negative value of the number.	Only available in data explorer.

The following are special functions supported in the application for a list of values. In each case, the syntax is *function* [*indexVariable* in *listName* | *expression using indexVariable*], where the *indexVariable* is chosen by the formula writer to represent each entry in the list and the expression used to evaluate each entry must reference that variable.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The syntax supported for a given use of the formula in a functional area is driven by that particular functional area. For example, in Oracle Public Sector Revenue Management, a formula is supported in the “conditional element validation” form rule. In that form rule all variables including lists are declared in the form rule using letters and the formulas in turn use these letters. In that scenario, the functions below would reference the declared variable letter as the “listName”. Other specific functional area that use this expression parser may support different syntax for referencing elements or lists.

Function	Description	Examples
`any [ ]`	This function returns the value **true** if any of the entries in list satisfies the expression.	The following returns **true** if any entry in the Balance list is greater than 0. `any [ i in list/Balance | i > 0 ]`
`all [ ]`	This function returns the value **true** if all of the entries in the list satisfy the expression.	The following returns **true** if all phone numbers are populated. `all [ i in list/phoneNumber | i != ' ' ]`
`collect [ ]`	This function returns a new list of elements from the referenced list where the value of each entry of the new list is the result of the expression applied to each original value.	The following returns a new list with the tax rate applied to each amount. `collect [ i in list/amount | i * taxRate ]`
`select [ ]`	This function returns a list of all the values of the original list that satisfy the Boolean expression.	The following returns a new list with only the amounts that are negative numbers. `select [ i in list/amount | i < 0 ]`
`reject [ ]`	This function returns a list of all the values of the original list that do not satisfy the Boolean expression.	The following returns a new list with only the amounts that are not negative numbers. `reject [ i in list/amount | i < 0 ]`

Debug Mode

Your implementation team can execute the system using a special mode when they are configuring the application. To enable this mode, enter `?debug=true` at the end of the URL that you use to access the application. For example, if the standard URL was `http://CD-Production:7500/cis.jsp`, you'd enter `http://CD-Production:7500/cis.jsp?debug=true` to enable configuration mode.

In addition, if you have security access to the application service `F1OUDBUG`, you can enable this mode without changing the URL via the Help Menu - Enable Debug option. If debug mode

Copyright © 2026, Oracle and/or its affiliates.

is enabled, the help menu entry shows a Disable Debug option, allowing you to turn it off without changing the URL.

When in this mode certain debugging oriented tools become available via an icon visible.  Clicking the icon displays a slide-out panel with options defined below.

The following are related to the advanced debugger:

- **Start Debug** starts a logging session. During this session the processing of various configuration tools objects are logged. For example, the log will show the data areas that are passed in at each step and the data areas returned after the step is processed.
- **Stop Debug** stops the logging session.
- **Show Trace** opens a window that contains the logging session. All of the steps are initially collapsed.
- **Clear Trace** clears your log file.

> **Note**
>
> **The buttons are secured.** An application service `F1CONFIGLOG` has been provided for this functionality to allow implementations to restrict user access to these buttons. An additional application service `F1DEBUG` is checked when the environment is not a 'development' environment and is only applicable on premises. This allows implementations to provide an additional layer of security for their production users. (For cloud customers, the advanced debugger is only available in a development environment).

The following are available to view log entries captured while using the system.

- **Show User Log** allows you to view your own log entries. The number of "tail" entries to view may be specified in the adjacent **Log Entries** field before clicking the button. Limiting the number of entries to view allows the user to quickly and easily see only the latest log entries without having to manually scroll to the end of the log.
- **Show Log** allows for advanced settings. It provides for a date / time range to allow a user to target the output for a specific period. In addition, administrative users may request to view logs of other users. The following are examples of when this capability is needed:
  - Batch and integration logs are captured by special user codes created for this type of processing. These users typically do not represent a human and therefore wouldn't access logs using "show user log". Instead someone else would be accessing these logs.
  - Customer admins would like to retrieve logs for an end user possibly for support reasons. An end user might not have technical skills to gather or assess the logs nor would they be given the privilege to access the logs for security reasons
- Checking the **Global Debug** indication starts various tracing options that are added to the user log.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Show User Log button is secured.** An application service `F1USERLOG` has been provided for this functionality to allow implementations to restrict user access to this button. Only users with access to the **Administrative** access mode of the application service are allowed to view logs of other users. Such restrictions may be called for in production environments.

> **Note**
>
> If your environment configures redaction rules, personal identifiable information may be redacted in the log output. Refer to Data Redaction for Export and Logging for more information.

For Cloud implementations only, the following buttons are available in this mode:

- **Show Access Logs** - Clicking this button launches a window allowing you to define a date / time range and the number of "tail" entries to view. Click 'Save' to view the access logs that satisfy the criteria.

  > **Note**
  >
  > **Show Access Log button is secured.** This button is secured using the application service `F1ACCLOG` and **Administrative** access mode.

- **Restart TPWs** - This button forces batch containers to terminate, interrupting all running batches. New containers will spin up and the interrupted batches resume. Refer to the Batch Best Practices technical reference paper for more information about this button and when to use it.
- **Restart CM Jar Servers** - This button is provided to aid with support in the cloud for custom Java code in certain use cases. Refer to the Cloud Services Implementation Guide for more information.

Other parts of the system may show additional configuration oriented icons when in this mode. For example, explorer zones may provide additional tools to assist in debugging zone configuration. These icons are described in the context of where they appear.

Also, in debug mode drop down lists in data explorer and UI map zones will contain the code for each item in addition to the item's display string.

Prepare Issue Details

When a user encounters an error with the product and submits a service request, information about the environment, the release and the configuration of the user is often very helpful in understanding the cause of the problem. The system provides a script to capture the environment details and issue description for reporting purposes.

To access the script use **Menu**, and then **Tools**, and then **Prepare Issue Details**. The option is also available in the help menu in the toolbar.

A pop-up window is provided to display the details that are being captured, including a generated Issue ID to uniquely identify the report. The pop-up allows the user to add additional information describing the specific issue. When the user presses Save, the details are added

Copyright © 2026, Oracle and/or its affiliates.

to the system log with the issue ID, which can be searched when investigating the problem. A message will be displayed to either confirm the details were successfully logged or indicate that an error occurred.

> **Note**
>
> **The script and menu item are secured.** An application service `F1-PREPISSUEDTLS` has been provided to allow implementations to restrict user access to this functionality.

System Override Date

The system provides a way to override the system date used for online operations. This feature is available if the server administrator has enabled it in the environment properties. For instructions on configuring environment properties see the *Server Administration Guide*. The system date override feature is not recommended for production environments.

Under the **General System Configuration** Feature Configuration, the **System Override Date Option Type** holds the date the application will use as the global system date instead of retrieving the same from the database. This feature can be especially useful in running tests that require the system date to be progressed over a period of time.

The system override date feature is also available at the user level. This is useful when a user wants override the system date to run tests without affecting the system date for other users in the environment. In order to override the system date for the user, open the User — Characteristics page, add the **System Override Date** characteristic type with a characteristic value set to the desired date in the YYYY-MM-DD format.

If system override dates are defined at both the feature configuration level and the user level, the date set at the user level will take precedence.

Additional Search Options

The product supports fuzzy searching in explorer zone types using the Oracle Text CONTAINS operator.

Refer to the DBA guide for details on setting up the database to support fuzzy searching. Note that there are some implementations where fuzzy searching will not be possible. For example, it's only available for implementations using the Oracle database. Additionally, not all languages are supported. Refer to the Oracle Database documentation for more information about fuzzy searching.

For information about the particular syntax to use in the explorer zones, refer to SQL Statement in the zone parameter details section.

Extensions Dashboard

This portal provides a high level summary of configuration entities that have been customized and extended to meet your business requirements. Included are customer owned entities and extended product owned entities. The portal is intended to be used by your implementation team to get an appreciation of areas that have been customized and may need to be reviewed or retested for various reasons.

Use **Admin**, and then **Implementation Tools**, and then **Extensions Dashboard** to navigate to the portal.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page

- **Extensions Summary.** This zone provides a summary of customized entities by maintenance object. You may select a specific maintenance objects to review entities of this type.
- **Extension Entities.** This zone lists customized entities of the selected maintenance object. It is only visible when a maintenance object is selected on the Extensions Summary zone.
- **Extension Details.** This zone provides additional information as to which specific tables contain the custom owned content for the selected entity. It is only visible when a specific customized entity is selected.
- **Revision Control.** This zone lists revisions made to the selected entity. It is only visible when revision information exists for the selected entity.
- **Configuration Migration.** This zone lists CMA migration objects that applied changes to the selected entity. It is only visible when import information exists for the selected entity.

The following zone may appear as part of the portal's **Review** tab page

- **Duplicate Configurations.** This zone highlights duplicate configuration issues that may rarely arise after an upgrade. Customers should use this information to adjust thir configuration accordingly.

To Do Lists

Certain events that occur within the system will trigger messages describing work that requires attention. For example, if a bill segment has an error, the system generates a To Do message to alert the person responsible for correcting such errors.

Each type of message represents a To Do list. For example, there are To Do lists for bill segment errors, payment errors, customer contact reminder, etc.

We refer to each message as a **To Do Entry**. Each To Do entry is assigned a specific **To Do Role**. The role defines the users who may work on the entry. A To Do entry has a **To Do log** that maintains record of the progress on the To Do entry. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed.

NOT_SUPPORTED

Refer to To Do Processing for a description of end-user queries and tools assisting in reviewing, assigning and processing To Do entries.

- The Big Picture of To Do Lists
- Setting Up To Do Options

The Big Picture of To Do Lists

The topics below provide more information about To Do configuration.

- To Do Entries Reference A To Do Type
- To Do Entries Reference A Role

Copyright © 2026, Oracle and/or its affiliates.

- The Lifecycle Of A To Do Entry
- How Are To Do Entries Created?
- Refining a To Do Entry's Details at Creation Time
- The Priority Of A To Do Entry
- What Happens When You Work A To Do Entry
- Monitoring A To Do Entry
- To Do Entries Have Logs
- Implementing Additional To Do Entry Business Rules
- To Do Entries May Be Routed Out Of The System
- To Do Information May Be Formatted By An Algorithm
- Periodically Purging To Do Entries

To Do Entries Reference A To Do Type

Every To Do entry references a To Do type. The To Do type controls the following functions:

- The page into which a user is taken when they drill down to the related object on an entry.
- The message associated with the To Do that appears on various pages. Note this message can be overridden for specific To Do messages by specifying a different message number in the process that creates the specific To Do entry. For example, the process that creates To Do entries associated with bill segments that are in error displays the error message rather than a generic "bill segment is in error" message.
- The To Do entry sort keys. Note that the processes that create To Do entries are responsible for populating the sort key values.
- Whether (and how) the To Do entry is downloaded to an external system (e.g., an email system).
- The roles to which an entry may be reassigned.
- The default priority of the To Do entry. Note that this value may be overridden by a **Calculate Priority** algorithm.
- An indication of whether a To Do of that type may be created manually by a user.
- The algorithms used to perform specific business rules for To Do entries of this type.
- The characteristics applicable to the To Do.

To Do Entries Reference A Role

Every To Do entry references a role. The role defines the users who may be assigned to **Open** entries.

The permissible roles that may be assigned to a To Do entry are defined on the entry's To Do type. After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type.

The base package determines the initial To Do Role as follows:

- The system checks if an entry's message category / number is rerouted to a specific role. If so, it defaults this role. Refer to Adjusting the To Do Role for more information.

Copyright © 2026, Oracle and/or its affiliates.

- There may be some other condition related to the object linked to your To Do entry that warrants an override role. A To Do pre-creation plug-in would be used to check the conditional logic and override the role, if applicable.
- If a Role wasn't determined in one of the previous steps and a Role is provided by the initiating process (algorithm or background process), the entry is created with that Role.
- If the entry does not have a role after the above takes place, the entry's To Do type's default role is assigned to the entry.

> **Note**
>
> At installation time, the system provides a default role assigned to the system To Do types when first installed called `F1_DFLT`. This is done to allow testing of the system prior to implementing of appropriate To Do roles for your organization. The recommendation is to configure all the To Do Types with appropriate business oriented To Do roles once they are defined.

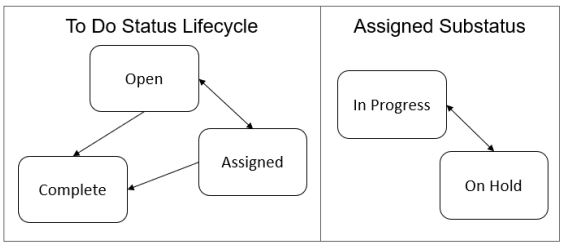
Supervisors for a Role

Most organizations have the notion of a supervisor who is responsible for all entries assigned to a given role. A user that is considered a supervisor should be a valid user in a role. In addition, application security is used to provide supervisors access to additional capabilities, namely the ability to review To Do entries assigned to other users and the ability to assign To Do entries to other users.

The pages listed in To Do Supervisor Functions are meant for supervisors only. So security for these pages should be restricted to those users that are considered supervisors. The To Do Management portal and the To Do Search page are available to all users. There is a special access mode for Supervisors to allow those users the ability to assign To Do entries to other users.

The Lifecycle Of A To Do Entry

The following state transition diagram will be useful in understanding the lifecycle of a To Do entry.



A To Do entry is typically created in the **Open** state. Entries of this type can be viewed by all users belonging to the entry's role. Refer to How Are To Do Entries Created? for information about how entries are created.

An **Open** entry becomes **Assigned** when it is assigned to a specific user or when a user proactively assumes responsibility for the entry. While an entry is **Assigned**, it is visible on the To Do Summary page only by the user who is assigned to it.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **To Do entries may be created in the Assigned** state. Some To Do background processes may create To Do entries in the **Assigned** state. When a user adds a To Do entry online and assigns the To Do to a user (as opposed to a role), the To Do entry is also created in the **Assigned** state.

An **Assigned** entry may be forwarded to a different user or role. If the entry is forwarded to a role, it becomes **Open** again. This may also be referred to as "reopening" or "unassigning" a To Do entry.

While a To Do Entry is in the **Assigned** state, if your implementation has configured it, you may further indicate the substatus of **In Progress** or **On Hold**.

- When you are ready to start work on an entry you can put it in the **In Progress** state. Refer to Working On A To Do for more information on how to do this.
- If you are not able to complete the To Do entry, you can put it **On Hold**, or stop working on it to put it back to the **Assigned** state (basically resetting the substatus). In either case, the To Do entry is still assigned to you.
- You may only have one To Do entry in the **In Progress** state. If you already have an entry in this state and you perform any action that puts a new To Do entry in **In Progress**, the other To Do entry will automatically be put into the **On Hold** state.

> **Important**
>
> To configure the system to allow users to use the substatus capability, you simply need to give users security for the Start Progress and Hold Progress and Stop Work access modes on the To Do Entry application service (**CILQTDEP**). The recommendation is that an implementation either grants all To Do users access to these access modes or grants no users access to these access modes.

> **Note**
>
> The substatus values are captured in a separate field (`TD_SUB_STATUS_FLG`). They are only populated if the main status field (`ENTRY_STATUS_FLG`) is in the Assigned status (`W`). But most pages in the system treat these substatus values as another status value. You typically won't see both values Assigned and In Progress or Assigned and On Hold. You'll just see In Progress and On Hold.

When an entry becomes **Complete**, it is no longer visible in various To Do queries (but it remains on the database for audit purposes). There are several ways an entry can become **Complete**:

- A user can manually indicate it is **Complete**.
- There are some background processes provided that create To Do entries based on the condition of some object. When subsequently running the same background process, the system automatically marks the To Do entry **Complete** when the object is no longer in the given condition. For example, an entry that's created when an account doesn't have a bill cycle is completed when the account has a bill cycle.

Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> The automatic completion of To Do entries occurs when the background process responsible for creating entries of a given type is executed. Therefore, depending on how often that process is run, there may be a delay where a To Do entry remains in a non-complete state even if the reported condition no longer exists.

- Algorithms are provided for lifecycle based objects that can create a To Do entry when entering or exiting a status. And corresponding algorithms may be used to complete the To Do entry when entering or exiting a status.

> **Note**
>
> The base product provided algorithm to complete a To Do Entry when entering or exiting a status includes the ability to configure that To Do entries of a certain type should not be automatically completed. These algorithms rely on a characteristic type of F1-EXLTD configured on the To Do Types that should not be auto-completed.

- The To Do monitor process may be used to evaluate whether or not the To Do entry can be closed automatically. Refer to Monitoring a To Do Entry for more information.

How Are To Do Entries Created?

A To Do Entry may be created in the following ways:

- A background process can create To Do Entries.
- An algorithm can create entries of a given type. Because the use of algorithms is entirely dependent on how you configure the control tables, the number of types of such entries is indeterminate.
- A user can create entries of To Do types that have a **Manual** usage. Refer to To Do Entries Created Manually for information about setting up manual To Do types.

For any base product process that includes logic to create a To Do entry, the system supplies a sample To Do type that may be used. Although the To Do types provided by the product are system data, the following information related to each To Do type may be customized for an implementation and is not overwritten during an upgrade:

- The creation process. If the To Do is created by a background process where the background process is referenced on a To Do type. Refer to To Do Entries Created By Background Processes for more information.
- The routing process. Refer to To Do Entries May Be Routed Out of the System for more information.
- The priority. Refer to To Do Type for more information.
- The roles that may be associated with the To Do type. Refer to To Do Entries Reference a Role for more information.
- The message override information. Refer to Suppress the Creation of the To Do Entry, Adjusting the To Do Role and What Happens When You Work A To Do Entry for more information.
- Suppress the Creation of the To Do Entry
- To Do Entries Created By Background Processes

Copyright © 2026, Oracle and/or its affiliates.

- To Do Entries Created By Algorithms
- To Do Entries Created Manually

Suppress the Creation of the To Do Entry

The system provides the ability to indicate that a To Do Entry should not be created based on conditions. When the system detects that no To Do should be created, there is no error. The system simply returns to the caller with an indication that no To Do was created. This functionality only applies to automatic To Do Entries created by an algorithm or background process. It does not apply to manually created To Do Entries.

Suppress Based on To Do's Message

There are some types of To Do Entries that are generated with a message that is specific to a condition that was found for a record. For example, business errors found when trying to update a record through a system process will generate a To Do entry using the error message as the To Do Entry's message. There may be some use cases where the error reported is not something that a user needs to address and it may be beneficial to your organization to suppress the To Do Entry. Note that this is only appropriate if the message generated is for informational or warning purposes and doesn't require action or if you know that the error will be resolved by a subsequent step.

The configuration to suppress a To Do entry based on its message is on the To Do type's message overrides page.

> **Note**
>
> **Message Category / Message Number.** Every error in the system has a unique message category / number combination. Refer to The Big Picture of System Messages for more information about message categories and numbers.

We do not supply documentation of every possible message that can be handled by a given To Do type. The best way to build each To Do type's reroute list is during the pre-production period when testing the system. During this period, compile a list of the messages that warrant special behavior described here.

Note that if the To Do type is configured to reference a specific message category / number, it means all To Do entries have the same message. If your organization does not want To Do entries created for this To Do type, then rather than using this suppression technique, you should identify the background process or algorithm that generates the To Do entries of this type and change the configuration there such that it does not try to create the To Do in the first place.

Suppress Based on a Condition

There may be use cases where a To Do entry should be suppressed based on some conditional logic. For example, consider the billing process in Oracle Utilities Customer Care and Billing. There is typically a bill cycle schedule that allows for attempting to bill all the accounts for the bill cycle over a multi-day window. There are some errors that occur on day 1 of the window that may be resolved by day 2 or 3. An implementation may choose to suppress To Do entries for this condition if this is day 1 or day 2 of the bill cycle. If the error still occurs on day 3, then the To Do entry should be generated. This type of condition needs to be determined using an algorithm. The To Do pre-creation plug-ins support returning an indication to suppress the To Do entry. Refer to To Do Pre-creation Plug-ins for more information.

Copyright © 2026, Oracle and/or its affiliates.

To Do Entries Created By Background Processes

There are different types of To Do entries created by background processes:

- To Do entries created by dedicated To Do background processes
- To Do entries created for object-specific errors detected in certain background processes
- To Do entries created based on a specific condition
- Dedicated To Do Background Processes
- To Dos Created for Object-Specific Error Conditions
- To Dos Created by Background Processes for Specific Conditions

Dedicated To Do Background Processes

There are To Do entries that are created by system background processes whose main purpose is to create To Do entries based on a given condition. For these background processes, the To Do Type indicates the creation background process.

Note that the Plug-in Driven batch program cannot be used for this purpose. This program assumes that any To Do Type that references it as a creation process is using the link for the Object-Specific Error Condition use case. If you want to use the plug-in driven batch process to create To Do entries based on a given condition, then you need to follow the pattern described in To Dos Created by Background Processes for Specific Conditions.

> **Note**
>
> **If you don't schedule the background process, the entries will not be created!** The To Do entries of this type will only be created if you have scheduled the associated background process. Therefore, if you want the system to produce a given entry, schedule the background process.

To Dos Created for Object-Specific Error Conditions

A system background process may create a To Do entry when an error is detected during object-specific processing. This is applicable for processes that do not have built in error handling, for example where there is an explicit "error" state or where the record has an explicit "exception" record.

For these background processes, the To Do Type must reference the creation background process.

To have the system create To Do entries for some or all of the errors generated by one of these processes, you must do the following:

- If you want the system to generate To Do entries for errors detected by one of the background processes below, go to the appropriate To Do type and populate the creation background process.
- If you want the system to generate To Do entries for some errors for the process, but not for all errors, populate the creation background process and then proceed to the message overrides tab to suppress certain messages. Do this by indicating the message category and message number you want to suppress. Any error that is suppressed is written to the batch run .

Copyright © 2026, Oracle and/or its affiliates.

The functionality will only create a new To Do entry if there is not already an existing (non-complete) To Do for the same To Do type, drill key and message category / message number. It will also check for an existing To Do for a successfully processed record and complete that To Do.

If you do not populate the creation background process, the errors are written to the batch run .

> **Note**
>
> **Errors received while creating a To Do entry**. If the background process cannot successfully create a To Do entry to report an object-specific error, the error is written to the batch run along with the error encountered while attempting to create the To Do entry.

> **Note**
>
> **System errors are not included**. To Do entries are not created for a system error, for example an error related to validation of input parameter. These errors are written to the batch run . Refer to Processing Errors for more information.

To Dos Created by Background Processes for Specific Conditions

There are some system background processes that create a To Do entry when the process detects a specific condition that a user should investigate. For each background process, the To Do type is an input parameter to the process. The system provides To Do types for each base package background process that may create a To Do entry.

> **Note**
>
> **No Creation Process.** These To Do types do not need (and should not have) a **Creation Process** specified.

To Do Entries Created By Algorithms

There are To Do entries that are created by algorithm types supplied with the base package. The system supplies a To Do Type for each of these To Do entries that you may use.

If you want to take advantage of these types of entries for system algorithm types, you must do the following:

- Create an algorithm:
  - This algorithm must reference the appropriate Algorithm Type.
  - These algorithms have a parameter of the To Do Type to be created. You should specify the To Do Type indicated in the table.
- Plug the algorithm into the respective control table.

To Do Entries Created Manually

You must set up manual To Do entry types if you want your users to be able to create To Do entries online. Users may create a manual To Do entry as a reminder to themselves to

Copyright © 2026, Oracle and/or its affiliates.

complete a task. Online To Do entries may also be used like electronic help tickets in the system. For example, if a user is having a problem starting service, the user can create a To Do that describes the problem. The To Do can be assigned to a help resolution group that could either resolve the problem or send the To Do back to the initiating user with information describing how to resolve the problem.

If you want to take advantage of manual To Do entries, create a To Do type and specify the following information.

On the Main tab:

- Set the **To Do Type Usage** flag to **Manual**.
- Set the **Navigation Option** to **toDoEntryMaint** (To Do entry maintenance).
- Set the **Message Category** and **Message Number** to the message you want to be used for To Do entries of this type. The system will populate the message parameter with the Subject. To show only the subject in the To Do's message, use a message with "%1" as its text.

On the Roles tab:

- Specify the To Do roles that may be assigned to To Do entries of this type.
- Indicate the To Do role that should be defaulted when you create To Do entries of this type.

On the Sort Keys tab:

When a user adds a manual To Do entry, the system creates an entry with three sort key values. (Sort keys may be used on the To Do list page to sort the entries in a different order.) The To Do type should be set up to reference the sort keys as follows:

Sequence	Description
1	Created by user ID
2	Created by user name
3	Subject

We recommend that the keys have an **Ascending** sort order and that the Subject is made the default sort key.

> **Note**
>
> It is possible to define additional sort keys and use a To Do Post Processing algorithm to populate the values. In this case, the base sort keys defined above should still be defined.

On the Drill Keys tab:

When a user adds a manual To Do entry, it is created with a drill key value equal to the To Do entry's ID. When the user clicks the Go To button next to the message in the To Do list, the system uses the drill down application service (defined on the main tab) and the drill key to display the associated To Do entry.

The To Do type must be set up with a drill key that reference the To Do entry table and the To Do entry ID:

Sequence	Table	Field

Copyright © 2026, Oracle and/or its affiliates.

Refining a To Do Entry's Details at Creation Time

When a To Do Entry is created, the key information provided to the system are the To Do Type and the drill key, which is the identifier of the object related to the To Do entry. With this information, system logic along with algorithms called at creation time can add details to the To Do entry or update important attributes so that the To Do can be worked on by the right team in the right priority.

The topics in this section will highlight the features provided to further refine a To Do Entry at creation time.

Linking Additional Information to a To Do Entry

Additional information may be linked to a To Do entry using characteristics. This may include master data so that the To Do entry is associated with one or more specific main objects in the system. It may also include other information that is helpful to any user that may be working on the To Do entry.

For **manual** To Do entries, the valid characteristic types that may be linked to the To Do entry must be defined on the To Do Type for that To Do entry. When create a manual To Do entry, the user populates the related characteristics as part of creating the To Do.

For **automatic** To Do entries, the program that generates the To Do may link related data to the To Do using characteristics.

In addition, the system To Do Pre-creation plug-ins can be used to link additional characteristics. These algorithms would use the drill key, which is the related object for the To Do entry. Based on the drill key, it can look up other information and create appropriate characteristics. Your specific product may be providing one or more algorithms to determine key master data records associated with the related object. For example, Oracle Utilities Customer Care and Billing provides an algorithm that attempts to link a related person, account, premise, service agreement and service point for that drill key. Other edge applications provide a similar algorithm for their relevant entities.

Any information that your user may need to have to help work a To Do entry should be linked to the To Do via a characteristic. For example, if the Rate Class of an account is important or the amount of the previous bill will help determine the priority of the To Do entry, this information may need to be linked as a characteristic.

> **Note**
>
> With the volume of To Do entries in the system, it is recommended to carefully consider what information you are linking to the To Do as a characteristic. Users can drill down on details when investigating a To Do entry. Only very important and critical information that would help users in reviewing To Do entries in the To Do Management portal and make decisions about prioritization and assignments should be linked as characteristics.

Note, before you can set up these algorithms, you must define the characteristic types that you'll use to hold each of these entities. Also note that it is not necessary to define these characteristics as valid characteristic types on the To Do type.

Copyright © 2026, Oracle and/or its affiliates.

If your To Do entries reference characteristics that are related to your global context data, you may want to configure an alert algorithm to display an alert if a related entry is **Open** or **Assigned**. Refer to Installation Options for more information.

> **Note**
>
> Count of related To Do entries. There are some user interfaces that display a count of related To Do entries. This logic looks at foreign key characteristics linked to the To Do entry where the underlying object is a master or transaction object. It counts the distinct list of other non-closed To Do entries that reference the same characteristic(s).

Adjusting the To Do Role

When creating a To Do entry, for a To Do type, there may be attributes of the To Do entry that require that a special set of users work on it. This is sometimes referred to as 'rerouting the To Do entry' to a different role. This section highlights ways to override the role based on conditions specific to a To Do entry.

The information in this section only applies to automatic To Do Entries created by an algorithm or background process. It does not apply to manually created To Do Entries.

Refer to To Do Entries Reference A Role for the hierarchy used to determine the role .

Reroute Based on To Do's Message

As described in Suppress the Creation of the To Do Entry, To Do entries created for a given To Do type may have different messages based on the specific condition for a record. You may decide that based on the message, the To Do should be routed to a specific Role that is more knowledgeable about a specific error condition. The same message override configuration that may be used for suppression of a To Do entry, may alternatively be used to assign a specific Role. For example, if certain bill segment errors are always resolved by the rate specialist team, you can indicate the role for that team on the To Do type, message override.

Reroute Based on a Condition

The To Do Pre-creation plug-ins may be used to assign a specific role based on some condition related to the To Do entry's specific data.

The product provides configuration on access group to define To Do assignment overrides. The override list defines a To Do type and To Do role. This is useful if your product associates access groups with specific master data for row level security purposes and you would also like to direct To Do entries related to this data to a specific role.

Oracle Utilities Customer Care and Billing also defines similar To Do assignment override configuration on CIS division and Account Management Group.

A pre-creation algorithm is responsible for checking if data linked to the To Do entry is related to a configuration object that supports an assignment override and then looks for an override for the To Do type. You can decide if this algorithm should be plugged in at the To Do Type level or the Installation level. The decision is based on how many To Do types you define overrides for. If it's a small number, then it makes more sense to plug the algorithm in for each To Do type that you define an override for. This ensure that the configuration is only checked for To Do entries that will have an override. However, if the number of To Do types that are eligible for overrides is larger or if the list changing or growing often, it may be better to plug the algorithm in at the Installation level to ensure that every To Do is checked for a possible override.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note that your product may already provide a base delivered pre-creation algorithm that is checking for this override. If so, that can be used by your implementation. If not, your implementation can provide an appropriate algorithm.

To Do Pre-creation Plug-ins

The system provides two To Do Pre-creation plug-in spots that allow for functionality to add or update information on the To Do Entry that is about to be created. The two plug-in spots are To Do type and installation options. In general, the installation option To Do pre-creation algorithms would be used for logic that applies to many To Do types and the To Do type pre-creation algorithms would be used for logic that is specific to that To Do type.

If at least one To Do pre-creation algorithm is plugged in on the To Do type, only those algorithms are executed when creating the To Do entry and the installation option algorithms are ignored. The system provides a base delivered To Do type pre-creation algorithm that executes the Installation option To Do pre-creation algorithms. This allows you to control whether the installation algorithms should also be executed for a To Do type that has specific algorithms and if so, when.

The following points highlight functionality supported by both To Do pre-creation plug-in spots.

- Indicate that the To Do entry should be suppressed (not created). The expectation is that this type of algorithm is configured on the To Do type rather than at the installation options level, as this allows for more granular conditions to be checked for a given To Do entry. Refer to Suppress the Creation of the To Do Entry for more information.
- Linking context specific data to the To Do entry using characteristics. Refer to Linking Related Information to a To Do Entry for more information.
- Overriding the Role of a To Do entry based on specific configuration related to the To Do's characteristic data. Refer to Adjusting the To Do Role for more information.

Assigning a Priority

As described in The Priority of a To Do Entry, a To Do type may reference a Calculate Priority algorithm. If so, this algorithm is called as part of the steps to create a To Do entry. The algorithm may use the drill key or any of the related information linked to the To Do as described above to determine priority information.

The Priority Of A To Do Entry

Some To Do entries may be more urgent to resolve than others. A To Do entry is associated with a priority level representing its relative processing order compared to other entries.

Priority level is assigned as follows:

- If one or more **Calculate Priority** plug-ins are defined on the To Do entry's type, the system calls them to determine the entry's priority. They are called initially when a To Do entry is created and each time it gets updated. You may want to use this method if an entry's priority is based on context or time-based factors. For example, when priority takes into consideration account specific attributes. The system also provides a batch process (`F1-TDCLP`) that calls the calculate priority algorithms "at will" for non-closed To Do entries. This is useful when the priority should be reassessed periodically based on factors external to the To Do entry's information. Refer to To Do Type for more information on

Copyright © 2026, Oracle and/or its affiliates.

priority calculation algorithms. When the priority is determined by one of the algorithms, a log entry is created indicating that the priority was calculated.

- If a priority value has not been determined by a plug-in, the system defaults a To Do entry's initial priority to the value specified on its type.

A user may manually override a To Do entry's priority at any time. When a user overrides the priority, a log entry is created indicating that the priority was overridden. Notice that once a To Do entry's priority is overridden, **Calculate Priority** plug-ins are no longer called so as to not override the value explicitly set by the user.

> **Note**
>
> The system does not use priority values to control order of assignment nor processing of To Do entries. Priority is available to assist your organization with supporting a business practice that ensures higher priority issues are worked on first.

What Happens When You Work A To Do Entry

Many user interface tools allow the user to indicate that they will start working on a To Do entry. Refer to Working On A To Do for more details. When a user works on an entry, the user is transferred to the maintenance page associated with the entry (the To Do entry's drill key). The page to which the user is taken is configured on the To Do type.

It is also possible to configure the To Do type to launch a script when a user works on a To Do entry rather than taking the user to a specific page. The script can check conditions related to the To Do entry to determine a more appropriate page or could be configured to walk the user through the steps required to resolve the To Do entry.

A script is linked to a To Do type based on its message number using the To Do type's message overrides. Refer to Executing A Script When A To Do Is Selected for more information.

In certain pages, taking an action to work on a To Do entry automatically assigns the To Do entry to the user. For users in implementations that also take advantage of the assignment substatus values, the To Do entry is also updated to the **In Progress** substatus. Refer to The Lifecycle Of A To Do Entry for information about configuring the system to use the substatus values.

Monitoring A To Do Entry

A To Do type may reference a To Do monitor algorithm. The algorithms are executed by the To Do monitor background process (F1-TDMON). The process executes the monitor algorithms for any non-complete To Do entry whose To Do type references at least one monitor algorithm.

Monitor algorithms may be useful if there are business requirements where a To Do could be automatically completed based on criteria that the algorithm can review. For example, an algorithm may be used to detect if the situation that caused the To Do Entry to be generated has been remedied in the meantime, allowing for the To Do Entry to be completed.

To Do Entries Have Logs

Each To Do entry has a To Do log that maintains a record of the To Do's progress in the system. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed.

Copyright © 2026, Oracle and/or its affiliates.

Users can view the log to see who assigned them a particular To Do and whether any work has already been done on the To Do.

A log entry is created for all actions that can be taken on a To Do entry. Log entries are created for the following events:

- A To Do entry is created (either by the system or by a user)
- A To Do entry is completed (either by the system or by a user)
- A user takes an open To Do entry
- A supervisor assigns a To Do entry
- The To Do entry's status is updated and the event is not already covered by another log entry type
- A user forwards an entry to another user or role
- A user sends back a To Do to the user who forwarded it
- A user manually adds a log entry to record details about the To Do's progress
- A user manually overrides the To Do entry's priority
- The To Do entry's priority was updated as a result of a calculate priority algorithm.

> **NOT_SUPPORTED**
>
> For information about the contents of log entries for each of the events, refer to Log Entry Events.

Implementing Additional To Do Entry Business Rules

If your business practice calls for additional validation rules or processing steps to take place after a To Do Entry is created or updated, you may want to take advantage of the **To Do Post Processing** plug-ins defined on To Do type.

For example, you may want to validate that To Do entries are only assigned to users with the proper skill levels needed to resolve them. Refer to `F1-VAL-SKILL` for a sample algorithm handling such validation.

To Do Entries May Be Routed Out Of The System

A To Do type can be configured so that its entries are interfaced to another system.

For example, a given To Do type can be configured to create an Email message whenever a new To Do entry is created. The following points describe how to do this:

- Define the name of the background process responsible for interfacing the new To Do entries to another system on the respective To Do type. The base package contains a batch process called `F1-TDEER` that can be used for most situations. This batch process invokes the **External Routing** algorithms defined on each entry's To Do type.
- Plug in an appropriate **External Routing** algorithm on the respective To Do type. The logic in this type of algorithm performs the interface efforts for a specific To Do entry. For example, if an Email message should be created for a To Do entry, the logic in the algorithm would compose and send the Email message(s) for a specific To Do entry.

Copyright © 2026, Oracle and/or its affiliates.

To Do Information May Be Formatted By An Algorithm

A **To Do Information** algorithm may be plugged in on the installation record to format the standard To Do information that appears throughout the system. This algorithm may be further overridden by a corresponding plug-in on the To Do Type.

Periodically Purging To Do Entries

The To Do Entry is configured for Information Lifecycle Management . This is the recommended tool for managing older records.

A purging batch process is also supplied `F1-TDPG`. However, it has some limitations that should be considered. The purging process does not perform any referential integrity. As such, if you are proactively purging records that are not very old, they could be referenced by other objects that are still active.

The process purges **Completed** To Do entries. This background process offers you the following choices:

- You can purge all To Do entries older than a given number of days.
- You can purge To Do entries for a specific list of To Do types that are older than a given number of days.
- You can purge all To Do entries except for a specific list of To Do types that are older than a given number of days.

> **Note**
>
> **Different retention periods for different types of To Do entries.** Keep in mind that the purge program allows you to retain different types of entries for different periods of time.

Setting Up To Do Options

The topics in this section describe how to set up To Do management options.

- Installation Options
- Messages
- On Demand Assignment Configuration
- Defining To Do Roles
- Defining To Do Types
- Implementing The To Do Entries

Installation Options

There are a number of installation options that may be configured to govern various aspects of To Do processing.

- A **To Do Information** algorithm may be plugged in on the Installation record. Refer to To Do Entries My be Formatted By An Algorithm for more information.

Copyright © 2026, Oracle and/or its affiliates.

- A **To Do Pre-creation** algorithm may be plugged in on the Installation record to set additional information for a To Do entry before it is created. Refer to To Do Pre-creation Plug-ins for more information.
- A **Next To Do Assignment** algorithm must be plugged into the Installation record if your organization opts to use the on demand assignment feature.
- If your To Do entries reference characteristics related to your global context data and your product supports sidebar alerts generated by algorithms, you may want configure an algorithm to display an alert if an entry is **Open** or **Assigned** for the data currently in context. Refer to your product's documentation to determine if these types of alerts are supported.

Messages

You need only set up new messages if you use algorithms to create To Do entries or prefer different messages than those associated with the base package's To Do types.

On Demand Assignment Configuration

The base package provides configuration related to allowing your users to click the "Assign me a To Do" button on the Current To Do Zone on the sidebar. If your organization plans to use this feature, be sure that users are granted the **Get Next Assignment** the Current To Do Zone application service (FWLCTDEP).

Next Assignment Plug-in

In order for the user to get an assignment on demand, a Next Assignment algorithm needs to be plugged in on installation options. The algorithm is responsible for determining the next To Do entry the user should work on.

The product provides an algorithm out of the box (`F1-NEXT-TDNS`) that can assign work solely based on priority. Another algorithm (`F1-NEXT-TDSK`) may be used to only consider To Do entries the user has the proper skills to resolve. Your implementation may provide a different algorithm based on your own business rules.

Skills

To use the algorithm that attempts to match a user's skills to the To Do Type's skills, the product provides additional configuration that is needed.

- Define skills. You need to define one or more measurable skills so that you can assign a skill level to both the To Do Types and to your users. The algorithm expects you to use predefined characteristic types, one for each different type of skill needed. Define the characteristic values as numeric, where the higher the number, the higher the experience / skill level. The characteristic type should configure both the To Do Type and the User characteristic entities.
- Define message overrides, if applicable. If your To Do Type is used for exceptions, you may refine the skill needed for a given message. Do this by using an ad-hoc characteristic on the
- Update your To Do types with the characteristic types that indicate the skills needed and the minimum skill level for each.
- Skill requirements may also be overridden on a message level. The algorithm assumes an ad-hoc characteristic on To Do Type is used to capture skill override for a given message. The ad-hoc value format is as follows: msg=12,12011 skill=BILL-EXC level=1 where msg is the message number, skill is the characteristic type defining the type of skill and level is the minimum skill level.

Copyright © 2026, Oracle and/or its affiliates.

- Update your users that work on To Dos to indicate all their capable skills and skill levels. Create a characteristic entry for each skill defined and use the characteristic value to indicate the user's skill level. The algorithm provided will ensure that the user's skill is equal to or higher than the required skill for the To Do Type.
- Register the characteristic types you are using for skills so that the algorithm knows which ones to evaluate. Go to Feature Configuration and find (or create) the feature with the **To Do Options** feature configuration type. Define an entry with the option type **Skill** for each characteristic type defined.
- Register the characteristic type you are using to indicate the Override Skills for a message level override on the same feature option. Define an entry with the option type **Override Skill** for the characteristic type.

The base **To Do Post Processing** To Do Type algorithm `F1-VAL-SKILL` validates that a user has the necessary skill levels required to work on a given To Do entry.

Defining To Do Roles

This section describes the control table used to maintain To Do roles.

- To Do Role - Main
- To Do Role - To Do Types

To Do Role - Main

The **Main** page is used to define basic information about a To Do role.

To maintain this information, select **Admin**, and then **General**, and then **To Do Role**.

Description of Page

Enter a unique **To Do Role** and **Description** for the To Do role.

The grid contains the ID of each **User** that may view and work on entries assigned to this role. The First Name and Last Name associated with the user is displayed adjacent.

> **Note**
>
> **System Default Role**. The system supplies a default role `F1_DFLT` linked to each system To Do type. This is done so that To Do functionality may be tested prior to the creation of appropriate business oriented To Do roles.

In addition, various "type" objects or algorithms may reference a To Do role to use when creating a To Do for a given business scenario. This is dependent on your specific product.

To Do Role - To Do Types

The **To Do Types** page defines the To Do types that may be viewed and worked on by users belonging to a given To Do role.

To maintain this information, select **Admin**, and then **General**, and then **To Do Role** and navigate to the **To Do Types** page.

Description of Page

Enter the ID of each **To Do Type** whose entries may be viewed and worked on by the role.

Copyright © 2026, Oracle and/or its affiliates.

Use As Default is a display-only field that indicates if the role is assigned to newly created entries of this type. You may define the default role for a given To Do type on the To Do Type maintenance page.

> ⚠ **Caution**
>
> If you remove a To Do type where this role is the default, you must define a new role as the default for the To Do type. You do this on the To Do Type maintenance page.

Defining To Do Types

Use this page to define basic information about To Do types.

You can access the portal from the **Admin**, and then **General**, and then **To Do Type**. You are brought to a query portal with options for searching for a specific To Do Type. Once a To Do Type has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a To Do type, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **To Do Type.** This zone provides information about the selected To Do Type.
  - The **drill keys** used to navigate to the To Do entry's related page are defined on To Do Type.

    > ⚠ **Caution**
    >
    > Do not change this information unless you are positive that the process / algorithm that creates entries of a given type stores this information on the entries.

  - **Sort Keys** define attributes that may be used to sort To Do entries of this type. Note that this information is captured as a string and is meant for sorting purposes. The system does not capture data type information to be able to format the information for display. For example, dates are not going to display as per the user's display profile configuration.
    - The To Do Information zone on the To Do Management portal provides the ability to search for To Do entries using sort keys and includes up to five sort key values as columns in the results, which may be used for sorting as required.
    - The To Do list page includes an 'Extra Information' option to allow To Do entries to be sorted by a selected sort key. Note the default sort key for this type of sorting is also defined on To Do type.
  - Refer to To Do Entries Reference A Role for more information about **roles** and To Do entries.

Copyright © 2026, Oracle and/or its affiliates.

  - Refer to The Priority Of A To Do Entry for more information about **priority** related setup.
  - Refer to To Do Entries Created By Background Processes and To Do Entries May Be Routed Out Of The System for more information about associating a To Do Type with **batch processes**.

- **To Do Type Characteristics.** This zone lists characteristic types and values common to all To Do entries of this type.

The following zones may appear as part of the portal's **To Do Characteristics** tab page.

- **To Do Characteristics.** This zone is used to maintain the valid list of characteristics that can be defined for To Do entries of this type. The characteristic types for characteristics that are linked to the To Do entry as a result of a pre-creation algorithm do not need to be defined here.

The following zones may appear as part of the portal's **Override Messages** tab page.

- **Override Messages.** If you want To Do entries that reference a given message category / number to be routed to a specific To Do role or suppressed altogether or if you want to associate a script to a given message category / number, use this zone to specify these rules.
  - Refer to Suppress the Creation of the To Do Entry for more information on suppressing To Do creation.
  - Refer to To Do Entries Reference A Role and Adjusting the To Do Role for more information about overriding roles.
  - Refer to What Happens When You Work A To Do Entry for more information on user guidance.

The following zones may appear as part of the portal's **Algorithms** tab page.

- **Algorithms.** This zone is used to maintain the algorithms associated with the selected To Do Type.

Actions

The page supports the following special actions:

- **Create an Algorithm.** Use this action, on the Algorithms tab to creates a script based algorithm related to the selected To Do Type. You are prompted for information regarding the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the To Do Type. You may then navigate to the script to adjust it as needed.
- To Do Type Algorithms

To Do Type Algorithms

The following table describes all the system events that control important To Do functions. If you haven't already done so, you must set up the appropriate algorithms in your system.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Calculate Priority	Optional	Algorithms of this type may be used to calculate a To Do entry's priority. Refer to The Priority of a To Do Entry for more information on when this plug-in is called. Note that it is not the responsibility of the algorithms to actually update the To Do entry with the calculated priority value but rather only return the calculated value. The system carries out the update as necessary. If more than one algorithm is plugged-in the system calls them one by one until the first to return a calculated priority.
External Routing	Optional	Algorithms of this type may be used to route a To Do entry to an external system. The base package `F1-TDEER` background process invokes the algorithms for every To Do entry that its type references the process as the **Routing Process** and that the entry was not already routed. The background process marks an entry as routed by updating it with the batch control's current run number. If more than one algorithm is plugged-in the batch process calls them one by one until the first to indicate the To Do entry was routed.
To Do Information	Optional	We use the term "To Do information" to describe the basic information that appears throughout the system to describe a To Do entry. The data that appears in "To Do information" is constructed using this algorithm. Plug an algorithm into this spot to override the "To Do information" algorithm on installation options or the system default "To Do information" if no such algorithm is defined on installation options.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
To Do Monitor	Optional	Algorithms of this type are executed by the To Do Monitor background process and may be used to periodically review a To Do entry and perform actions, if needed. Refer to Monitoring a To Do Entry for more information.
To Do Post-Processing	Optional	Algorithms of this type may be used to validate and/or further process a To Do entry that has been added or updated.
To Do Pre-Creation	Optional	Algorithms of this type are called when a To Do entry is being added. Refer to Refining a To Do Entry's Details at Creation Time for more information. To Do Type pre-creation algorithms override the Installation level pre-creation algorithms.

Implementing The To Do Entries

To enable the To Do entries visible in the To Do Type page, you must configure the system as follows:

- Define the To Do roles associated with each To Do type and link the appropriate users to them. Once you have defined the roles appropriate for your organization's To Do types, remove the reference to this system default role `F1_DFLT`. Refer to To Do Entries Reference A Role for more information.
- For any To Do Type that is provided for a specific background process, the To Do simply needs to reference the appropriate Creation Background Process. When the background process is scheduled, To Dos are created based on the logic of the related background process. This applies to To Dos Created for Object-Specific Error Conditions and Dedicated To Do Background Processes.
- For any To Do Type that is provided for creation by an algorithm or other process, there may be configuration required to populate that To Do type as an algorithm parameter or as an attribute on a control table.

> **Note**
>
> Refer to the description of the To Do type for more information.

Background Processes

This chapter covers various topics related to background processes. Besides providing an overview of background process functionality, the various tools available within the application to define, submit and monitor background processes are covered.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Your specific source application may have additional background process topics. Please refer to the documentation section that applies to your source application for more information.

- Understanding Background Processes
- Defining Batch Controls
- Defining File Integration Types
- Defining File Integration Records
- On-Line Batch Submission
- Tracking Batch Processes
- Requests

Understanding Background Processes

This section describes various topics related to the background processes that perform many important functions throughout your product such as:

- Processing To Do Entries
- Monitor processes that select records in a given state to progress them to their next state in their lifecycle
- Processes that purge data
- Processes that extract data
- And many more...

- Background Processing Overview
- Batch Categories
- Parallel Background Processes
- Parameters Supplied To Background Processes
- Processing Errors
- Error Post-Processing Logic
- Post-Processing Logic
- Timed Batch Processes
- Monitor Background Processes
- Plug-in Driven Background Processes
- How to Re-extract Information
- How to Submit Batch Jobs
- How to Track Batch Jobs
- How to Restart Failed Jobs and Processes
- Assessing Level of Service
- Batch Resource Groups

Copyright © 2026, Oracle and/or its affiliates.

- Batch Control Options

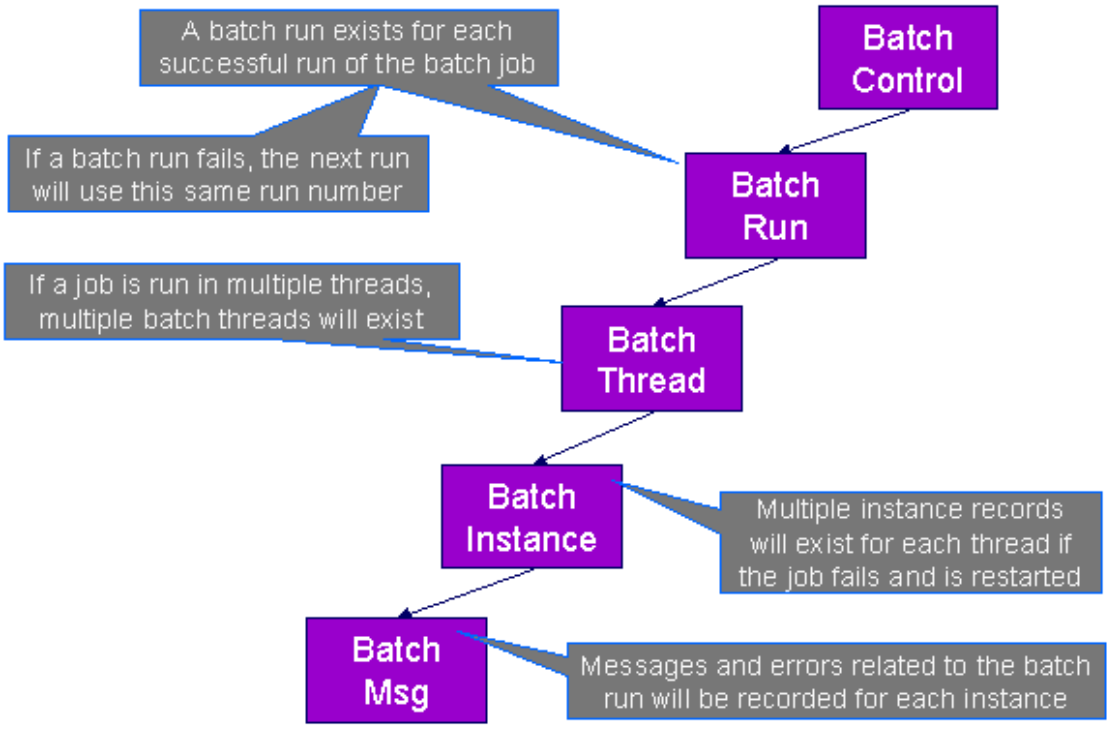
Background Processing Overview

While the system relies on a scheduler to secure and execute its background processes, there are additional issues that you should be familiar with:

- Batch control records are used for the following purposes:
  - Define the code that executes the logic associated with the background process.
  - For processes that extract information, the batch control record defines the next batch number to be assigned to new records that are eligible for extraction. For example, the batch control record associated with the process that routes To Do entries to an external system defines the next batch number to be assigned to new To Do entries that are configured with this batch control. When this To Do external routing process next runs, it selects all To Do entries marked with the current batch number (and increments the next batch number).
  - The batch control record for each background process organizes audit information about the historical execution of the background process. The system uses this information to control the restart of failed processes. You can use this information to view error messages associated with failed runs.
  - Many processes have been designed to run in parallel in order to speed execution. For example, the process that applies updates for a migration data set import for CMA can be executed so that multiple "threads" are processing a different subset of records (and multiple threads can execute at the same time). Batch control records associated with this type of process organize audit information about each thread in every execution. The system uses this information to control the restart of failed threads. Refer to Parallel Background Processes for more information.
  - Some processes define extra parameters. These parameters are defined with the batch control. Default values may also be captured for each parameter. They will be used when the background process is submitted on-line.

The following diagram illustrates the relationships that exist for batch control records.

Copyright © 2026, Oracle and/or its affiliates.



Results of each batch run can be viewed using the Batch Run portal.

Batch Categories

A batch process may be categorized for documentation purposes. The base values provided are as follows:

- **Ad Hoc**. Processes of this type are run on an ad hoc basis, only when needed. For example, if there is a process to do a mass cancel / correction of data, it would only be run when a situation occurs requiring this.
- **Extract**. Extract processes extract information that is interfaced out of the system. Processes of this type typically extract records marked with a given run number. If the requester of the process does not supply a specific run number, the system assumes that the latest run number should be extracted. If you need to re-extract an historical batch, you simply supply the respective run number when you request the batch process.
- **ILM**. Information Lifecycle Management jobs are crawler background processes that are associated with the ILM based storage solution.
- **Monitor**. Processes of this type are processes related to business objects with a lifecycle state that defines monitor algorithms. The monitor process selects records in a given state and executes its algorithms, which may cause the record to transition to another state or may trigger some other logic to occur. Using configuration, the monitor process may target only specific records. Refer to Monitoring Batch Processes for more information. Note that these types of background processes can be considered a subset of **Process What's Ready**
- **Process What's Ready**. Processes of this type create and update records that are "ready" for processing. The definition of "ready" differs for every process. For example, a payment upload process creates payments for every record that is **pending**. An overdue event monitor activates pending overdue events that have reached their trigger date.

Copyright © 2026, Oracle and/or its affiliates.

- **Purge**. Processes of this type are used to purge historical records from certain objects that generate a large number of entries and may become unwieldy over time.
- **To Do Entry**. Processes of this type are used to detect a given situation and create or complete a To Do Entry. Refer to To Do Entries Created by Background Processes for more information.
- **Upgrade**. Processes of this type are typically run only one time and are used to populate data in a table based on information that can be determined elsewhere in the system. These are processes that could have been done through an upgrade script, but perhaps due to data volume or timing, the upgrade logic warranted a separate stand-alone process.
- **Upload**. Processes of this type are used to upload data from a file into the system.
- The following categories are related to the data conversion / migration processes:
  - **Conversion**. Processes of this type are dedicated to converting or migrating data from external applications into the product.
  - **Object Validation**. Processes of this type are dedicated to validate data within objects for conversion or migration purposes.
  - **Referential Integrity**. Processes of this type are dedicated to validate referential integrity within objects for conversion or migration purposes.

> **Note**
>
> Additional categories may be introduced by your specific product.

Parallel Background Processes

Many processes have been designed to run in parallel in order to speed execution. This is referred to as running the process with multiple “threads”.

The Submission Details section on the Batch Control page indicates whether the process is multi-threaded or not.

The system provides the following strategies for distributing the data to the multiple threads.

- **Thread Level SQL Select** (THDS). This strategy is sometimes referred to as the “thread iterator” strategy. In this strategy, the batch job uses the primary key to figure out how to evenly distribute key ranges to each thread. Each thread is then responsible for selecting the records. In this strategy, the threads should also re-select the data periodically to release the cursor, which aids in performance. Note that this strategy is preferred but may only be used under the following conditions:
  - The data from only one maintenance object is being processed.
  - The primary key for the maintenance object is a single, numeric system generated key.

  > **Note**
  >
  > Parameters may be used to override the low and high id. Refer to Parameters Supplied to Background Processes for more information.
- **Key Ranges** (KEYS). This strategy is similar to the **Thread Level SQL Select** strategy, and follows the same maintenance object and prime key conditions, but uses actual key values to calculate thread ranges. Such strategy may be needed in situations where keys

Copyright © 2026, Oracle and/or its affiliates.

do not evenly distribute across threads, resulting in uneven thread completion times. For example, conversion entities in the staging schema have legacy keys, which are typically not evenly assigned. This impacts the Object Validation and XML resolution batch processes that thread by legacy keys.

> **Note**
>
> This strategy should not be widely used and as such is only supported by the Monitor and Ad-hoc Plug-in Driven batch programs.

- **Job Level SQL Select** (JOBS). This strategy is sometimes referred to as the “standard commit” strategy. In this strategy, the keys for the records to be processed by the batch job are all selected first and stored in a temporary table. The batch job then supplies each thread with a range of keys that it should process. This strategy is used if multiple maintenance objects are being processed by the batch job; if the primary key of the maintenance object has multiple parts or if the primary key is non-numeric.

The multi-threading logic relies on the fact that primary keys for master and transaction data are typically system generated random keys. In addition, if the data is partitioned, it is expected to be partitioned based on the primary key.

> **Note**
>
> The detailed description in the metadata for each batch control provided with the system should indicate if it may be run in parallel. Note that the strategy used is not typically indicated in the detailed description.

> **Note**
>
> **Overriding the thread ranges**. Your implementation has the ability to override the thread ranges if certain data in your system takes longer to process. For example, imagine you have a single account in Oracle Utilities Customer Care and Billing that has thousands of service agreements (maybe the account for a large corporation or a major city). You may want to set up the thread ranges to put this large account into its own thread and distribute the other accounts to the other threads. To do this, you should create the appropriate batch thread records ahead of time in a status of **Thread Ready** (`50`) with the key ranges pre-populated. Note that the base product does not provide the ability to add batch thread records online. If you are interested in more information about this technique, contact Customer Support.

- Optimal Thread Count

Optimal Thread Count

Running a background process in multiple threads is almost always faster than running it in a single thread. The trick is determining the number of threads that is optimal for each process.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> A good rule of thumb is to have one thread for every 100 MHz of application server CPU available. For example if you have four 450 MHz processors available on your application server, you can start with 18 threads to begin your testing: (450 * 4) / 100 = 18.

This is a rule of thumb because each process is different and is dependent on the data in your database. Also, your hardware configuration (i.e., number of processors, speed of your disk drives, speed of the network between the database server and the application server) has an impact on the optimal number of threads. Please follow these guidelines to determine the optimal number of threads for each background process:

- Execute the background process using the number of threads dictated by the rule of thumb (described above). During this execution, monitor the utilization percentage of your application server, database server and network traffic.
- If you find that your database server has hit 100% utilization, but your application server hasn't one of the following is probably occurring:
  - There may be a problematic SQL statement executing during the process. You must capture a database trace to identify the problem SQL.
  - It is also possible that your commit frequency may be too large. Commit frequency is a parameter supplied to every background process. If it is too large, the database's hold queues can start swapping. Refer to Parameters Supplied to Background Processes for more information about this parameter.
- It is normal if you find that your application server has hit 100% utilization but your database server has not. This is normal because, in general, all processes are CPU bound and not IO bound. At this point, you should decrease the number of threads until just under 100% of the application server utilization is achieved. And this will be the optimal number of threads required for this background process.
- If you find that your application server has not hit 100% utilization, you should increase the number of threads until you achieve just under 100% utilization on the application server. And remember, the application server should achieve 100% utilization before the database server reaches 100% utilization. If this proves not to be true, something is probably wrong with an SQL statement and you must capture an SQL trace to determine the culprit.

Another way to achieve similar results is to start out with a small number of threads and increase the number of threads until you have maximized throughput. The definition of "throughput" may differ for each process but can be generalized as a simple count of the records processed in the batch run. For example, in the Billing background process in Oracle Utilities Customer Care and Billing, throughput is the number of bills processed per minute. If you opt to use this method, we recommend you graph a curve of throughput vs. number of threads. The graph should display a curve that is steep at first but then flattens as more threads are added. Eventually adding more threads will cause the throughput to decline. Through this type of analysis you can determine the optimum number of threads to execute for any given process.

Parameters Supplied To Background Processes

This section describes the various types of parameters that are supplied to background processes.

Copyright © 2026, Oracle and/or its affiliates.

General Parameters

The following information is passed to every background process.

- **Batch code.** Batch code is the unique identifier of the background process.
- **Batch thread number.** Thread number is only used for background processes that can be run in multiple parallel threads. It contains the relative thread number of the process. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). Refer to Optimal Thread Count for Parallel Background Processes for more information.
- **Batch thread count.** Thread count is only used for background processes that can be run in multiple parallel threads. It contains the total number of parallel threads that have been scheduled. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. Refer to Optimal Thread Count for Parallel Background Processes for more information.
- **Batch rerun number.** Rerun number is only used for background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run).
- **Batch business date.** Business date is only used for background processes that use the current date in their processing. For example, a billing process may use the business date to determine which bill cycles should be downloaded. If this parameter is left blank, the system date is used. If supplied, this date must be in the format YYYY-MM-DD. Note: this parameter is only used during QA to test how processes behave over time.

  > **Note**
  >
  > There may be a slight lag in submission time for batch jobs submitted online. If the desired execution date/time is close to midnight, it is possible that your batch job will run on the day after you submit it. If you have left the business date blank in this case, keep in mind that your business date would be set to the day after you submit the job.

  > **Note**
  >
  > When rerunning a batch for a given run number, the system uses the batch business date of the original run.

- **Override maximum records between commits.** This parameter is optional and overrides each background process's Standard Commit. You would reduce this value, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources. You might want to increase this value when a background process is executed at night (or weekends) and you have a lot of memory on your servers.
- **Override maximum minutes between cursor re-initiation**. This parameter is optional and overrides each background process's Standard Cursor Re-Initiation Minutes. You would reduce these values, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources (or more frequent cursor initiations). You might want to increase these values when a background process is executed at night (or weekends) and you have a lot of memory on your servers.
- **User ID.** Please be aware of the following in respect of user ID:

Copyright © 2026, Oracle and/or its affiliates.

- Both the user submitting the job and the user ID recorded on the batch submission should have access to the application service for the batch control that secures execution.
- Any batch process that stamps a user ID on a record it creates or updates uses this user ID in applicable processing.
- This user ID's display profile controls how dates and currency values are formatted in messages.
- When submitting a batch job online, the batch user, language and email address are defaulted to the details of the user submitting the job. Only users that are granted the User Override access mode on the Batch Job Submission application service can override the batch user ID.
- If the submission user has user override privileges, they have the ability to retain the original batch user details when duplicating or rerunning a batch job from online submission. If the 'Run as original user' box is checked, the batch user, language and email address from the original batch run will be copied to the new batch job. If not, the user details are defaulted from the submission user.

- **Password.** Password is not currently used.
- **Language Code.** Language code is used to access language-specific control table values. For example, error messages are presented in this language code.
- **Trace Program Start**, **Trace Program Exit**, **Trace SQL** and **Trace Output**. These switches are only used during QA and benchmarking. If trace program start is checked, a message is displayed whenever a program is started. If trace program exit is checked, a message is displayed whenever a program is exited. If trace SQL is checked, a message is displayed whenever an SQL statement is executed. If trace output is checked, special messages formatted by the background process are written.

> **Note**
>
> The information displayed when the trace output switch is turned on depends on each background process. It is possible that a background process displays no special information for this switch.

Common Additional Parameters

Each batch control supports the definition of additional parameters. There are some additional parameters that are common to all batch processes or common to a specific type of batch process. The batch control should be delivered with the appropriate additional parameters. However, when new additional parameters are introduced, existing batch controls may not be updated with the new additional parameter.

The following table highlights the common parameters that may be linked to a batch control. Note that for batch parameters, although there is a sequence number that controls the displayed order of the parameter, the batch process does not use the sequence to identify a particular parameter but rather uses the parameter name. In some cases multiple parameter names are supported (a ‘camel case’ version and an ‘all caps’ version).

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Additional Comments
MAX-ERRORS / **maxErrors**	Each of the batch processes has, as part of its run parameters, a preset constant that determines how many errors that batch process may encounter before it is required to abort the run. A user can override that constant using this parameter.	The input value must be an integer that is greater than or equal to zero. The maximum valid value for this parameter is 999,999,999,999,999.
DIST-THD-POOL	Each batch process executes in a thread pool. This parameter is only necessary if the batch process should execute in a different thread pool than the default thread pool.	The default thread pool name is **DEFAULT**.
emailMode	When the batch job is submitted with an associated email address, the default logic is to send an email when the job completes regardless of success or failure. Use this parameter to limit the email based on the status of the job when it ends.	Valid Values • **ERROR** — send an email only when the job ends in Error status. • **SUCCESS** — send an email only when the job ends in successfully. • **ALL** — always send an email only when the job ends. (This is the default.)
The following parameters are only applicable to jobs that use the **Thread Level SQL Select** method of distributing work to threads as described in Parallel Background Processes.		
overrideLowIdValue	Specifies a new low id to use in calculating the range for a thread. The framework by default assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 9999999999), but this parameter will override the low value.	The parameter value can be an actual number or it can be set to **auto**. If **auto** is configured, it is set to the lowest current value on the database table associated with the background process.
overrideHighIdValue	Specifies a new high id to use in calculating the range for a thread. The framework by default assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 9999999999), but this parameter will override the high value.	The parameter value can be an actual number or it can be set to **auto**. If **auto** is configured, it is set to the highest current value on the database table associated with the background process.
idRangeOverrideClass	Use this parameter to specify a custom class to do thread range calculation. During batch execution, this override class is instantiated and the setter methods called to initialize the Ids as required. The low and high getter methods are called to retrieve the high and low ids to be used for the run.	The class name specified must implement interface com.splwg.base.api.batch.BatchIdRangeOverride.

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Additional Comments
batchStrategy	Use this parameter to override a THDS batch strategy with either JOBS - Job-Level or KEYS - Key Ranges strategy.	This parameter is only supported by the Monitor and Ad-hoc Plug-in Driven batch programs.
The following parameters are only applicable to jobs that perform a single commit, for example for extract batch jobs.		
numRecordsToFlush	This parameter defines how frequently to flush the Hibernate cache to prevent high heap consumption and Out Of Memory Errors.	

Specific Batch Parameters

Some background processes define additional parameters that are specific to their functionality. When a process receives additional parameters, they are defined and documented in the batch control entry in the application.

When submitting a batch job online, any default parameter values configured on the batch control record will be defaulted and may be overridden. Note that if the parameter value is configured to be Encrypted on the batch control configuration, the value will be shown encrypted.

- Indicating a File Path or File Name
- Encrypting and Signing Files
- Manifest File
- File Exchange through Oracle Cloud Storage

Indicating a File Path or File Name

Some of the system background processes use extra parameters to indicate a File Path and/or File Name for an input file or an output file. For example, most extract processes use File Path and File Name parameter to indicate where to place the output file.

For the file path parameter, refer to Referencing URIs for several points related to populating the file path and the supported syntax.

For the file name parameter, there are extract processes that support defining a file name that ends in ".zip" or ".gz", which will automatically result in writing the file with zip or gzip compression (based on the suffix). Similarly, there are upload processes that support decompression of files using the appropriate method based on the file name ending in ".zip" or ".gz". If the extract or upload process supports compression / decompression, it should be indicated accordingly in the description of the File Name parameter. If a batch control doesn't explicitly indicate that it supports compression or decompression in its description, the assumption should be that it is not supported.

In addition, there are extract and import processes that support automatic signing, encryption and decryption of the file content using PGP standards. File names ending in .pgp will be parsed to determine whether encryption, signing, compression or a combination of the three applies. If the file has a .pgp extension and requires decryption or signature verification, a run time error will be issued if no encryption or signing key ring parameters are supplied.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> File name / extension validation. The system ensures that the file name for any file being uploaded does not contain special characters and uses only standard UTF-8 supported characters. In addition, the system validates that the file's extension is referenced in the base delivered allow list F1-FileExtensionAllowList, defined as a managed content record.

> **Note**
>
> Custom file extensions. If your implementation has defined custom file extensions that are not found in the base product delivered allow list, you may create a custom managed content record to define your additional valid extensions. To do this, duplicate the F1-FileExtensionAllowList record and give it the name CM-FileExtensionAllowList. Change the description as desired. Remove the base delivered extensions and add your valid extensions using the <extension></extension> tags.

Encrypting and Signing Files

Some files created by data export related batch processes, may need to be encrypted and/or digitally signed for security purposes. Similarly, batch upload processes may need to verify the signature and/or decrypt the content of the file being imported.

New batch parameters control whether encryption or digital signing applies to files extracted or uploaded by the process. Both encryption and signing use Pretty Good Privacy (PGP) standards.

Processes that include a file name parameter, such as those using the plug-in driven extract template, support a file name extension of '.pgp'. Files ending with this extension are encrypted and/or signed using the encryption key parameters described in the next section.

Processes that have an internally constructed format for the file name, such as those using the generalized initial export template, perform encryption and/or signing according to whether the encryption parameters are supplied. An extension of '.pgp' is added to the constructed file name.

Extract Process Encryption Parameters

Parameter Name	Description	Additional Comments
`encryptExternalKey`	Use this key ring parameter to configure the external key provided by the recipient of the file.	If populated, the batch output file will be encrypted using a runtime generated key. The generated key will be encrypted using the active public key stored on the key ring. The recipient will use their private key to decrypt the generated key so the file may be decrypted. Only external public key rings may be used.

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Additional Comments
encryptSignatureKey	Use this key ring parameter to configure the internal key used to sign the file.	If populated, the batch output file will be signed using the active private key stored on the key ring. The assumption is that the corresponding public key has been shared with the recipient. Only internal file encrypting / signing key rings may be used.

Upload Process Encryption Parameters

Parameter Name	Description	Additional Comments
decryptInternalKey	Use this key ring parameter to configure the internal key to be used to decrypt the file to be uploaded.	If populated, the generated key received with the input file will be decrypted using the active private key stored on the key ring. The data is then decrypted using the generated key. Only internal file encrypting / signing key rings may be used.
decryptSignatureKey	Use this key ring parameter to configure the public signature key provided by the sender of the file.	If populated, the input file signature will be verified using the active public key stored on the key ring. Only external key rings may be used.

You may use the Batch Control Query portal to find all batch controls supporting these options.

Refer to File Encryption and Signing Keys for more information on maintaining internal and external encryption keys.

Manifest File

Batch processes that create files, for example data export related batch processes, may create one or more files in each processing thread. External applications that consume these files need to process them only when the entire batch process is successfully completed. To automate the consumption of such files, the external application may leverage a function to be triggered upon the creation of a manifest file to process all the files referenced by the manifest. The manifest file includes details such as content hash value, file size information etc that assist in confirming that file content is completed and has not been tampered with prior to processing the files.

The following sections provide additional information about manifest files.

The Manifest Option Batch Parameter

Manifest file creation is supported on certain base product extract related batch controls only. These batch controls include the **Manifest Option** batch parameter that controls whether a manifest file should be created or not upon successful completion of the batch process. You may use the Batch Control Query portal to find all batch controls supporting this option.

Copyright © 2026, Oracle and/or its affiliates.

Manifest File Information

Manifest file content is provided in JSON format and includes the following information:

- The version of the manifest details structure.
- The batch control code, run number and rerun number identifying the batch run producing the files
- Total number of files created by the batch run.
- Total number of records processed by the batch run.
- The manifest creation timestamp in ISO format.
- The previously completed batch run number, if any.
- The manifest file produced by the previously completed batch run, if any. This may be used to ensure files are processed in a proper order and assist in detecting potential situations of missing files.
- A list of files created by the batch run that includes the following details for each file: the file name, its size in bytes, and the SHA-256 hash value of the file's content.

The manifest file name is constructed as follows:

```
[batch control]_[batch run number]_[batch rerun number]_manifest.json
```

The following is an example of the manifest file structure.

```
{
  "version": 1,
  "batchCode": "HD-PDE",
  "batchRunNumber": 106,
  "batchRerunNumber": 0,
  "totalFiles": 3,
  "recordsProcessed": 10,
  "timeStamp": "2020-05-02T16:55:43.000353Z",
  "previousBatchRunNumber":"105"
  "previousManifestFileName":"HD-PDE_105_0_manifest.json",
  "fileDetails": [
    {
      "fileName": "SOME_FILE_NAME_001.json.gz",
      "fileSize": 1543,
      "hashAlgorithm": "SHA-256",
      "hashValue":
"ff7d4e466f93cadc4a18ab009080916baff631e03b728e2f2684c4a4f92ccbe1"
    },
    {
      "fileName": "SOME_FILE_NAME_002.json.gz",
      "fileSize": 1182,
      "hashAlgorithm": "SHA-256",
      "hashValue":
"78869e7ce971f28a70f5b35693a05726ba66109fecf492edb60291a62ae1ba9f"
    },
    {
      "fileName": "SOME_FILE_NAME_003.json.gz",
      "fileSize": 1182,
      "hashAlgorithm": "SHA-256",
```


Copyright © 2026, Oracle and/or its affiliates.

```
      "hashValue":
"78869e7ce971f28a70f5b35693a05726ba66109fecf492edb60291a62ae1ba9f"
    }
  ]
}{
```

Verifying The Manifest File

The **Verify Manifest File** web service may be used by an external system to validate that the manifest file it has received for a specific batch run is not corrupted or tampered with. The service returns the manifest file name and the hash value of its content for the requested batch run allowing the caller to compare their file details with the returned values.

File Exchange through Oracle Cloud Storage

Oracle Utilities Cloud Services can exchange files bidirectionally with Opower Cloud Services and Energy and Water Cloud Services through an Oracle Utilities Cloud Service-managed Object Storage bucket. Specific base product integrations may be configured to write files to a cloud service managed bucket or to read information from a file found in a cloud service managed bucket. (Integrations that support these use cases document the capability directly in the description of the batch control delivered for each use case.)

> **Note**
>
> This option is only available in a cloud installation.

Sharing Customer Exported Data

Some data export related batch processes support an option that streamlines the process by which customer extracted data may be routed to and shared with Oracle so it can be uploaded to Oracle's data lake for analytic purposes.

Implementations may choose to share data with Oracle in addition to exporting data for their own purposes. For this use case, you could configure the batch control parameters as follows:

- Define the file path / file location parameter with the appropriate configuration for extracting the data to your location. Refer to Indicating a File Path or File Name for more information.
- The batch controls that support this include the **Copy to Oracle** batch parameter. Use this parameter to indicate that you want to also share the data with Oracle.

You may have a use case where you only want to extract the data to Oracle and you do not need to write this data to any extract location for your own use. For this use case, configure the batch control parameter for file path / file location with the mnemonic %ORACLE%. This mnemonic signals to the program to extract the data to the configured Oracle location only. There is no need to populate anything in the Copy to Oracle parameter.

Processing Errors

When a background process detects an error, the error may or may not be related to a specific object that is being processed. For example, if the program finds an error during batch parameter validation, this error is not object-specific. However, if the program finds an error while processing a specific bill, this error is object-specific. The system reports errors in one of the following ways:

- Errors that are not object-specific are written to the error message log in the Batch Run.

Copyright © 2026, Oracle and/or its affiliates.

- Some batch processes create entries in an "exception table" for certain object-specific errors. For example, an error detected in the creation of a bill in Oracle Utilities Customer Care and Billing may be written to the bill exception table. If an error is written to an exception table, it does not appear in the batch run portal. For each exception table, there is an associated to do entry process that creates a To Do Entry for each error to allow a user to correct the problem on-line.
- For some background processes, errors that do not result in the creation of an exception record may instead generate a To Do entry directly. For these processes, if you wish the system to directly create a To Do entry, you must configure the To Do type appropriately. Refer to To Do entry for object-specific errors for information about configuring the To Do type. If the background process detects an object specific error and you have configured the system to create a To Do entry, the error is not written to the batch run. If you have configured your To Do type to not create To Do entries for certain errors, these errors are written to the Batch Run.

> **Note**
>
> **Some processes create exceptions and To Do entries**. It is possible for a background process to create entries in an exception table and create To Do entries directly, depending on the error. Consider batch billing in Oracle Utilities Customer Care and Billing; any conditions that cause a bill or bill segment to be created in **error** status result in a record added to the bill exception table or the bill segment exception table. However, any object-specific error that is not related to a specific bill or bill segment or any error that prevents a bill or bill segment from being created may result in a To Do entry for the object-specific error.

Error Post-Processing Logic

The product supports executing one or more algorithms when a batch process encounters an error that causes execution to stop. This allows for some special processing to occur to handle the failure of the batch job. Algorithms for this plug-in spot receive the batch control, batch run number, batch processing business date, number of threads and the list of the ad hoc parameters of the batch job.

> **Note**
>
> This plug-in spot is available for all Java based batch programs. For programs of type **Java (converted)**, the individual batch programs need to explicitly support this plug-in spot. Unless otherwise noted, assume that a **Java (converted)** program does not support it.

The following are some examples of functionality that may be executed when a batch job fails:

- Another object or record that is monitoring the batch job may have its status updated to reflect the batch status.
- An outbound message service may be invoked to perform a task related to the failure.

Note that the units of work for all threads are committed prior to executing the error post-processing logic.

Copyright © 2026, Oracle and/or its affiliates.

Post-Processing Logic

The product supports executing one or more algorithms after all the threads of a given batch job have completed. This allows for some special processing to occur at the end of a batch job. Algorithms for this plug-in spot receive the batch control, batch run number, batch processing business date, number of threads and the list of the ad hoc parameters of the batch job.

> **Note**
>
> This plug-in spot is available for all Java based batch programs. For programs of type **Java (converted)**, the individual batch programs need to explicitly support this plug-in spot. Unless otherwise noted, assume that a **Java (converted)** program does not support it.

The following are some examples of functionality that may be executed at the end of a batch job:

- Another dependent batch job can be kicked off. Note that this use case is only needed when the multiple dependent jobs are not part of a scheduler (which can also detect the successful end of one batch job so as to submit the next job).
- Statistics for the batch run may be analyzed and based on results, a To Do Entry may be sent to an administrator.
- If the current batch job is processing a large number of child records in multiple threads, a parent record could be updated to a different status or with some other audit information.

Note that the units of work for all threads are committed prior to executing the post-processing logic. The algorithm should perform standard error handling. If an error occurs in one of the post-processing algorithms, the overall batch job's status is set to Error so that it can be re-submitted to retry the logic in the finalize step.

Timed Batch Processes

Most batch jobs are submitted via a batch scheduler. In the absence of a scheduler, a batch control may be configured as "timed" triggering the framework to monitor and schedule these batch jobs as defined by the timer interval. The timer interval defines the desired interval between starts (in seconds). The system schedules new batch runs at each interval if the last instance of the job has completed.

When configuring a batch control as "timed", other default information must be provided, including the User ID and Language to use for submitting the job and the email address for notification, if desired.

Timed batch controls also include an Active setting, allowing for an implementation to temporarily stop further executions of the batch job (but retain the other timer settings).

Timed jobs are controlled by the default threadpool and not by a scheduler. When the **DEFAULT** threadpoolworker starts it will start executing any job for a Batch Control configured as **Timed** with the Timer Active set to **Yes**. This is whether the batch daemon or batch server is enabled or not.

Copyright © 2026, Oracle and/or its affiliates.

Monitor Background Processes

In many areas of the system, functionality is driven from business object configuration as a BO driven record progresses through its lifecycle. Refer to Business Object Lifecycle for details. As part of that functionality, it is possible that a background process, called a monitor batch process, is used to execute functionality for the record.

Monitor processes are designed to support parallel threads processing. When the number of records to process is unpredictable and can range from zero or few records to a many, submitting the batch with a fixed number of threads that fits the higher volumes means waste of threads with each run. Monitor processes are therefore designed to not create any batch run, and none of the thread level artifacts associated with a batch run, when there is no data to process. The batch job submission record is completed as usual but indicates that no batch run was needed.

> **Note**
>
> In an on-premise installation, monitor batch processes may be submitted via the command line, i.e. without creating a batch job submission record. In this situation, a monitor batch run will still be created even if there is no data to process because without a batch job record there is no way to communicate the reason for not creating a batch run.

The following sections further expand on concepts and features associated with monitor rule processing.

- Monitor Batch Template
- Deferred Requests Queue

Monitor Batch Template

A single batch program is provided for the BO monitor functionality. The product also provides a batch control template (`F1-MNTPL`) that should be used as a starting point. Parameters are used to limit the records processed by maintenance object and other optional parameters that may further limit the records. The product typically provides at least one monitor batch control for each maintenance object that supports a configurable lifecycle on its business object.

This topic highlights the parameters supported by the monitor batch job. Not all parameters are applicable to all maintenance objects and therefore may not be configured on a given base monitor batch control.

Parameter Name	Description	Comments
`maintenanceObject`	Maintenance Object	For most base delivered batch controls, this parameter is delivered already populated with the value of the maintenance object value. Note that it is supported to leave this value blank, at which point, the program will determine the maintenance object (objects) to process by looking for an MO that refers to this batch control record as an option.

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Comments
isRestrictedByBatchCode	Restrict by Batch Code	Set this to **true** to indicate whether the process should only select records that explicitly refer to this batch control on its current BO state. This is also referred to as "deferred" mode. If set to **false**, the program includes all records that refer to the current batch control in its BO state and records that don't refer to any batch control in its current state (but monitor algorithms exist in the current state). This is commonly referred to as "periodic" mode. Note that if the value is not set at all, the program will determine whether to run it as "deferred" or "periodic" based on whether the batch code is configured on the MO option as a **State Monitor Process** (“deferred”) or a **Periodic Monitor Process**.
restrictToType	Restrict by Related Type	This parameter is only applicable to maintenance objects that have a related 'type' object and the maintenance object has configured an option indicating the field for the related type column. This parameter may be used to limit the processing to records that are in the indicated type.
restrictToBusinessObject	Restrict by Business Object	This parameter may be used to limit the processing to records that are in the indicated business object.
restrictToBOStatus	Restrict by Status	This parameter may be used to limit the processing to records that are in the indicated status.
restrictToDate	Restrict By Date	Enter a valid date on the record's primary table to limit processing to records that are have a value in this field that is on or before the batch business date. Optionally, enter "+" or "-" followed by a number to shift the comparison date to the batch business date plus or minus the given number. For example, entering REVIEW_DT -3 will retrieve all records whose review date is on or before the batch business date minus 3 days.
sampleRecordNumber	Sample Record Number	This is not a commonly used parameter. It is only applicable when the monitor is used for a business use case that supports processing a subset of the records during a testing phase. For example, if the process is validating a large number of records, it may be an option to only validate every 100 records to determine if there are repeated validation errors that may indicate a common problem that may be solved to fix many errors.
minimumRecordsPerThread	Minimum Records Per Thread	Enter a number to force the batch framework to dynamically adjust the number of threads used by the batch program, ensuring that threads are not created for a small number of records. For example, • If the batch is submitted with 100 threads, the parameter is set to 50 records per thread, and there are 1M records to process, then there is no change to the number of allocated threads. • If the batch is submitted with 100 threads, the parameter is set to 50 records per thread, and there are 109 records to process, then the number of allocated threads is set to 3.

Also note that when submitting a monitor process with multiple parallel threads, the program will use a **Thread Level SQL Select** strategy unless any of the following are true (in which case it will use the **Job Level SQL Select** strategy:

Copyright © 2026, Oracle and/or its affiliates.

- The input maintenance object is left blank and the program finds more than one maintenance object that refers to this batch control in its options.
- A single MO is applicable but it has a multi-part primary key.
- A single MO is applicable and it has a single primary key, but it is a user defined key instead of a system generated key.
- The sample record number parameter is populated.

In the rare situation when the **Thread Level SQL Select** (THDS) strategy should be used but actual prime keys are not evenly distributed you may override the strategy to use actual key ranges by setting the **Batch Strategy** general batch parameter to **Key Ranges** (KEYS).

Deferred Requests Queue

There are use cases where an entity is set to a status that waits for a subsequent task to be performed asynchronously before it can progress in its lifecycle. Such task is implemented using monitor rules plugged into the entity's business object status definition along with a batch control. These monitor rules are not processed right away when the record enters that status but rather get deferred to when the associated batch runs next.

When the deferred task should be processed as soon as possible, a message can be added to the Deferred Requests Queue to perform the task near real-time. This mechanism offers a managed queue of messages, each representing a request to perform the deferred monitor rules of a given entity, similar to how these rules are processed in batch.

> **Note**
>
> This mechanism is only available in a cloud installation.

> **Caution**
>
> This feature should only be used to support business flows with time dependent events that require near real-time processing. All other asynchronous flows should remain deferred to batch processing.

The following sections describe concepts and features of the near real-time processing mechanism.

Requesting Near Real-time Processing

When an entity enters a status that is associated with time dependent deferred monitor rules, a request can be made to process these rules as soon as possible after the current transaction is committed.

The product supports the following options for requesting near real-time monitoring of an entity:

- To always send the entity for near real -time processing when it reaches a specific status, add the 'Add a Deferred Request' (`F1-DFREQADDN`) 'Enter' algorithm to the respective business object status definition. Refer to the algorithm detailed description for more information.
- You may use the **DeferredRequestAPI** function in a Groovy based algorithm to implement custom rules around when and how to add such request to the queue.

Copyright © 2026, Oracle and/or its affiliates.

- When applicable, the Generic Data Synchronization functionality also supports an option for near real-time processing that uses the Deferred Requests queue.

Exception Handling

When the deferred request is successfully processed by the real-time queue agent, the related entity is updated as per its business object rules. However, if the request keeps failing after several retries, the queue agent discards the request and the related entity stays in its current status, waiting to be picked up by the corresponding deferred monitor batch. At that point, if the error still persists the batch reports the issue as usual.

> **⚠ Warning**
>
> Do not remove the deferred batch control from the business object status definitions. These flows must remain deferred to batch as designed for data integrity reasons.

When applicable, sending entities to be processed near real-time allows the corresponding monitor batch to be scheduled less often as it no longer needs to meet near real-time frequency requirements. With most of the records successfully processed by the near real-time mechanism, the monitor batch is mainly used to handle exceptions.

Correlated Requests

By default, messages in the queue are processed in no particular order. In situations where multiple messages update the same resource there may be a need to process them serially to prevent concurrency errors.

When adding a message to the queue, the message can optionally reference the main entity it updates for the purpose of correlating it with other messages that update the same entity. Messages explicitly referencing the same main entity are processed serially.

Apply Polling Delay

When an entity's current status is associated with monitor rules that wait for a specific condition to be met in order to determine the next status the entity should transition to, it is not enough to perform these monitor rules once. The condition may not be met initially but could be met within a short period of time. In this situation, the deferred request placed in the queue could indicate that it requires polling delay.

When polling delay is requested, the deferred request message in the queue is processed as follows:

- When the message is added to the queue the current status of the related entity is noted.
- After successfully processing the message, if the related entity's status remains unchanged, the queue agent completes the current message and adds another message to the queue along with a short time delay, i.e. sets the entity to be processed again later.
- This process repeats itself for a limited number of attempts. With each attempt the delay in seconds increases.
- If the related entity's status remains unchanged after all attempts then the message is discarded with no exception. The related entity would be monitored as usual by the next scheduled batch run.

Copyright © 2026, Oracle and/or its affiliates.

Plug-in Driven Background Processes

Although the product is delivered with a rich library of background processes, implementations may have business requirements that require new processes to be introduced. It is possible for an implementation to write a background process from scratch using a base process as a template. However, the product also provides base background processes that call algorithms to do the work that is needed. These are called plug-in driven background processes. There are three major types of plug-in driven batch processes:

- Processes that act on records that are stored in the database in the system. These types of processes require SQL to select the records along with the logic to process the records.
- Processes that extract information to a file. These types of processes require SQL to select the records along with the logic to select the information and return it such that the process can write the information based on various settings.
- Processes that import data from a file and store new records in the system as a result. These types of processes require an algorithm that is able to map the data from the file to appropriate new records in the system

The subsequent sections provide more detail about the three types of plug-in driven background processes.

- Ad hoc Processes
- Extract Record Processing
- Uploading Records

Ad hoc Processes

A process that retrieves records in the system and does some action on them requires an algorithm to select the records to be processed and another algorithm to process the records. We refer to this type of process as an Ad hoc Process. The base batch control Plug-in Driven Generic Template (`F1-PDBG`) may be used as a template.

The base processes implement standard background process functionality including parallel background process logic and the ability to create To Do entries for errors. This allows for an implementation to take advantage of the pre-built support and provide plug-ins that include the logic that is unique to the specific use case.

The following sections provide more information about the functionality for ad hoc processes. Note that extract processes share some of the same functionality. Refer to Extract Record Processing for additional information specific to extract processes.

Select Records Algorithm

The first important algorithm to design when implementing a plug-in driven batch process is the Select Records algorithm, plugged in on the batch control page. This algorithm type must define the first parameter as the SQL. The batch job will directly access the SQL parameter value in the metadata (rather than invoking the algorithm). All other parameters are available for the algorithm to use for its own logic.

In addition, when invoking the algorithm, it must return the strategy to use (**Thread Level SQL Select** or **Job Level SQL Select**. Refer to parallel background processes for more information about the two strategies and when to use each. When choosing the **Thread Level SQL Select** strategy, the algorithm should return the name of the primary key in the Key Field parameter. In addition, the SQL should include a **BETWEEN** clause that includes the bind variables for the low and high ID for the ranges. See below for the bind variable syntax.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In the rare situation when the **Thread Level SQL Select** (THDS) strategy should be used but actual prime keys are not evenly distributed the strategy may be overridden at batch submission time to use actual key ranges by setting the **Batch Strategy** general batch parameter to **Key Ranges** (KEYS).

If the SQL statement includes variables that are determined at execution time, it must use bind parameters. Bind parameters are referenced in the SQL statement using a colon and a parameter name (for example `:parameter`). There are some variables provided by the system that are populated by the batch job at execution time. These have `f1_` as its prefix.

The system supports the following pre-defined bind parameters:

- **:f1_lowID** and **:f1_highID** - these should be used in the **BETWEEN** clause for the **Thread Level SQL Select** strategy. They should be selecting data from the Key Field that is returned as part of the Select Records algorithm. The batch job will substitute the appropriate ID range as required.

  > **Note**
  >
  > The table where this key field is selected from should be the first table in the FROM clause.

- **:f1_batchCode** and **:f1_batchNumber** - these are common attributes of the batch control that are referenced on a record for selection purposes. Note that the batch run number is set according to whether the batch job is a re-run of a previous run or not.
- **:f1_businessDate** - the batch job will populate the input batch business date, if populated otherwise the current date.

> **Note**
>
> The system supports both "f1." and "f1_" as a prefix for the bind variables. However, the "f1." prefix will result in an error if the SQL security property setting is turned on. As such, the underscore syntax is recommended. The period syntax remains for backwards compatibility.

For any other custom parameters, the Select Records algorithm may return one or more sets of field name / variable name / value where the variable name matches a bind variable in the SQL. The field name provides information about the length that assists the SQL binding logic to properly substitute the values. Note that the variable name cannot start with `f1.` as its prefix. The batch job will use the value returned by the algorithm to set the bind parameter in the SQL statement. Also note that the logic binds all variables as a string, so for date variables the SQL must include the "TO_DATE" function.

The plug in spot receives a list of the ad hoc parameters for the batch job as name / value pairs. If the list includes parameters whose values are to be used in selecting records, your algorithm may be used to identify the relevant batch parameter passed as input and populate the field name and output bind variable appropriately.

The product provides a base algorithm type for this plug-in spot that simply defines a parameter for the SQL. It also includes parameters for the strategy and the key field name.

Copyright © 2026, Oracle and/or its affiliates.

This algorithm type may be used by any custom batch process where the SQL does not rely on any special bind variables that must be determined. Simply create an algorithm for the algorithm type and provide the appropriate SQL. Refer to the algorithm type Select Records by Predefined Query (F1-PDB-SR) for more information.

Process Records Algorithm

The other important algorithm to design when implementing a plug-in driven batch process is the Process Record algorithm, plugged in on the batch control page. This algorithm is called for each record selected for the process. It receives all the information that was selected from the Select Records plug-in.

For the ad hoc processing batch process, algorithms plugged into this spot are responsible for doing the work for each record based on the desired logic.

For the extract batch process, algorithms plugged into this spot are responsible for returning the data that should be written to the file. Many of the parameters for this plug-in spot are only applicable to extract processes. Refer to Extract Record Processing for more information on the functionality specific to extract processes.

Also note that algorithms for this plug-in spot will be passed two Booleans, isFirst and isLast, to indicate if the current work unit is the first and/or last for that thread. This allows for the plug-in to do additional work if needed. Note that the isFirst indication is available for both types of batch processes, ad hoc and extract. However, the isLast indication is only applicable for the file extract batch. For the ad hoc batch process this value will always be set to **false**. Extracts will always execute in a single database transaction. In a single transaction run, any error causes the run to be aborted so that it restarts from the beginning when resubmitted. This is done to avoid partial files being written along with inaccurate setting of the isLast element.

Parameter Validation Algorithm

Plug-in driven batch processes may require batch parameters specific to that process. The Parameter Validation algorithm is responsible for checking the parameters at run time and issuing errors if any values are invalid. The algorithm is invoked prior to the Select Records and Process Records algorithms. It is plugged in on the batch control portal. If multiple algorithms are configured, the one with the highest sequence is used. Algorithms for this spot should use standard error message reporting.

Configuring a New Process

The following points summarize the steps needed to implement a new background process that acts on records in the system using plug-ins for the specific functionality:

- Verify the SQL that the background process should execute. Keep in mind that all the data selected in the SQL is available to pass to the plug-in that processes the records. If the performance of the background process is important, be sure to consult with a DBA when designing the SQL.
  - If the SQL does not require any custom variables to substitute at runtime, create an algorithm for the base algorithm type F1-PDB-SR and configure the SQL. In addition, configure the strategy and the primary key name (for the **Thread Level SQL Select** strategy).
  - If the SQL requires custom variables, a new plug-in script must be designed and coded to populate the variable names and values using the algorithm entity **Batch Control - Select Records**. Besides defining the variables, the algorithm must also indicate the strategy and the primary key name (for the **Thread Level SQL Select** strategy). Define the algorithm type for the newly created script. The first parameter of the algorithm type must be the SQL as illustrated in the base algorithm type. Note that

Copyright © 2026, Oracle and/or its affiliates.

the other parameters are available for use by this algorithm type if needed. Define the algorithm, populating the SQL as appropriate (using the custom variables).

- Design the logic required to process each record. If your edge product does not deliver a suitable plug-in, create a plug-in script where the algorithm entity is **Batch Control - Process Record**. Note that the plug-in receives all the information selected in the SQL defined in the Select Records plug-in. For an ad hoc process, the algorithm should perform whatever process is required based on the business use case. Note that the background process is responsible for committing the records.

  In either case, if a new processing script is required, define the algorithm type and algorithm for the newly created script.
- If the process has special parameters, create a script to determine if the parameters passed at run time are valid. Define the algorithm type and algorithm for the newly created script.
- Create a batch control by duplicating the base template `F1-PDBG`. Plug in the algorithms created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the Select Records and Process Records plug in algorithms.

Extract Record Processing

A process that selects records and produces output for each record to write to a file to integrate with external systems is called an Extract Process. The base batch control Plug-in Driven Extract Template (`F1-PDBEX`) may be used as a template.

> **NOT_SUPPORTED**
>
> Please read the topic Ad hoc Processes. Its information about the Select Records plug-in and some of the information about the Process Record plug-in apply to extract processes. This topic focuses on aspects that are unique to extract processes.

The system provided process includes parameters to configure the file path and file name for the created file. The file path supports all the functionality described in the topic Referencing URIs. The file name supports system substitution variables, such as run number, thread number, user and date / time. The process also supports compression if an appropriate extension is used. Refer to the File Name parameter description in the batch control for more information. Also see sections below for additional information for XML or JSON ouput formats.

> **Note**
>
> Refer to Flexible File Name / Writing Multiple Files for information about support for substituting a business value into the file name.

For extract processes, the Process Record algorithm is responsible for returning the data that should be written to the file in one or more XML instances along with the schema name(s) that describes the XML instance(s). The program will write the data to the file as per the format indicated in the File Format batch parameter.

By default the service uses the OUAF format for date and time. To override this and use XSD format, configure the Date Time Format batch parameter to 'XSD''.

Copyright © 2026, Oracle and/or its affiliates.

If an existing schema satisfies the output requirements, it may be used. Otherwise, you should define a data area to indicate the output format of the records as appropriate. Note that you should define an appropriate Field for each element in the schema. This is needed for the fixed position and for JSON / JSON lines formats, but is good practice for any output format. Refer to `F1-PDBGenProcExtractRecord` for an example of a data area with fields defined for each schema element.

There are two options for designing and configuring the Process Record algorithm:

- Create a specific **Process Record** algorithm type that encapsulates the extract logic. The product provides a base algorithm type that illustrates the basic technique to follow. Refer to the algorithm type General Process - Sample Process Record Extract (`F1-GENPROCEX`) for more information.
- Create a file integration type that defines the records to be included in the extract and configure a plug-in driven batch control that references that integration type. Refer to Extract Using File Integration for more information about implementing this type of functionality.

Your specific edge product may provide other Process Record algorithm types out of the box. Use the Algorithm Entity portal to find examples.

XML Format - Additional Considerations

When using XML format for the output, the batch process includes a parameter to define the Grouping Name. This is the root node to use to surround the whole output file. Refer to the batch parameter description for more information. When requesting a Multi-threaded Extract and using the concatenation option, the Grouping Name surrounds the full content of the concatenated output.

Each call to the Process Record plug-in returns one or more schemas with content. The content is written within a node using the schema name.

Your Process Record plug-in may return multiple schemas with information that is all part of the same overall record. For example, the output could include account information and related service agreement details:

```
<root>
...
<CM-AccountRecord>
<account type="group"
    <accountId>1234567890</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa type="group"
    <saId>123457665</saId>
    ...
</sa>
</CM-SAInfo>
<CM-SAInfo>
<sa type="group"
    <saId>1234588913</saId>
    ...
</sa>
</CM-SAInfo>
```


Copyright © 2026, Oracle and/or its affiliates.

```
...
</root>
```

In this case, it may be desired to wrap all that information in a grouping XML tag so that all the information for one record is grouped together. Because the Process Record plug-in could be returning header or footer records that are not part of a given selected record's extract information, the batch process doesn't know what schemas returned by the Process Record plug-in belong together logically. The output parameter 'record XML node' in the schema collection should be used to indicate the outer XML node to use to group related information together. In this example, one call to the Process Record plug-in returns results with the node 'record' populated for the account and service agreement schemas:

```
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-AccountRecord</schemaName>
<schemaType>F1DA</schemaType>
<data><account><accountId>1234567890</accountId>...</data>
</SchemaInstance>
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-SAInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><sa><saId>123457665</saId>... </data>
</SchemaInstance>
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-SAInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><sa><saId>1234588913</saId>... </data>
</SchemaInstance>
```

All schemas returned from a single call to the Process Record plug-in that have the same record XML node will be grouped in the written output within that XML tag. This snippet shows the results of two calls to the Process Record plug-in.

```
<root>
<record><CM-AccountRecord>
<account>
    <accountId>1234567890</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa>
    <saId>123457665</saId>
    ...
</sa>
</CM-SAInfo>
<CM-SAInfo>
<sa>
    <saId>1234588913</saId>
    ...
</sa>
</CM-SAInfo>
</record>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<record>
<CM-AccountRecord>
<account>
    <accountId>987654320</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa>
    ...
</sa>
</CM-SAInfo>
</record>
</root>
```

JSON / JSON Lines Format - Additional Considerations

The extract process supports two types of JSON output.

- **JSON Document**. This format produces a valid JSON document. This output format supports returning a single JSON object with the records defined as an array or as an array of records.
- **JSON Lines**. The format produces a series of JSON objects. Each output record is a separate JSON object.

> **Note**
>
> These output formats do not support including a header or footer as part of the extract.

> **Note**
>
> The recommendation is to return only one schema instance for each call to the Process Record plug-in spot. This is because the service that converts from XML to JSON must have a schema name that describes the payload. If multiple schema instances are returned, it treats each returned instance independently and doesn't support nesting or other relationships between the data. For example, if the output returns one instance with Account data and then 3 instances with Service Agreement (SA) data (related to the account), in delimited, fixed position and XML, those four instances can be written one after the other. However, in JSON, the data for the 3 SA instances should be returned as an array within the Account object. In order to support this, the Process Record plug-in should have a governing schema that has the Account and the SA list all defined within it.

JSON Document - Single JSON Object

If you provide a Grouping Name in the batch parameter, the program will produce a JSON object which contains an array of the output records using the grouping name as the array name.

Copyright © 2026, Oracle and/or its affiliates.

For example, imagine the Grouping Name is 'records' and the output from one call to Process Records is this:

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Now imagine three calls to the Process Record plug-in with separate results. In this case the 'files' node is the group node for the content of each call. This is mapped to the 'records' array name.

```
{
 "records": [
  {"fileName": "file1.txt", "externalReferenceId": "1234"},
  {"fileName": "file2.txt", "externalReferenceId": "3456"},
  {"fileName": "file3.txt", "externalReferenceId": "5678"}
 ]
}
```

The following are points related to how the schema returned by Process Records may be defined and how the JSON output format handles the use case. In all cases, the assumption is that the batch parameter Grouping Name is defined as 'records'. And in each case, the Resulting File Output imagines three calls to Process Records.

Table JSON Output

Use Case	Schema Definition	Resulting File Output

If the output schema contains a single group, output returned by the plug-in is grouped by the a group node. This node does not have importance when writing the record to file and the group node is removed. The record is written directly inside the JSON array.	<schema>
	  <group1 type="group">
	  <fileName dataType="string"/>
	<externalReferenceId dataType="number"/>
	  </group1>
	</schema>
		The 'group1' node is dropped and its contents are mapped to the 'records' array.
		
		{
		"records": [
		{"fileName":"file1.txt",
		"externalReferenceId":"1
		,234"},
		{"fileName":"file2.txt",
		"externalReferenceId":"1
		,234"},
		{"fileName":"file3.txt",
		"externalReferenceId":"1
		,234"}
		]
		}

Copyright © 2026, Oracle and/or its affiliates.

Table (Cont.) JSON Output

Use Case	Schema Definition	Resulting File Output

If there is an outer group node in the schema which contains nested group(s) within it, the program skips the topmost group node.	<schema>
	  <group1 type="group">
	  <group2 type="group">
	  <fileName dataType="string"/>
	
	<externalReferenceId dataType="number"/>
	  </group2>
	  </group1>
	</schema>
		{
		"records": [
		{"group2":
		{"fileName":"file1.txt",
		"externalReferenceId":"1
		,234"}},
		{"group2":
		{"fileName":"file2.txt",
		"externalReferenceId":"1
		,234"}},
		{"group2":
		{"fileName":"file3.txt",
		"externalReferenceId":"1
		,234"}}
		]
		}

If schema contains more than one group or a group along with other types like list, field or raw then it does not skip any group node name.	<schema>
	  <group1 type="group">
	  <fileName dataType="string"/>
	
	<externalReferenceId dataType="number"/>
	  </group1>
	  <list1 type="list">
	  <fileDetails dataType="string"/>
	  </list1>
	  <fileSize dataType="number"/>
	</schema>
		All nodes are included in each entry in the array.
		
		{
		"records": [
		{"group1":
		{"fileName":"file1.txt",
		"externalReferenceId":"1
		,234"},"list1":
		[{"fileDetails":"Test
		file"}],"fileSize":"1,02
		4"},
		{"group1":
		{"fileName":"file2.txt",
		"externalReferenceId":"1
		,234"},"list1":
		[{"fileDetails":"Test
		file"}],"fileSize":"1,02
		4"},
		{"group1":
		{"fileName":"file3.txt",
		"externalReferenceId":"1
		,234"},"list1":
		[{"fileDetails":"Test
		file"}],"fileSize":"1,02
		4"}
		]
		}

JSON Document - Array of Output

Copyright © 2026, Oracle and/or its affiliates.

If you choose the JSON document output format and populate 'suppress' for the Grouping Name, the program will produce an array instead of an object.

To illustrate, we assume the same definitions used in the above Single JSON object format examples. The following is the output of one call to the Process Records plug-in.

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Now imagine three calls to the Process Record plug-in with separate results. In this case the 'files' node is the group node for the content of each call. This node is dropped and the output from each call is an object within the array.

```
[
  {"fileName": "file1.txt", "externalReferenceId": "1234"},
  {"fileName": "file2.txt", "externalReferenceId": "3456"},
  {"fileName": "file3.txt", "externalReferenceId": "5678"}
 ]
```

The same considerations in the JSON Output table above apply to this configuration.

JSON Lines

For the JSON Lines output format, the Grouping Name is not applicable (and ignored). Each call to the Process Record plug-in spot is written as an object. All the considerations in the JSON Output table above apply to this configuration.

To illustration, here is the output for a single call to the Process Record plug-in spot.

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Then this is an example after three calls to Process Record. The output is written out as objects with no commas.

```
{"fileName": "file1.txt", "externalReferenceId": "1234"}
{"fileName": "file2.txt", "externalReferenceId": "3456"}
{"fileName": "file3.txt", "externalReferenceId": "5678"}
```

Multi-threaded Extract

When extracting data, it's possible that the volume of data warrants running the job multi-threaded. In this case, a separate file is produced with the thread number included in the file name. You may proactively include the thread number as a substitution variable when indicating the file name. If not, the system appends the thread number.

Copyright © 2026, Oracle and/or its affiliates.

As a convenience, the system supports concatenating the extract files produced by the various threads at the end. The system does this by finding files whose file names match except for the thread number. As such, the file name may not contain date or time if the concatenation parameter is true. Please note the following with respect to the concatenation feature:

- The content of each file is concatenated together "as is". If the individual files have header, footer or summary information, there is no logic to consolidate that information in the concatenated file. If you want a single header, footer or summary, you must run the extract single threaded.
- Concatenation is not supported if the file name indicates the zip compression should be used.
- The individual thread files are retained.
- If the file format is XML, the files written for the individual threads will be written with an extension of ".tmp". The final concatenated file is surrounded by the Grouping Name defined on the batch job as a parameter (if not suppressed).
- If the file format is JSON Document, the files written for the individual threads will be written with an extension of ".tmp". The final concatenated file is treated as an object with an array defined by the Grouping Name, or treated as a single array (if the grouping name is configured as "suppress").

Skipping Records

By default, the extract process expects one or more schemas to be returned by each call to the Process Record plug-in. (Ideally when designing any process, the Select Records algorithm only selects records that should be processed and the Process Record algorithm does not need to confirm whether the record should be part of the processing or not.) If however there is a need to check a condition in the Process Record plug-in and it is found that no data needs to be extracted for this record, the process record algorithm should return the 'is skipped' output parameter to **true** so that the program doesn't write an empty row.

Note that the base process records algorithm for file integration types `(F1-FILEX-PR)` will set the ‘is skipped’ parameter to **true** if the file integration record extract record algorithm returns no data.

Configuring a New Process

The points documented in the topic related to configuring a new process with respect to the Select Records algorithm apply to extract processes.

For the Process Records algorithm, it is responsible for returning one or more schema instances populated with information that should be written to the file. If your edge product does not deliver a suitable plug-in, create a plug-in script where the algorithm entity is **Batch Control - Process Record**. Note that the plug-in receives all the information selected in the SQL defined in the Select Records plug-in.

Refer to Configuring an Extract Process Using File Integration Type for information on designing extracts using this configuration.

If a new processing script is required, define the algorithm type and algorithm for the newly created script. Create a batch control and refer to the batch template Plug-in Driven Extract (`F1-PDBEX`). Plug in the algorithms created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the Select Records and Process Records plug in algorithms.

- Flexible File Name / Writing Multiple Files

Copyright © 2026, Oracle and/or its affiliates.

- Extract Using File Integration

Flexible File Name / Writing Multiple Files

By default, the data is written to the File Name defined in the batch parameter. The file name supports several dynamic substitution variables as defined in the detailed description for the batch parameter in the Plug-in Driven Extract Template batch control (F1-PDBEX). Also refer to Extract Record Processing for information about the file name and multi-threaded processes.

The system also supports more advanced file creation for a single batch run. Consider the following use cases

- An extract is for data associated with multiple service providers where each service provider should receive its own file. In this use case, the unit of work is the service provider so a file is produced for each work unit.
- An implementation operates with multiple jurisdictions represented by CIS Division. A certain extract should result in a file per division so that each division only gets its relevant data.
- An extract of Person information should separate individual person data from business person data. In addition, for each person, the contact information should result in a separate file. In this use case, one unit of work includes data to be written to more than one file.

To support the above use cases, the Process Record algorithm supports returning the File Name in the output:

- The file name is included in the output Schema Instance group. You would populate this if you want the data for a specific Schema entry to go to this file name. In the person extract example above, the schema that includes the contact information would indicate the specific contact file.
- The file name is an output field outside the schema list. You would populate this if you don't have different files for different schemas but you still want to override the file name defined in the batch parameter. The service provider and division examples above would populate this parameter with the desired file name. The person extract example could populate this field for all the non-contact information.

The batch process checks the output after each call to the Process Records algorithm. For each schema, if there is a specific File Name associated with the schema, the data for that schema is written to that file. Otherwise, if the output file name is provided, the data is written to that file. Otherwise, the File Name in the batch parameter is used.

> **Note**
>
> **Substitution Variables.** After the call to Process Records, if there are any system variables in the file name, they are resolved at this time.

Managing File Names

This type of requirements means that the Process Record algorithm needs to provide a file name that has some business value to distinguish the individual files. For example, the service provider or the division or the person type value must be part of the file name. However, the recommendation is not to have the file name entirely determined by the algorithm. Rather the suggestion is that you still allow the user submitting the batch job to provide a desired file name as a batch parameter. That way, it can include any of the supported substitution variables for system information (such as batch run number, or date /time stamp) that ensure

Copyright © 2026, Oracle and/or its affiliates.

your files are unique for a given batch run. The algorithm can designate a substitution variable that it looks for to replace with the business value.

For example, using the files per division use case, imagine the user indicates the file name EXTRACT_FILE_<DIV>_{BN}_{RN}_{TN}.csv. The process record algorithm receives this parameter. It knows to look for "<DIV>" and replace it with the actual division. Imagine the division code is A100, it would return the file name EXTRACT_FILE_A100_{BN}_{RN}_{TN}.csv in its output, along with the data to write to that file for this unit of work.

For the person information use case above, where two files are written, the suggestion is to use the File Name parameter for one of the files, for example the person information file. An additional parameter should be used to define the second file, the contact information file. So the users submitting the batch job would define the following parameters:

Parameter Name	Description	Value	Comments
fileName	File Name	persons_<personType>_{BN}.txt	This is the file name parameter provided with the extract.
contactFileName	Contact File Name	contacts_{BN}.txt	This is a special parameter defined by the CM extract batch control.

The Process Record algorithm would look for the File Name parameter provided as input, find the text "<personType>" and replace it with the person type value and return that file name in the output. When building the list of schemas to return in the output, the one that includes the person contact information would include the file name found in the Contact File Name parameter provided as input.

> **Note**
>
> If an extract uses file integration types, the file integration record's extract process algorithm would have the above responsibility for indicating the file name and pass it back to the Process Record algorithm.

Multiple Threads

This functionality works as expected with multiple threads. Based on the use case and the implementation of the Select Records algorithm, each thread could be producing files for the same business value. Let's use the division use case. Imagine an implementation has three divisions: A100, B200, C300. Now imagine that the extract is for financial information for an account. The select records algorithm should be threading by account id so that the data is evenly distributed across the multiple threads. But within each thread, accounts could be in any of the 3 implementation's divisions. So each thread could produce 3 files for each of the 3 divisions.

As described in Multi-threaded Extract, the system supports an option to concatenate, allowing you to consolidate the files such that you have 3 complete files for the 3 divisions. The same features and limitations described in that topic apply for this use case.

Closing Files

An important consideration when using this functionality is that attention should be paid to how many files are open during the batch process. Each open file takes some memory to manage it. As such, the product has imposed a limit of only 10 open files (per thread) during the run of

Copyright © 2026, Oracle and/or its affiliates.

the batch job. When designing a batch to write to multiple files, the product strongly encourages you to design the process so that whenever possible, all the data for one file can be written and that file can be closed before going on to the next file.

Let's look at our examples.

- For the service provider example, the unit of work for the batch process is service provider. In this case, after each call to process records, the data can be written to the file and then the file can be closed.
- For the division example, the data selected should be threaded by the proper unit of work related to the purpose of the extract, such as Account or Asset. But the data should be ordered by division so that all the data for each division is processed before the next division's data. Each division's file can be closed once the next division's data starts getting processed.
- For the person extract example, the contact file is used for all the data so that file needs to remain open throughout the duration of the job. However, the person information gets segrated by person type. The Select Records could order the data by person type so that all the data for one type could be written and then all the data for the other type could be written. However, given that for any run no more than 3 files will ever be open, this use case is one where there is no need to actively manage closing files during the process.

How will the batch program know when to close a file? Since the batch program is agnostic and does not know what type of data is being processed nor when files should be closed, the Process Record algorithm should provide this information.

> **Note**
>
> The batch process will issue an error if Process Records indicates that the file to write to is one that was already closed. Reopening closed files is not allowed. The design of the process needs to be reviewed if this error is received.

To manage this, the batch process provides the Process Record algorithm with the list of open files (with the system substitution variables unresolved) and a Boolean to mark whether the file should be closed.

> **Note**
>
> If an extract uses file integration types, the file integration record's extract process algorithm receives this list of open files and would have the responsibility described here to manage closing files.

To explain further, we'll use our examples.

File Per Division Example

This use case is that there are 4 divisions in the implementation: NORTH, EAST, SOUTH, WEST and that the data (account data) should be grouped in separate files by division.

The recommendation is that the Select Records algorithm selects the data ordered by Division.

The batch process passes to the Process Record algorithm the list of open files (there should be only be zero or one). In this case, the only indication that the data for one division is finished is if the Process Record algorithm sees that its data is for one division and the open file listed is for a different division. Each call should prepare its file name. Then it should see if its file

Copyright © 2026, Oracle and/or its affiliates.

name is in the list of open files. If not, it should indicate that the other open file should be closed. (Note that it doesn't need to add itself to the 'open list'. The batch program is only looking for entries marked as closed. It will ignore entries that are not marked as closed).

Assume the file name in the batch parameter is myFile_<DIV>_{BN}.txt

The data is sorted so that the accounts in the EAST division are processed first. Let's imagine this is a work unit that is also in the EAST division. It receives the file name and the list of open files.

```
...
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>myFile_<DIV>_{BN}.txt</value>
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <closeFile></closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

The algorithm substitutes its division into the <DIV> portion of the file and populates that in the output. It then compares its file name with the current list of open files. It finds its file so it doesn't need to do anything else.

```
<hard>
...
<fileName>myFile_EAST_{BN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile></closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

If we then imagine a call to the process records algorithm where the division is NORTH and it's the first entry for that division, here is what it receives:

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>myFile_<DIV>_{BN}.txt</value>
```


Copyright © 2026, Oracle and/or its affiliates.

```
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

The algorithm substitutes its division into the <DIV> portion of the file and populates that in the output. It then compares its file name with the current list of open files and doesn't finds its file name. So it indicates that the existing open file should be closed.

```
<hard>
...
<fileName>myFile_NORTH_{BN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

File Per Service Provider Example

In this use case, the unit of work is service provider and each service provider should have its own file. In this case, as the Process Record algorithm is preparing the data and the file name unique to its service provider, it can also indicate that the file can be closed. There is no need to wait for the next call to Process Record to indicate that the previous file can be closed. Let's assume the file name is MYFILE_<serviceProvder>_{BN}_{TN}.txt. Each call to Process Records will pass the batch parm file name and no open files.

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>MYFILE_<serviceProvider>_{BC}_{TN}.txt</value>
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <fileName></fileName>
    <closeFile></closeFile>
  </OpenFile>
</openFiles>
```


Copyright © 2026, Oracle and/or its affiliates.

```
...
</hard>
```

The algorithm is processing data for service provider A100. It returns the following information related to the file name, namely the single File Name in the output related to this service provider and adds its file to the open file list and indicates that it should be closed. (Note that file-per-work-unit is the only known use case where the algorithm will know that it is the last entry for a file and to close itself.)

```
<hard>
...
<fileName>MYFILE_A100_{BC}_{TN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>MYFILE_A100_{BC}_{TN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

Multiple Open Files Example

This example assumes that the file names are defined as per the Managing File Names section above. We'll assume that the algorithm designer chose to not actively close the person information file since there are never more than three files open. Similarly, we'll assume that they chose not to order the data by person type. Here is what is passed in in the middle of the run:

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>persons_<personType>_{BN}.txt</value>
  </BatchParm>
  <BatchParm>
    <name>contactFileName</name>
    <value>contacts_{BN}.txt</value>
   </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <fileName>persons_P_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
   <OpenFile>
    <fileName>persons_B_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
  <OpenFile>
    <fileName>contacts_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</openFiles>
...
</hard>
```

An instance of one call to Process Records algorithm that has details for an individual ("P" person type).

```
<fileName>persons_P_{BN}.txt</fileName>
<fileOutput>
  <listValue>
    <SchemaInstance>
      <recordXMLNode>person</recordXMLNode>
      <schemaName>CM-PersonDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName></fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
  <listValue>
    <SchemaInstance>
      <recordXMLNode>person</recordXMLNode>
      <schemaName>CM-AddressDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName></fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
  <listValue>
    <SchemaInstance>
      <recordXMLNode></recordXMLNode>
      <schemaName>CM-ContactDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName>contacts_{BN}.txt</fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
</fileOutput>
```

This XML example illustrates the following points:

- There is an override file name in the fileName element.
- The first two schemas don't have any value in the fileOutput\listValue\SchemaInstance\fileName so they should be written to the file in fileName. Note that these two schemas also include a Record XML Node. That will ensure that if the output is XML format, the data from both of those schemas will be wrapped by the 'person' XML Node. This is not functionality related to this epic. It's just trying to illustrate multiple features.
- The third schema has an entry in fileOutput\listValue\SchemaInstance\fileName . So that data is written to a separate file.

Copyright © 2026, Oracle and/or its affiliates.

Extract Using File Integration

The product provides separate configuration that may be used to define how to extract the data and how to format it. This configuration provides more metadata related to defining how to get the data to extract and how to format the data. You may have many use cases where using this configuration rather than having all the logic directly in the Process Records algorithm is beneficial.

The configuration can support simple extracts, containing a single record type and possibly a header or footer record. It also supports more complex integrations, involving multiple record types and parent/child relationships between the records.

File Integration Type Configuration

The framework supplies two objects that work together to define the processing needed to create a specific type of extract:

- The file integration type which describes and classifies the extract configuration.
- The file integration record where the sequence, format and logic to extract the data required for a given record type are captured.

File Integration Type

The file integration type defines the integration classification. Currently, only **Extract** is supported. In addition, the file integration type can be assigned to a category. The category may be used to identify extracts with a common purpose or target such as collection agency referrals. Your edge product may deliver a number of categories or you may define your own.

The system provides a business object for extract file integration types (**`F1-ExtractFileIntegrationType`**) that is expected to be used for most use cases

File Integration Record

The file integration record is the object that is responsible for defining a record's format and capturing the record details. A separate integration record must be defined for each record type included in the extract.

The record sequence defines the order that records are processed by the system. The parent sequence defines the record type of which this record is a child. Child records may also be referenced as a parent of lower level child records. The base Process Records algorithm provided by the framework processes all child records of a record type before processing the next record type in sequence.

For extracts that are XML format, the system supports creating a grouping node surrounding all the output that belongs to the same entity. File integration records define a record XML node that will be used as the grouping node. Each record type in the list that belongs within the same group would indicate the same value in the record XML node. Refer to theXML Format - Additional Considerations section for more details.

> **Note**
>
> The system does not support nested record XML nodes. All child records of a record type must define the same record XML node as the parent.

Each file integration record references a data area that defines the extract record’s format, allowing the data to be extended. The **Extract Records** algorithms are responsible for

Copyright © 2026, Oracle and/or its affiliates.

populating the data area. The file integration record supports multiple extract processing algorithms so that the custom information can also be populated.

The extract record plug-in also supports the functionality described in the Flexible File Name / Writing Multiple Files topic.

The system provides a business object for extract file integration records (**`F1-ExtractFileIntegrationRec`**) that is expected to be used for most extract record use cases.

Configuring an Extract Process Using File Integration Type

The following points summarize the additional steps needed to implement a new extract background process that uses a file integration type to define the specific functionality:

- Choose or create data area objects to define the extract record formats.
- Define the logic required to populate each extract record type. Your edge product may deliver suitable plug-ins or you may need to create a plug-in script where the algorithm entity is **File Integration Record - Extract Record**. Note that the Extract Record plug-in receives most of the information defined in the Batch Control - Process Records plug-in.
- Define the algorithm type and algorithm for each newly created script.
- Choose or create a category for the file integration type using lookup `FILE_INT_OBJ_FLG`.
- Create the file integration type and configure the file integration records using the defined data areas and algorithms.
- Create a batch control by duplicating the appropriate base template for an extract background process. Include the File Integration Type parameter (fileIntegrationType) on the batch control and reference the appropriate file integration type.
- Plug in a Process Records algorithm that reads the file integration type configuration and processes the record types. The system provides a base file integration record extraction algorithm (`F1-FILEX-PR`) that is expected to be used for most file integration plug-in extract processes.

Extending a File Integration Extract Process

Your edge product may deliver file integration types and batch controls designed for standard extract processes. However, extending the delivered extracts to add custom data is common practice for most implementations. You can customize the output from a delivered extract by:

- Identifying the record types that need additional details
- Creating a data area that defines the additional details and using that to extend the data area configured on the record type
- Creating a File Integration Record — Extract Record plug-in script, algorithm type and algorithm that populates the extra details
- Adding the algorithm to the integration record's extract algorithms collection

In some cases, you may wish to extend the extract by adding additional record types. To do this, add your custom file integration records to the file integration type, following the steps described in Configuring an Extract Process Using File Integration Type.

Uploading Records

The base product provides a background process to upload data from a file. The batch control Plug-in Driven File Upload Template (`F1-PDUPL`) may be used as a template. The process

Copyright © 2026, Oracle and/or its affiliates.

includes parameters to configure the file path and file name for file to upload along with other parameters to control how to handle missing files and how to rename the file once processed. The process supports uploading multiple files and may be run multi-threaded to process multiple files. Refer to the description of the batch control and its parameters for more information.

The following sections describe concepts and features related to this process.

A Step in the Overall Upload Process

This step in the upload of data is only one part of a typical upload end-to-end process. This step is sometimes referred to in the product as "Process X". The goal of this step is to get records from a file into database records, with minimal validation and processing. The data should be stored in records that are then processed by a second step (often referred to in the product as the "upload" step, for example "Payment Upload"). The second step, independent of the plug-in driven batch process described here, is responsible for validating the data and should be able to be threaded by individual records and have proper error handling for individual records.

Depending on the type of data being uploaded, the product may already supply appropriate tables that the plug-in driven upload process may populate. These could be staging tables, such as payment upload staging. Or they may be records with business objects that have a lifecycle designed to handle uploaded data, for example Business Flag. In such cases, the product will typically supply out-of-the-box background processes to validate and further process the data and finalize the upload. If the data to upload does not already have a base provided staging table, be sure to work with your implementation team to identify an appropriate table to use for the plug-in driven batch upload. In addition, confirm the design for the second step that is responsible for the detailed validation and finalization of the data.

> **Note**
>
> Designing your upload process to directly update data, as opposed to using a staging table, increases the likelihood of data validation exceptions. Refer to the **Exception Handling** section for more information and guidelines to be considered when designing such upload process.

The File Upload Algorithm

This background process requires an algorithm plugged into the plug-in spot **File Upload**. By default, this plug-in is called once for a given file but it can also be designed to process records in multiple unites of work.

The batch process opens the file and passes to the algorithm the file reader element. The algorithm is responsible for using provided APIs to read the content of the file and store the data in appropriate table(s). The algorithm can be designed to process all the records in the file as a single unit of work. Alternatively, it can be designed to process a smaller set of records and request to be called again to process the next set etc, until all records are processed.

The batch process commits the records uploaded by each call to the algorithm and keeps track of the position in the file of the last processed record. That position is provided to the algorithm with each subsequent call, allowing it to reposition the file to that record and continue processing from there. When all records in the file are processed the batch process closed the file and proceeds to the next file if any.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Plug-in scripts written to implement this type of algorithm must use the Groovy script engine version as the APIs are not accessible using the XPath scripting language. Refer to Additional Coding Options For Server Scripts for more information.

Sample Algorithms

The product supplies sample algorithms to illustrate calling the supplied APIs for processing different types of source data: comma delimited, fixed position and XML. In every case, the sample data supported by the upload uses 'degree day' information to illustrate the process. The system provides sample target records (based on the Fact maintenance object) in order to illustrate the step to store records based on the input data. Note that only sample plug-in scripts have been provided. No algorithm, algorithm type or batch control is configured to use the sample plug-in scripts. To view the scripts, navigate to the Script page, search for a Script Type of **Plug-in Script** and Algorithm Entity of **Batch Control - File Upload** and look for the 'Sample' scripts.

Note that the sample plug-in scripts provided by the product are supplied to illustrate use of the provided APIs. They should not be considered a template for how to implement a plug-in script for a real use case. The following highlight some points to consider when designing a file upload algorithm:

- **Error handling / resolution.** The sample plug-in scripts do some basic error handling related to the data to illustrate error handling. However, any errors found in this step require processing of the whole file to stop. As such, this plug-in should only report errors that are not possible to fix, but where the whole file should be rejected. If there are errors that can be adjusted in the data, then the recommendation is to not check for those errors at this step. Rather, this plug-in should simply populate the appropriate staging tables and let the next step check for validity. As described above, the next step should include the ability to mark individual records in error, allowing for users to fix the data and retry.
- **Target tables.** The sample plug-in scripts use Fact as the target for the resulting insert statements. As mentioned above, the decision of where to store the uploaded data must be carefully considered. There may already be existing tables that are specific to a given use case. If the data being uploaded does not have existing tables to use, review the product to verify what existing tables may be useful, such as Inbound Sync Request or Service Task. Be sure that the tables chosen support error handling, either out-of-the box or via designing an appropriate business object with a lifecycle that supports an error status and the ability to resolve the error. Also note that the Sample Flat File Upload plug-in illustrates a header record / detail record scenario. In this case, the header record is linked to the child record via a CLOB element. This is not the recommended technique. In a real use case, the header record should be linked to the child record via a separate database column to allow for searching.

Exception Handling

By default, any error that occurs while uploading a file causes the entire file to be skipped. The upload batch program managed an exception as follows:

- A rollback is issued to reverse updates since last commit.
- The error message is reported as part of the batch run.
- The file is skipped from further processing and its name remains unchanged.
- Since the default unit of work for an upload process is a file, and not an individual record within it, an error counts as one file unit in error towards the **Override maximum errors (maxErrors)** limit if specified as a batch parameter. If a limit was not specified or not

Copyright © 2026, Oracle and/or its affiliates.

reached, the batch run is considered completed regardless of how many files were successfully uploaded. If a limit has reached, the upload process is stopped and the batch run status is set to error.

If your upload process requires a more granular row level exception handling, your upload algorithm should be designed accordingly. With this approach, errors are fully trapped, managed and reported by your algorithm and never reach the upload batch program. In other words, to the batch process there were no errors.

Trapping errors by your custom upload algorithm is not trivial and should consider the following guidelines:

- Use the 'Savepoint Script Dispatcher' (F1-SavePointDispatcher) business service to perform all your updates along with a request to trap any errors. The service adheres to standard rollback and data integrity rules and as such is the recommended approach for trapping application errors.
- When the business service traps an error, your algorithm may report it as a batch run message or any other option that meets your business requirements.

> **Note**
>
> Do not report the number of errors as part of the algorithm's plug-in spot response 'unitsInError' element. Any value provided in this element counts as if the entire file is in error, i.e. increments the total number of errors by one towards the 'maxErrors' limit if specified.

- The upload batch process is not aware of errors trapped by your algorithm and will rename the file as successfully processed even if no record was successfully processed. It is the responsibility of your algorithm to determine when a file should be considered as not processed and issue an explicit error for the batch program to handle.

Splitting Large Files Prior to Upload

A single file is processed by a single thread of this batch process. Splitting a large file into smaller files allows the smaller files to processed by multiple threads of the upload process.

The base product provides a background process to split large files prior to upload. The batch control File Split Template (`F1FSPLIT`) may be used as a template. The process supports splitting multiple files and may be run multi-threaded to process multiple files. The process can also be set up to submit the subsequent upload process once the files are split.

Note that not all file contents can be split using this batch process. Basic XML, json and flat file records separated by a new line may be supported.

Refer to the description of the batch control and its parameters for more information.

Configuring a New Process

The following points summarize the steps needed to implement a new file upload background process:

- Verify the details of the data in the upload file and map the data to fields in one or more appropriate tables in the system.
- Design the logic required for reading the record details and identifying each record to properly create the insert statements for storing the data. The sample plug-in scripts provided by the product illustrate using the various APIs available for use. Create a plug-in

Copyright © 2026, Oracle and/or its affiliates.

script where the algorithm entity is **Batch Control - File Upload**. Create an appropriate algorithm type and algorithm for this plug-in script.

- Create a batch control by duplicating the base template. Plug in the algorithm created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the File Upload algorithm.

How to Re-extract Information

If you need to recreate the records associated with an historical execution of an extract process, you can - simply supply the desired batch number when you request the batch process.

How to Submit Batch Jobs

Most batch jobs are submitted via a batch scheduler. For cloud customers, refer to the Batch Scheduler for Cloud Implementations topic in the Oracle Utilities Cloud Service Foundation documentation for more information. For on premises customers, refer to your third party scheduler documentation for more information.

Batch jobs may be configured as Timed, which means they will automatically be run based on the timer frequency.

You can manually submit your ad hoc background processes or submit a special run for one of your scheduled background processes using the online batch job submission portal.

The system provides a service script `F1SubmitJob` to create a batch job submission request. Notes about this service:

- No parameter defaulting occurs when submitting a record using this script. All the desired parameter values must be provided.
- If the requested batch control is defined as part of a pool of batch controls using the **Batch Group** option then the batch job request is created for a batch control within the pool that has a low number of in progress requests. This technique allows for a more balanced load of submission requests across pool-based batch controls.

The script is also exposed as a REST web service (using the inbound web service `F1-SubmitJob`) to support batch job submission requests from an external system.

How to Track Batch Jobs

You can track batch jobs using the batch process pages, which show the execution status of batch processes. For a specified batch control id and run id, the Batch Run portal shows information about each thread, the run-instances of each thread, and any messages (informational, warnings, and errors) that might have occurred during the run.

> **NOT_SUPPORTED**
>
> For more information, refer to Tracking Batch Processes.

Copyright © 2026, Oracle and/or its affiliates.

How to Restart Failed Jobs and Processes

Every process in the system can be easily restarted if it fails (after fixing the cause of the failure). All you have to do is resubmit the failed job; the system handles the restart.

Assessing Level of Service

For some background processes, an implementation may wish to supply one or more algorithms that check some conditions to assess whether or not the process is performing as expected. An algorithm could be used to check the performance of the job to see if it is running as efficiently as expected. Or it could analyze the data processed by the background process to assess whether there may be some problem with the quality of the data.

The system provides a Level of Service plug-in spot on batch control to configure the appropriate algorithms for a given background process, if desired. Each algorithm is expected to return a value to indicate the 'level of service' determined along with a message indicating the reason for the value. The following Level of Service values are supported:

- **Normal**. Indicates that the algorithm did not detect any issues.
- **Warning**. Indicates that the algorithm found some issues that may or may not indicate a problem.
- **Error**. Indicates that the algorithm found some issues that should be investigated.
- **Disabled**. Indicates that the algorithm could not properly execute the level of service logic.

When multiple Level of Service algorithms are plugged in on the batch control, the overall Level of Service assessment for a batch control is populated as follows: if any algorithm returns Error, that value is returned; otherwise, if any return a Warning, that value is returned; otherwise Normal is returned. The results returned by each algorithm are also made available for review.

When viewing a batch control record, use the **Level of Service** action to display this information. The level of service is also part of the Health Check service.

Batch Resource Groups

Batch resource groups may be configured to prioritize resource allocation between various batch processes. For example, export type of processes may have a lower resource allocation priority relative to billing related processes.

The **Batch Resource Configuration (F1-BatchResourceConfig)** extendable lookup associates certain batch processes with one of three priority resource groups: 10 (High), 20 (Medium), 30 (Low). No setting for a batch control means it received the default priority relative to other processes.

Resource group information may be leveraged as follows:

- Database resource management tools may be configured to allocate resources to batch processes by their association to resource groups.
- In a Cloud installation, thread allocation prioritization can be fine-tuned by associating batch processes to batch resource groups.

The following sections provide additional information on these features.

Copyright © 2026, Oracle and/or its affiliates.

Batch Thread Prioritization

In a Cloud installation, batch processing functionality supports the capability of submitting batch jobs with high thread counts which leads to many threads of work being initially queued by the system. Batch resource groups are used to promote fair queuing of batch work such that batch controls associated with a higher priority resource group receive more service relative to those associated with lower priority resource groups.

Database Resource Management Tools

Database resource management tools may be used to prioritize resource allocation between various batch processes. For example, export type of processes may have a lower resource allocation priority relative to billing related processes. These tools allocate resources to groups of consumers, not to an individual batch process. They use rules that are based on certain session variables, to associate a batch process with a resource group.

In a Cloud installation, if a batch control is associated with a resource group the latter is included by default as part of the MODULE database session variable as follows: '<batch control>,R=<resource group>'. This allows DBAs to potentially take advantage of this information for resource prioritization purposes. On premise, this feature needs to be explicitly opted in using the **Expose Batch Resource Group** option in the **General System Configuration** feature configuration.

With this configuration enabled, DBAs can set up rules that parse the MODULE variable to identify the resource group for a batch process and apply prioritization rules accordingly. Note that most batch controls may not be associated with a resource groups so default allocation rules should be considered.

Note. This configuration does not include nor enforce the database resource management configuration. It only allows for such tool to be leveraged by DBAs as needed.

Batch Control Options

A batch control supports defining options. An option allows the product or your implementation to mark a batch control with additional information to be used for special processing or reporting.

Note. You can add new option types. Your implementation may want to add additional batch control option types. To do that, add your new values to the customizable lookup field **BATCH_CTRL_OPT_FLG** .

Defining Batch Controls

The system is delivered with all necessary batch controls. Implementations may define default values for parameters. In addition, implementations may define their own background processes.

To view background processes, open **Admin**, and then **System**, and then **Batch Control.**

Use the **Batch Control Query** portal to search for an existing batch control. Once a batch control is selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Background Processing Concepts and Online Batch Submission for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! If you introduce a new batch process, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Batch Control.** This zone provides information about the selected batch control.

The following zone may appear as part of the portal's **Algorithms** tab page

- **Algorithms.** This zone is used to maintain the algorithms that control important functions for instances of this batch control. Refer to the table that follows for a description of all possible system events. If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to Plug-in Scripts for more information.

The following zone may appear as part of the portal's **Options** tab page

- **Options.** This zone is used to maintain various configuration options associated with the current batch control.

> **ⓘ Note**
>
> **You can add new option types.** Your implementation may want to add additional batch control option types. To do that, add your new values to the customizable lookup field `BATCH_CTRL_OPT_FLG`.

The following zone may appear as part of the portal's **References** tab page

- **Batch Control References.** This zone list various configuration entities referencing the current batch control.

Actions

The page supports the following special actions:

- **Level of Service.** Use this action to display the output of the level of service algorithms for the batch control.
- **View Source.** If the batch program is shipped with the base package, you can use the adjacent icon to display the source code of this program in the Javadocs viewer.
- Batch Control Algorithms

Batch Control Algorithms

The following table describes all the system events supported for maintenance objects.

The **Algorithms** grid contains algorithms that control important functions for instances of this batch control. You must define the following for each algorithm:

The following table describes all the system events supported by batch controls. Note that some system events are batch program specific and as such are not available to all batch controls.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Error Post-Processing	Optional	Algorithms of this type are called if the batch process fails to complete due to an error. Multiple algorithms are allowed and are executed in sequence order. Refer to Error Post-Processing Logic for more information.
File Upload	Optional	Algorithms of this type are only applicable to plug-in driven background processes and are used to upload all records for a file. This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used.
Level of Service	Optional	Algorithms of this type are called to determine the Level of Service provided by a batch control. Multiple algorithms are allowed. Refer to Assessing Level of Service for more information.
Parameter Validation	Optional	Algorithms of this type are only applicable to plug-in driven background processes and are used to validate batch specific parameters. The algorithm is invoked prior to the select and process records plug-ins. This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used.
`Post-Processing`	Optional	Algorithms of this type are called after all threads are complete. Multiple algorithms are allowed and are executed in sequence order. Refer to Post-Processing Logic for more information.
Process Record	Optional	Algorithms of this type are only applicable to plug-in driven background processes and are used to process a specific record. This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used.

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Select Records	Optional	Algorithms of this type are only applicable to plug-in driven background processes and are used to define the SQL to use to select the records to process. This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used.

> **Note**
>
> **You can add new system events**. Your implementation may want to add additional batch control oriented system events. To do that, add your new values to the customizable lookup field `F1_BATCH_CTRL_SEVT_FLG`.

Defining File Integration Types

This section describes the functionality supported for viewing and maintaining file integration types.

Refer to Extract Record Processing for an overview of the file integration type functionality provided with plug-in driven background processes.

To view or add a file integration type, open **Admin**, and then **Integration**, and then **File Integration Type.**

Once a file integration type record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **File Integration Records** zone lists the record definitions for this integration, in hierarchical order. Use the zone actions to add, maintain or view a file integration record. Press the **Record Layout** icon to view a pop-up display of the record's schema.

Defining File Integration Records

This section describes the functionality supported for viewing and maintaining file integration record definitions.

Refer to Extract Record Processing for an overview of the file integration type functionality provided with plug-in driven background processes.

To view or add a file integration record, open **Admin**, and then **Integration**, and then **File Integration Type** and navigate to it from the **File Integration Records** zone.

The **File Integration Record** zone displays details about the selected record. Use the **View Record Types** zone action to review the complete list of record types included in the file integration type configuration.

Copyright © 2026, Oracle and/or its affiliates.

On-Line Batch Submission

The on-line batch submission portal enables you to request a specific background process to be run. When submitting a background process on-line, you may override standard system parameters and you may be required to supply additional parameters for your specific background process. After submitting your background process, you may use this portal to review the status of the submission.

The following topics further describe logic available for on-line submission of background processes.

- Batch Submission Creates a Batch Run
- Jobs Submitted in the Background
- Email Notification
- Running Multi-Threaded Processes
- Submitting Jobs in the Future
- Lifecycle of a Batch Job Submission
- Granting Access To Batch Submission
- Submission User and Method
- Batch Job Submission Portal

Batch Submission Creates a Batch Run

When you request a batch job to be submitted from on-line, the execution of the desired background process will result in the creation of a batch run. Just as with background processes executed through your scheduler, you may use the Batch Run portal to view the status of the run, the status of each thread, the run-instances of each thread, and any messages that might have occurred during the run.

> **Note**
>
> Your on-line submission record is assigned a status value so that you may know whether your job has been submitted and whether or not it has ended, however, it will not contain any information about the results of the background process itself. You must navigate to the Batch Run portal to view this detail.

Jobs Submitted in the Background

When you save a record on the batch job submission page, the batch job does not get submitted automatically. Rather, it saves a record in the batch job table. A special background process will periodically check this table for pending records and will execute the batch job. This background process will update the status of the batch job submission record so that a user can determine when their job is complete.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> At installation time, your system administrator will set up this special background process to periodically check for pending records in the batch job submission table. Your administrator will define how often the system will look for pending records in this table.

It should be noted that this special background process only submits one pending batch job submission record at a time. It submits a job and waits for it to end before submitting the next pending job.

> **Note**
>
> If you request a batch job to be run multi-threaded, the special background process will submit the job as requested. It will wait for all threads to complete before marking the batch job submission record as **ended**. Refer to Running Multi-threaded Processes for more information.

Email Notification

If you wish the system to inform you when the background process completes, you may supply your email address. The email you receive will contain details related to the batch job's output; similar to the job results you would see from your batch scheduler.

The email includes information from the product installation to help users that may be working in and submitting background processes in more than one environment.

- It includes the product name and product version for the product in the installed products list that is marked as the one to display.
- If a domain name message is configured in the installation messages, this information is also included.

> **Note**
>
> This assumes that during the installation process, your system administrator configured the system to enable email notification. Your administrator may also override the amount of detail included in the email notification.

> **Note**
>
> A batch control parameter may be used to indicate if the email should only be sent in the case of an error or success or both. Refer to Parameters Supplied to Background Processes for more information.

Running Multi-Threaded Processes

Many of the system background processes may be run multi-threaded. When submitting a background process on-line, you may also run a multi-threaded process. The fields Thread

Copyright © 2026, Oracle and/or its affiliates.

Count and Thread Number on the batch submission portal control the multi-threaded process requests:

- To run a multi-threaded process, indicate the number of threads in **Thread Count** and enter `0` in the **Thread Number**. If a previous run ended with one or more threads in error, a new submission may be created to restart all the threads that were in error. This submission would result in new Batch Thread Instance records added for each thread that previously ended in error. Note that if the previous run was marked to "Do Not Restart", then a new submission will result in a new Batch Run.
- In the situation where a previous run resulted in one or more threads in error, you may use the Thread Number to indicate a specific thread to restart. This is only valid if there is a previous run found that is not marked "Do Not Restart". This will result in a new Batch Thread Instance created for the specific thread in the previous run.
- To run a process as a single thread, enter Thread Count = `1` and Thread Number = `1` (or `0`). This will execute the background process single-threaded.

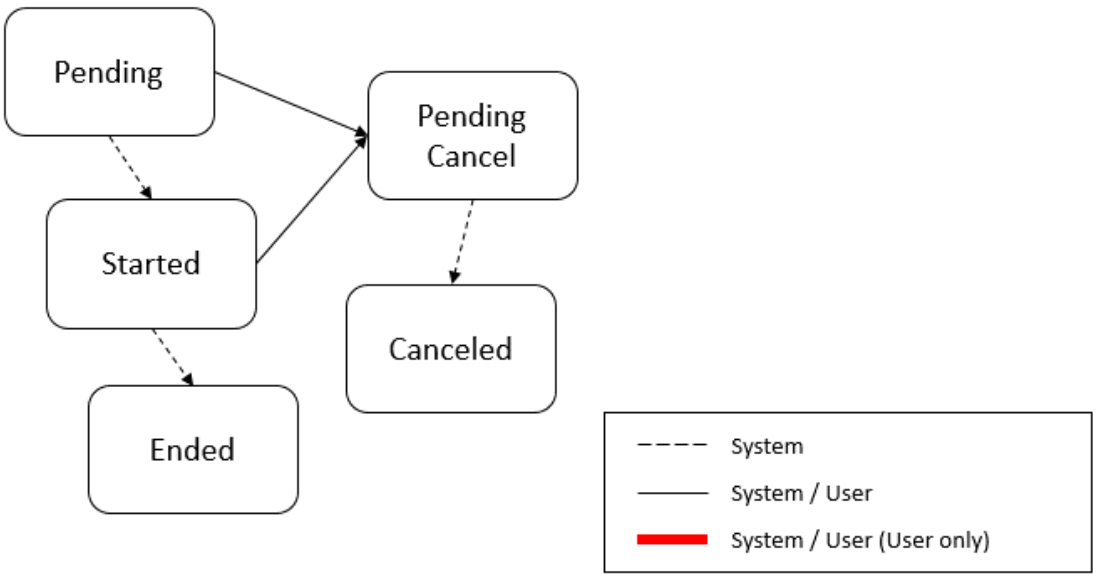
> **Note**
>
> When running a multi-threaded process, the special background process will wait until all threads are complete before marking the batch job submission record as **Ended**.

Submitting Jobs in the Future

If you wish to request a batch job to be submitted in the future, you may do so when creating your batch job submission record by entering a future submission date. The special background process, which looks for pending records in the batch job submission table, will only submit batch jobs that do not have a future submission date.

Lifecycle of a Batch Job Submission

The following diagram illustrates the lifecycle of a batch job submission record.



Copyright © 2026, Oracle and/or its affiliates.

Pending – Records are created in **Pending** status. Records in this state are put in a queue to be submitted.

Started – Once a pending record has been submitted for processing, its status will be changed to **Started**.

Pending Cancel – Users may request the cancellation of a pending record to prevent the batch job from being submitted. In addition, users may request to cancel a started batch job. The system will look for records in this status and look for batch runs that may need to be canceled for this batch job. Once it is confirmed that any batch runs are stopped, the status changes to **Canceled**.

Ended – When the batch job has finished processing, its status will be changed to **Ended**. Note that batch jobs in **Ended** status may have ended in error. You will need to navigate to the Batch Run portal to determine the status of the batch run.

Granting Access To Batch Submission

Every batch control must reference an application service. When you link a batch control to an application service, you are granting all users in the group the ability to submit the associated batch job for execution.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

When the batch control is added to the batch job table, the system checks if both the user submitting the job and the user ID recorded on the batch submission record have access rights. Application security does not apply when viewing a batch control or an associated batch run.

> **Note**
>
> Base batch controls will be associated with base owned application services by default. Implementations will need to ensure the appropriate user groups are linked to these application services.

Submission User and Method

Batch jobs track submisssion details using the following attributes.

The **Submission User** indicates who initiated the batch job and may differ from the batch user who controls permissions and auditing during the actual execution of the job. This value is not populated for timed batch jobs or batch jobs submitted via a scheduler.

The **Submission Method** indicates how the batch job was initiated. Valid methods include:

- **Online** - populated when a user manually creates or manually duplicates a batch job submission.
- **Generated** - populated by algorithms that submit a batch job and by 'initiator' batch jobs that submit other batch jobs.

Copyright © 2026, Oracle and/or its affiliates.

- **Scheduled** - populated by the DBMS scheduler. Note that for other schedulers, it is not advised to use this value as there may be logic and validation related to records created with this value that may not apply to other schedulers.
- **Timed** - populated by the batch daemon that submits timed batch jobs.
- **Other** - populated when no other value is provided.

Additional values may be added by your implementation. Refer to the lookup `SUBMIT_METH_FLG` for a list of the submission methods supported by your system.

Batch Job Submission Portal

To view a submitted batch job, use **Menu**, and then **Tools**, and then **Batch Job Submission**. You are brought to a query portal with options for searching for a Batch Job Submisssion entry.

Once a batch job submission entry has been selected, you are brought to the maintenance portal to view the selected record.

The inline help provides information about several fields. Refer to Parameters Supplied to Background Processes for more information about the fields on this page.

Refer to On-Line Batch Submission for detailed explanations about many feaures of online submisssion.

Actions

This page has three special actions.

- Using the **Duplicate** button opens a window with the selected batch job submisssion details displayed. These fields, including the batch parameter values, may be overidden. Pressing **Save** will create a new Batch Job Submission entry.
- Use the **Rerun** button to rerun the selected batch job. A window opens displaying the details of the batch job submisssion to be rerun. These fields, including the batch parameter values, may be overidden. Pressing **Save** will create a rerun Batch Job Submission entry.

> **Note**
>
> Rerun is only intended to be used for background processes that download information that belongs to given run number. It is typically only required if you need to download an historical run (rather than the latest run).

- **Cancel** is only applicable for batch job submisssions that are in **Pending** or **Started** status and do not have a submission method of **Scheduled** . The action is not visible for other statuses. Refer to Lifecycle of a Batch Job Submission for more information.

Tracking Batch Processes

The topics in this section cover the tools provided for reviewing the status and performance of batch processes. Refer to Defining Batch Controls for more information on how batch control codes are defined.

- Batch Run
- Batch Day Dashboard

Copyright © 2026, Oracle and/or its affiliates.

Batch Run

The batch run portal shows the execution status of a selected batch process. The information includes thread status summary, details about each thread, its instances, and any error To Do Entries and messages (informational, warnings, and errors) that might have occurred during the run. It also includes information about files that may have been created by the process along with various types of analytical reports.

You can access the portal from the **Main Menu**, and then **Tools**, and then **Batch Run**. You are brought to a query portal with options for searching for a specific batch run. Once a batch run has been selected you are brought to this portal to view the selected record.

The following points should help understand the concept of threads:

- Many batch jobs cannot take advantage of your hardware's processing power when they are run singularly. Rather, you'll find that a large percentage of the CPU and/or disk drives are idle.
- In order to minimize the amount of idle time (and increase the throughput of your batch processes), we allow you to set up your batch processes so that multiple instances of a given batch job are executed at the same time. For example, in Oracle Utilities Customer Care and Billing when you schedule the **billing** process, you can indicate that multiple parallel instances should be executed (rather than just one instance). You'd do this so that the processing burden of creating bills for your customers can be spread over multiple processes.
- We refer to each parallel execution of a batch process as a "thread".
- Statistics and information messages are displayed in respect of each thread. Why? Because each thread is a separate execution and therefore can start and end at different times.

Refer to Understanding Background Processes for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Batch Run.** This zone provides information about the selected batch process, including a high level summary of key details such as threads status, processing record counts and more.
- **Batch Jobs.** This zone lists one or more batch job submission records that were used to execute the batch run. Typically a batch run is associated with a single job submission record but multiple may exist when some of the threads needed to be restarted. When a single job record exists the batch parameters specified at submission time are listed on the Batch Parameters zone. When multiple jobs exist, select a job to view its parameters.
- **Batch Parameters.** This zone lists the parameters specified as part of a selected batch job submission. This zone is hidden by default when more than one job exists for the batch run.

The following zones may appear as part of the portal's **Threads** tab page.

- **Batch Run Threads.** This zone allows you to filter and review thread instances information. When an instance is associated with messages, you may click on the message count to review the message in the Batch Instance Message zone. You may click on the Thread Summary action link to review a summary by thread.
  About Log files:
  - **Availability**. The ability to download batch log files depends on whether the application properties file has been configured with the location of the log files and the log files associated with the batch run are still available.

Copyright © 2026, Oracle and/or its affiliates.

  - **Security Access.** Only users that have security access to the **Download** access mode for the batch run application service can download log files.
  - **Compression.** The log files are compressed and include a *.gz extension. Different browsers treat this type of file differently. Some browsers may automatically decompress the file as part of the download and they are viewable in any text viewer by changing the extension to one the text viewer recognizes. However, some browsers download the compressed file and a user will need to unzip the file prior to viewing.
  - **Batch Run Logs.** The **Download stdout** and **Download stderr** zone action links may be used to download log files at the batch run level. These files may exist for processing that occurs before the thread execution. The links are visible when the corresponding file is available and the user has security rights to download files.
  - **Batch Thread Logs.** If the log files associated with a batch thread are still available and the user has security rights to download files, the **Download stdout** and **Download stderr** icons are visible for instances of that thread.

- **Batch Instance Messages.** This zone lists any messages reported by a selected thread instance. The zone is hidden by default. Refer to Processing Errors for information about these types of errors.
- **Batch Run Thread Summary.** This zone summarizes thread information across instances based on the "Accumulate All Instances" indication on the batch control record. The zone is visible upon request by using the Thread Summary link on the Threads zone.

The following zones may appear as part of the portal's **Errors** tab page.

- **Batch Instance Messages.** This zone allows you to filter and review messages across all thread instances. Refer to Processing Errors for information about these types of errors.
- **Batch Run To Do Entries.** If the background process has been enabled to create To Do entries for object specific errors, information about the To Do entries are displayed in this tab. This zone displays a summary of all To Do entries created by this batch process, grouped by status and error message. The zone is only visible when such entries exist.
- **Sample To Do Entries.** This zone displays a sample of To Do Entries for a specific status and application message selected in the Batch Run To Do Entries zone. The zone is hidden by default.

The following zones may appear as part of the portal's **Files** tab page.

- **Batch Run Files.** This zone lists file names created by the batch process.

The following zones may appear as part of the portal's **Threads Analysis** tab page.

- **Threads Analysis.** This zone includes several analytical reports around thread performance.

The following zones may appear as part of the portal's **History** tab page. This tab is only available when the maintenance object supports lifecycle definition.

- **Previous 20 Runs - Records Per Seconds.** This zone compares the throughput of the previous 20 runs.
- **Previous 20 Runs - Processed Records.** This zone compares the number of records processed by the previous 20 runs.
- **Previous 20 Runs.** This zone lists the previous 20 runs along with key details about each run.
- **Concurrent Batch Runs.** This zone lists other batch processes that ran at the same time as the current batch run. This information is sourced from the batch snapshot tables and is therefore only available when snapshot data is kept up to date.

Copyright © 2026, Oracle and/or its affiliates.

Actions

The page supports the following special actions:

- **Do Not Attempt Restart.** If the **Run Status** is **Error**, the system will attempt to restart this run when you attempt to execute the Batch Control. In most situations, this is exactly what you want to happen. However, there are rare situations where you do not want the system to execute a given batch run (e.g., if this run is somehow corrupt and you cannot correct the data for whatever reasons). If you want the system to skip the execution of a batch run (and proceed to the next run), click on the **Do Not Attempt Restart** action.
- **Attempt Restart.** Use this action to revert back to the standard restart functionality.

Batch Day Dashboard

This portal provides a high level summary of the status, duration and other details of the batch jobs that ran on a given business date. An implementation may introduce additional zones to this portal as needed.

Use **Menu**, and then **Dashboards**, and then **Batch Day Dashboard** to navigate to the portal.

Batch Runs By Business Day

This zone provides a summary by business date of the number of batch runs completed, in error, and in progress, together with the earliest start time and latest end time of the jobs on that day. sThe zone is sorted in descending order by business date.

> **Note**
>
> For performance reasons, the zone is configured to restrict the number of rows retrieved by the query. The 'Results' number will indicate how many rows have been retrieved. Use the Next and Previous buttons to retrieve additional dates or return to a prior set of dates.

The following filters are provided:

- **Batch Business Date** allows you to select a particular business date.
- **Process Date** allows you to restrict the batch runs included in the summary to those where the processing date is within the start and end dates for the run.
- **Exclude Reruns** allows you to exclude batch reruns from the summary.

> **Note**
>
> The values selected in these filters will be applied as hidden filters to the other zones in the portal.

When a specific business date is broadcast from the summary, the following zones will appear as part of the portal's **Main** tab page.

Batch Day By Status

This zone summarizes the batch runs for the business date by their status, displayed in the form of a pie chart. Selecting a specific slice of data on the chart broadcasts its value to the

Copyright © 2026, Oracle and/or its affiliates.

other zones on the portal allowing them to filter their data accordingly. Use the **Reset** action to reset the filter set by this zone.

Batch Day By Category

This zone summarizes the batch runs for the business date by their batch category, displayed in the form of a pie chart. Selecting a specific slice of data on the chart broadcasts its value to the other zones on the portal allowing them to filter their data accordingly. Use the **Reset** action to reset the filter set by this zone.

Batch Jobs Summary

This zone displays details of the individual batch runs for the broadcast business date. The zone is sorted in ascending order by start date/time and batch control.

> **Note**
>
> For performance reasons, the zone is configured to restrict the number of rows retrieved by the query. The 'Results' number will indicate how many rows have been retrieved. Use the Next and Previous buttons to retrieve additional rows or return to a prior set of rows.

The following filters are provided:

- **Batch Control** allows you to select a particular batch control to review. This filter supports wildcard searching.
- **From Start Date/Time** restricts the batch runs displayed to those where the start date time is on or after this date and time.
- **To Start Date/Time** restricts the batch runs displayed to those where the start date time is on or before this date and time.
- **From End Date/Time** restricts the batch runs displayed to those where the end date time is on or after this date and time.
- **To End Date/Time** restricts the batch runs displayed to those where the end date time is on or before this date and time.
- **Batch Control Type** restricts the batch runs displayed to the selected type.

The following columns are displayed initially:

- **Start Date/Time** is the start time of the run.
- **Batch Control** is the batch control ID of the batch run.
- **Description** is the description of the batch control. Click the hyperlink to navigate to the Batch Control portal.
- **Batch Run** is the run number of the batch job. Click the hyperlink to navigate to the Batch Run portal.
- **Status** is the status of the batch run.
- **End Date/Time** is the end time of the run. The current date and time may be displayed if the batch job is still in progress.
- **Duration** is the time taken for the run, formatted as Days:Hours:Minutes:Seconds.
- **Threads** is the total threads for the batch run.

Copyright © 2026, Oracle and/or its affiliates.

Click the **Column Configuration** icon  in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Batch Category** is the category of the batch run.
- **Batch Control Type** is the type of the batch run.
- **Do Not Attempt Restart** indicates whether the run has been marked not to be restarted.
- **Rerun Nbr** is the rerun number of this batch job.

Requests

Requests enable an implementer to design a process using the configuration tools and standard business object monitoring techniques. They are intended to support use cases where processing should be carried out in background, but is not performed often enough that a dedicated batch process is needed.

An example of such a process might be to send an email to a group of users that summarizes the To Do entries that are assigned to them.

The following topics provide more details of requests and their uses.

- Understanding Requests
- Setting Up Request Types
- Maintaining Requests

Understanding Requests

There are two objects that work together to define a request process:

- The request type which defines configuration related to the request.
- The request record which manages the request process and captures information required to perform the request.

Request Type

The request type business object defines the transactional business object for requests of that type. It may also define parameters to be used by the request, such as to do types for user notification at different points in the request process. In addition, the request type lifecycle may be used to automatically create requests of this type, using a monitor algorithm and the standard monitoring batch control.

Requests

The request object is the object that is responsible for managing a request. A separate request business object must be defined for each use case that follows a unique lifecycle. The business object schema defines the details that are captured during the process. In addition, the business object lifecycle must include at least one algorithm that is responsible for performing the request processing, as an enter plug-in on the appropriate state.

Managing Requests

The product provides a portal for searching and maintaining request objects.

Refer to Maintaining Requests for more information.

Copyright © 2026, Oracle and/or its affiliates.

- Designing a New Request
- Configuring Requests

Designing a New Request

The following sections highlight important considerations in setting up a new Request.

Defining the Request Process Details

When defining a new request object, you must first determine what information needs to be captured to perform the request. Some static details, such as to do types to be used for notifications may be configured on the request type BO. Others, such as the details of the records affected may need to be recorded in the request record. These elements are defined in the BO data area of the request using standard UI hint syntax.

In addition, there may be several steps in processing a request, depending on the complexity of the use case. A request may require approval or review before being processed. These steps are modeled using the request business object lifecycle.

The system provides a 'root' business object for data correction requests (`F1-DataCorrectionRoot`) that may be used as an example of a multi-step process. Your specific product may supply additional request business objects.

Previewing a Request

Users often want to preview a sample of the results of the request before submitting for processing. The system provides the following to support preview functionality.

- A special BO option of **Request Preview Service Script**. This script retrieves the information that is displayed when a user asks for a preview of a request.
- The `F1-DataCorrectionReqActions` map fragment. This map fragment includes support for a preview action using script `F1ReqPrvwAct`. This script invokes the request preview script configured on the request BO, using the preview script schema to generate the UI and display the results. The map also supports hiding the edit and preview actions based on flags set in the request BO.

Performing Request Processing

The assumption is that an enter algorithm is responsible for request processing. It is common for requests to be processed in background rather than real-time. The standard technique is to use a deferred monitor to transition the request to the status that has the algorithm plugged in. The `F1-OrphanRecordDeletion` BO uses this technique.

Request Security

Some users may not be authorized to submit certain requests. When adding a new request, the list of available request types is restricted to types where the user has security access to the related request business object's application service. This may affect the choice of application service and the additional configuration required for a new request business object.

Configuring Requests

If your product supports request use cases that you are planning to implement, all you need to do is add a request type record using the business object for that type and define the appropriate configuration for your business requirements.

The following topics provide information about the request types supplied in the framework.

Copyright © 2026, Oracle and/or its affiliates.

- Orphan Record Deletion
- Configurable Data Correction
- To Do Summary Email

Orphan Record Deletion

The product supports a data correction method to delete orphan records from a given table. Orphan records are child table records where the parent record no longer exists.

The following points describe the functionality at a high level:

- A request is created to capture the table from which the orphan records are to be deleted. The assumption is that prior investigation has identified the target table.
- Back end logic generates the SQL to delete the orphan records and provides counts of the rows to be deleted.
- When the request is submitted, a batch process is started that will run the SQL in background. Backups are taken prior to performing the deletion.
- When the request is completed, a separate request may be created that cleans up the backup tables from the database.

This functionality is supported by configuration and logic supplied in the base business objects Orphan Record Deletion (**F1-OrphanRecordDeletion**) and Orphan Record Deletion Type (**F1-OrphanRecDelType**). The following sections describe key steps in the request lifecycle.

Capturing the Deletion Details

The base product provides a business service **F1-SelectOrphanRecordsSQL** that constructs the SQL to select orphan records from a target table. The business service returns additional SQL statements for the target's child and grandchild tables, if applicable. The product also provides the business service **F1-CountRecords** which counts the orphan records based on the SQL generated by **F1-SelectOrphanRecordsSQL**.

The Orphan Record Deletion BO invokes these business services in a post-processing algorithm and records the details on the request record.

Approving and Submitting the Request

Once a deletion request is captured, there are several key steps to perform.

- Deletion request records are created in **Pending** state so that the SQL statements and record counts can be reviewed before proceeding.
- If the deletion details are correct, the request can be transitioned to **Ready for Approval**. A to do entry is created to notify potential approvers. The request must be approved by someone other than the user who created the record. The notification to do entry type and role are configured on the Orphan Record Deletion Type.
- The request must be approved before it can be submitted for processing. If the Orphan Record Deletion Type indicates that record counts must be verified before deletion, a warning will be issued at approval time if there is a record mismatch.
- Once the request is in the **Approved** state, it can be submitted. The **Submitted** state is configured to be processed by a batch monitor so that record deletion is carried out in background. The monitor batch job is submitted automatically. When record deletion is complete, the request transitions to the **Processed** state so that the results can be reviewed.

Copyright © 2026, Oracle and/or its affiliates.

- If the deletion could not be completed successfully, the request will enter the **Error** state and a to do entry is created. The error to do entry type and role are configured on the Orphan Record Deletion Type.

Completion and Clean Up

The record deletion process includes steps to take backups of the target tables in case the results are not as expected. A separate request may be created to handle deletion of unwanted backup tables at an appropriate time. If the Orphan Record Deletion Type specifies a clean up request type, a request record of that type is created when the deletion request enters the **Complete** state. The clean up functionality is supported by configuration and logic supplied in the base business objects Backup Table Cleanup (**F1-BackupTableCleanup**) and Backup Table Cleanup Type (**F1-BackupTableCleanupType**).

Configurable Data Correction

Implementers can use plug-in driven batches to perform some types of one-off data corrections but the process lacks the built in approval step supported by the request based data correction flows. Moreover, each custom plug-in driven batch involves creating a few custom algorithms, scripts and a batch control, all generated for a one-off update.

The product supports a simplified way to request, approve and perform plug-in driven based data corrections.

The following points describe the functionality at a high level:

- A data correction request is created to capture the following details:
  - An **SQL** query that selects the records to correct.
  - Either a **business object** or a **service script** to perform the data correction for each selected record.
  - The number of threads to submit the data correction batch process with.
- When the request is approved and a user submits it for processing, a batch job is created for the specific request. The batch job uses the query on the request to select the records to correct and calls the corresponding business object or script to perform the update for each record. The request remains in the 'Submitted' status while the batch process runs.
- When the batch completes, it updates the corresponding request to the 'Processed' status, which then creates a monitoring To Do Entry for a user to review the results of the batch. The user marks the request as complete and if further corrections are needed the user can create another request with different correction details as needed.

The following sections provide more information about this tool.

Request Type Setup

A "Data Correction via Batch" (**F1-DataCorrectTypeByBatch**) Request Type business object is provided to define a request type record for creating such requests. To enable this functionality, a request type should be created using this business object.

The request type should reference the "Configurable Data Correction" (**F1-DataCorrectPlugIn**) as the request business object. This business object is a subclass of the data correction root business object and as such leverages the same approval based lifecycle support.

Data Correction Implementation

The following is a suggested description for a one-off data correction procedure:

Copyright © 2026, Oracle and/or its affiliates.

- If the change is not a simple update of fields using a business object, implement a service script that gets the prime key of a selected record and uses standard scripting logic to perform the update. The script can be tested using the Test action on the Script page.
- Use any tool to compose and test an SQL that selects the records from a primary table of a maintenance object to correct.
- If the change is a simple update of non-list fields on a business object, note the xpath of the primary key fields and the xpath of each element that needs to be updated on that business object. Adjust the SQL to set the xpath element name as the alias of each field.

  This is an example of a data correction SQL that selects some batch control records and assigns each with a new "next batch run number". The "batchControl" and "nextBatchNumber" aliases are xpath references to the corresponding fields in the F1-BatchControlPhysicalBO business object.

  ```
  select batch_cd "batchControl",
  case
  when batch_cd = '...' then <new next batch run number>
  ...
  else ...
  end "nextBatchNumber"
  from ci_batch_ctrl
  where batch_cd in ('.... list of batch control records to update ... ')
  ```

  > **Note**
  >
  > The SQL is provided for illustration purposes only.

- If the change is not as simple, implement a service script that gets the fields from a selected record and uses standard scripting logic to perform the update.
  - The script's schema must include the **F1-DataCorrectionProcessRecord** data area. It is via this data area that your script gets the selected field from each row to update. The structure of this data area is similar to the plug-in driven **Process Record** plug-in spot.

    All selected fields are provided to your script in the 'selectFields' list of the data area. In addition, the primary key of the selected record may also be provided via the 'mo', and 'pk1-5' elements, if the 'MAINT_OBJ_CD', and 'PK_VALUE1-5' designated aliases are used in your query.

    For example:

    ```
    select 'F1-FACT' MAINT_OBJ_CD, fact_id PK_VALUE1, ...
    from f1_fact ...
    ```

  - The script can be tested using the 'Test' action on the Script page.
- Create a 'Configurable Data Correction' Request using the corresponding Request Type. Specify on the request the SQL and the business object or script to use for data correction.
- Progress the request through approval and submit for processing when ready.
- Review the data once the submitted batch is complete.

Approving and Submitting the Request

Once a data correction request is captured, there are several key steps to perform.

Copyright © 2026, Oracle and/or its affiliates.

- If the deletion details are correct, the request can be transitioned to **Ready for Approval**. A to do entry is created to notify potential approvers. The request must be approved by someone other than the user who created the record. The notification to do entry type and role are configured on the Request Type.
- The request must be approved before it can be submitted for processing.
- Once the request is in the **Approved** state, it can be submitted. The **Submitted** state is configured to be processed by a plug-in driven batch process that uses details from the request to perform the deletion in background. The batch job is submitted automatically. When batch job completes, the request transitions to the **Processed** state so that the results can be reviewed.
- A user marks the request as complete and if further corrections are needed the user can create another request with different correction details as needed.

To Do Summary Email

The base product includes a sample request process that sends an email to users that have incomplete To Dos older than a specified number of days.

Configuring the Email Request Type

The following configuration tasks are required to use this process:

- Define an Outbound Message Type. The business object usually defined for the Outbound Message Type is **F1-EmailMessage.**
- Define an External System that contains the Outbound Message Type in one of its steps. In the configuration determine if the communication is through SOA, batch, or real-time processing method when sending the email notification. Refer to Outbound Messages for more information about configuration needed for the different processing methods.
- Create a Request Type using the `F1-TodoSumEmailTyp` business object.
- Configure the Request Type with the outbound message type and external system.
- Provide values for the threshold, email subject and email address.
- If you want emails of this type to be sent periodically, enter a value for the frequency parameter.

Previewing the Email Request

When adding or editing a To Do email request, the following actions are possible:

- Click **Save** to submit the request.
- Click **Cancel** to cancel the request.
- Click **Preview** to see a sample of records that satisfy the selection criteria for this request. This information is displayed in a separate map. In addition, the map displays the total number of records that will be processed when the request is submitted. From this map you can click **Save** to submit the request, **Back** to adjust the parameters, or **Cancel** to cancel the request.

Submitting the Email Request

When an email request is saved, the job is not immediately submitted for real time processing. The record is saved with the status **Pending** and a monitor process for this record's business object is responsible for transitioning the record to **Complete**.

Copyright © 2026, Oracle and/or its affiliates.

As long as the record is still **Pending**, it may be edited to adjust the parameters. The preview logic described above may be repeated when editing a record.

The actual work of the request, such as generating an email, is performed when transitioning to **Complete** (using an enter processing algorithm for the business object).

Setting Up Request Types

This portal is used to view and maintain request types.

You can access this portal from **Admin**, and then **General**, and then **Request Type**. You are brought to a query portal with options for searching for a specific request type. Once a request type has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to the embedded help text for more information on the Request Type attributes.

Refer to Understanding Requests for background information about defining request types.

Maintaining Requests

This portal is used to view and maintain requests.

You can access this portal from **Main Menu**, and then **Tools**, and then **Request**. You are brought to a query portal with options for searching for a specific request. Once a request has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Requests for more information.

Attachments

Some implementations may require that attachments be available from the application. These attachments can be stored in the Attachment table and then linked to other objects if applicable.

- Attachment Overview
- Configuring Your System for Attachments
- Maintaining Attachments

Attachment Overview

The following topics provide additional information regarding attachment functionality.

Attachment Types

The system supports several different attachment content types, for example:

- PDF Document
- Excel Spreadsheet
- Jpeg Image
- Text Document

Copyright © 2026, Oracle and/or its affiliates.

The attachment data itself may be text or binary. When storing the data in the application however, it is stored as text information only. As a result, the upload of an attachment that is a binary type requires a conversion prior to storing the data. When viewing the attachment, the data is converted again for display.

The system also supports attachments that reference a URL to a document hosted in an external system. When viewing the attachment, the URL is launched to access the related document in the external system.

Each type of attachment is defined using an attachment business object. The business object includes configuration defining the supported file extensions, whether the data is binary or not and the content type that represents the type of data for the attachment.

> **Note**
>
> To view the attachment business objects provided with the base product, navigate using **Admin**, and then **System**, and then **Business Object**, and then **Search** and search for business objects related to the Maintenance Object for Attachments (`F1-ATCHMT`). You may set any of these business objects as not allowing new instances to prevent their use for creating attachments.

Owned Attachments

Attachments can be either ‘owned’ or ‘common’. An owned attachment is one that is related to a specific record. For example, the specific test results for a given device can be uploaded and linked to that device or to its test records. These types of attachments are typically uploaded and maintained via the object that owns it.

Common Attachments

Common attachments are ones that are uploaded independent of any transaction in the system. They can be used for general system or company information. Or they can be linked to more than one transaction. For example, instructions for performing a certain type of task can be uploaded as an attachment and linked to a task type where those instructions are relevant. These types of attachments are uploaded and maintained in the central Attachment portal. Objects that may refer to the attachments may link the attachments via characteristics or some other appropriate mechanism.

Content Location

An attachment's content may reside in either of the following locations:

- **Database.** The content is stored in the database as part of the attachment record. This is also the default location when uploading a file that is not too large.
- **Object Storage.** When an uploaded file is too large to be stored in the database its content is stored in Object Storage. Refer to the "Uploading a Large Attachment File" section for more information.
- **External URL.** An attachment may reference a URL to a document hosted by an external system.

> **Note**
>
> To reference an existing Object Storae file use the file storage syntax.

Copyright © 2026, Oracle and/or its affiliates.

When sending an attachment to an external system via an outbound message or including it in an email, the attachment's content can only be embedded in the message when the content resides in the database. For all other content locations, an external facing URL to view the attachment is provided instead.

Uploading a Large Attachment File

By default, an attachment is uploaded and stored in the database along with its entire content. The process of uploading very large content like videos etc. is often limited by memory and other resource constraints. You may configure a threshold above which content is stored in object storage and a link to the file is kept on the attachment record.

To enable this feature refer to the **attachment location** and related options in the **General System Configuration** feature configuration for more information. In a cloud installation, this configuration is required to upload large attachments due to stricter resource limits.

The following are known limitations with storing content in object storage:

- The responsibility for data recovery of attachment content shifts to the customer as part of their object storage files management.
- Custom logic if any that relies on attachment content to exist in the database will not work for attachments stored in object storage.

Emailing Attachments

The system supports a business service that may be used by system processing to send an email. The business service `F1-EmailService` supports receiving the IDs of one or more attachments as input parameters. Refer to Sending Email for more information.

By default, an attachment's content is embedded in the email message when the content resides in the database. For all other content locations, an external facing URL to view the attachment is provided instead. When the content of the attachment is too large to be embedded in the email, a link is provided instead. Refer to the **Maximum Email Attachment Size** in the **General System Configuration** feature configuration for more information.

External Reference

Attachment records may be added from an external system in which case, the external system could have a reference ID to capture. The attachment allows for including an external reference ID when adding an attachment internally.

Malware Scan

The system provides a plug-in spot where on premises implementations can configure a step to scan attachment for malware. If an algorithm is provided, it is called every time attachment content is added. This is not supported in cloud implementations.

Configuring Your System for Attachments

In order to link attachments to objects in the system, there may be some configuration or implementation required to support the link. It is possible that one or more objects in your product already support attachments out of the box. Consult the product documentation for the specific object for confirmation. For objects in the system that do not support attachments out of the box, the following sections provide some guidelines for enabling support for attachments. Contact product support for more information.

Copyright © 2026, Oracle and/or its affiliates.

Supporting Common Attachments

The attachments themselves are created / uploaded using the attachment portal. Refer to Maintaining Attachments for more information.

If your implementation has a use case where one or more common attachments may be linked to an object (and the object does not already support this functionality), the object may need to be extended to capture the attachments.

- If the object includes a characteristic collection, this is a recommended way to capture attachments. A characteristic type should be defined for each type of attachment. The characteristic type should be a foreign key type and should reference the Attachment FK reference. The characteristic entity collection should include the object that the common attachment will be linked to.
- Most characteristic collections are sequence based characteristics and would support multiple entries for the same characteristic types, if multiple attachments are applicable.
- If the object to support the attachments is governed by a business object, the implementation must extend the business object to define one or more appropriate elements used to capture the attachments. If only one attachment of a certain type is allowed, a single flattened characteristic may be used. If multiple attachments of a certain type are allowed, the BO schema may define a “flattened list” exposing the sequence and the characteristic type.
- If the object is maintained on a “fixed page” with a generic characteristic collection, no additional configuration is needed to allow users to link attachments to that object.

Supporting Owned Attachments

When creating an attachment for a specific record, the attachment itself captures the information about the related record, namely its maintenance object code and its primary key. For these types of attachments, no configuration is needed on the related business object to capture the attachments, as was the case with common attachments.

However, it is recommended to configure the user interface of the related object so that the owned attachments can be viewed and maintained from that page. To do this, you may use the generic attachment zone provided by the product: `F1-ATTCHOWN`.

Note that your product may already have support for viewing and maintaining owned attachments on one or more if its base delivered portals.

Configuring Size Limit

The system allows for configuration of a size limit for uploading attachments. This is optional but the recommendation is to set this to a reasonable value so as to limit the ability to upload unreasonably big files. To configure this, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type **General System Configuration**. If one does not exist, create a feature configuration of this type. Select the **Maximum Attachment Size** option and define an appropriate value.

You may also configure the system to upload very large attachments to Object Storage. To enable this functionality, set the **Attachment Location** option in the same feature configuration to define the file location in object storage. Configure the **Attachment Threshold Size** option to set the size limit above which attachment data is stored in object storage as files and not in the database.

If your organization performs many concurrent attachment uploads, you may optionally set the **Maximum Attachment Aggregate Size** option to override the system limit as needed.

Copyright © 2026, Oracle and/or its affiliates.

When the content of the attachment is too large to be embedded in an email, an external facing URL to view the attachment is provided instead. Refer to the **Maximum Email Attachment Size** in the **General System Configuration** feature configuration for more information.

Defining a New Attachment Type

As mentioned, the product provides support for several content types. If your implementation needs to support attachments for a content type not currently supported, create a new business object copying the configuration of an existing attachment business object.

Configure the following option types for the BO:

- **Binary** indicates whether the attachment data must be converted from binary format. Binary attachments are stored in the database as text, and are then converted back to the original format when retrieved.
- **Content Type** represents the browser's mime type of the attachment.
- **Supported File Extension** specifies the valid file extensions for the content type.

Once the business object is defined, it is ready for use.

Supporting File Name Exceptions

By default, the system prevents attachment file names from having special characters. If there is some reason that an implementation needs to define an attachment with special characters, there is a system property setting that may be configured to relax this validation. Contact your system administrator for support.

Enabling Malware Scan

> **Note**
>
> This functionality is only available for on premises implementations. It is not supported for cloud implementations.

The product provides a malware scan algorithm in the base product that uses ClamAV. ClamAV itself is not provided as part of the product. To use the base algorithm, your implementation needs to get ClamAV and define system properties for the host and port where ClamAV may be called. The algorithm, called `F1-ATCHM-WS` should be plugged into the installation options - algorithms collection using the system event **Attachment Malware Scan**

If your implementation wants to use a different algorithm for malware scanning, you may deliver your own algorithm and plugit into the installation algorithm collection.

Maintaining Attachments

This section describes the functionality supported for viewing and maintaining attachments.

Navigate using **Admin**, then **General**, the **Attachment**. You are brought to a query portal with options for searching for common attachments.

Once an attachment has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The base search options for the attachments query only support searching for common attachments. Owned attachments may also be viewed on the attachment maintenance portal, but a user may only drill into the attachment maintenance from the maintenance portal of the "owning" entity.

The Attachment zone provides basic information about an attachment, including the ability to upload the file and to view an uploaded file. Viewing an attachment will typically launch a download of the attachment file, following standard browser behavior.

- Adding Attachments

Adding Attachments

Common attachments may be added from the attachments portal (or via the standard menu path). In addition, your product may support attachments associated with specific records ("entity owned attachments") which may also provide the capability to add attachments.

In both cases, when adding an attachment, you are prompted for either a file to upload or an external URL to a document that resides in another system. When uploading a file then once a file is chose, the system determines the appropriate business object to associate with the attachment based on the file extension. Typically one and only one business object is found at which point you are prompted to provide the Attachment Name. (Your specific product may also require additional information at this time). Fill in the details and save.

Please note the following:

- If no business object is found for the uploaded file's file type, an error is issued. This type of file is not currently supported as an attachment.
- If multiple business objects are found, the user must choose the appropriate one. This should be rare.
- The ability to add URL based attachments is enabled by default but may have been disabled by your organization.
- The attachment's file extension is validated against the base delivered allow list F1-FileExtensionAllowList, defined as a managed content record.
- The system ensures that the attachment's file name does not contain special characters and uses only standard UTF-8 supported characters.

Monitoring and Analytics

This chapter describes various tools provided in the product to support system monitoring including tools related to reporting, monitoring and analytics.

- Understanding Batch Analytics
- Reporting Tool Integration
- Health Assessment Options
- Service Health Check
- Measuring Performance
- Capturing Statistics

Copyright © 2026, Oracle and/or its affiliates.

- Capturing Product Use Metrics

Understanding Batch Analytics

This section explains how to use batch analytics for strategic analysis of jobs, threads, and workload performance.

- Batch Job Analytics
- Batch Thread Analytics
- Batch Workflow Analytics

Batch Job Analytics

The Batch Job Analytics portal illustrates a set of metrics aligned with execution of batch processes at the job level. This portal is useful for summarization of the state and work performed by a batch control over a selected date range.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BRANSN - Batch Run Analytics Snapshot Update.**.

> **Note**
>
> Not all zones in this portal support broadcast as it is not appropriate in all situations. Unless indicated assume broadcast is not available from the zone.

> **Note**
>
> At the top of each tab in this portal, the selected batch control and date range is shown.

Batch Job Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Job Analytics search from the search dropdown. To help find the necessary batch control, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.

Copyright © 2026, Oracle and/or its affiliates.

Once the appropriate filters are populated a list of batch controls is returned with the number of executions in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired batch control from the list.

Batch Job Analytics - Main

The Main section of the analytics summarizes the state, submission source, performance, and volumes processed by the desired batch control over a date range. This includes analysis in the following areas:

- The Batch Job Status Summary zone summarizes the state of all executions of the desired batch control over the desired date range within the snapshot. In an ideal situation this graph should show a high percentage of complete (successful) executions.
- The Batch Job Submission Summary zone summarizes the method used to submit the desired batch control over the desired date range within the snapshot. In an ideal situation, batch jobs are submitted the using a small amount of methods to maximize efficiency.
- The Batch Job Elapsed Time Summary zone shows the minimum (best) elapsed time, average elapsed time and maximum (worst) elapsed time for the desired batch control over the desired date range within the snapshot. All elapsed times are shown in seconds, with the hover text translates the time into DD:HHMI:SS format. In an ideal situation, all the values should be relatively the same. Large deviations in values may be due to volume or inefficiencies.
- The Batch Job Volume Summary zone shows the minimum (lowest) volume, average volume and maximum (highest) volume for the desired batch control over the desired date range within the snapshot. Volume is expressed as the total number of objects processed by the job. In an ideal situation, all the values should be relatively the same. Large deviations in values may be due to business fluctuations or inefficiencies.

Batch Job Analytics - History

The History section of the analytics shows detail information for job level elapsed time and volumes for the desired batch control over the desired date range within the snapshot.

Each analysis includes additional reference information such as the average value and the confidence interval to use. Values above the confidence interval should be considered as outliers and require further analysis.

This includes analysis in the following areas:

- The Batch Job Elapsed Time History zone shows the job level elapsed time, in seconds (the hover text translates the time into DD:HHMI:SS format) for each batch run for the desired batch control over the desired date range within the snapshot. This can highlight fluctuations in elapsed time of data processed by the batch job over time.
- The Batch Job Volume History zone shows the total number of objects processed by each batch run for the desired batch control over the desired date range within the snapshot. This can highlight fluctuations in volumes of data processed by the batch job over time.

Batch Job Analytics - Submission

The Submission section of the analytics shows analysis of submission information periods applicable to the desired batch control over the desired date range within the snapshot.

This includes analysis in the following areas:

- The Batch Job Submission History zone shows the number of submissions per day over the for the desired batch control over the desired date range within the snapshot. This can

Copyright © 2026, Oracle and/or its affiliates.

highlight fluctuations in how often the desired batch control is submitted. Ideally the number of submissions should be consistent.

- The Batch Job Submission By Day Of Week zone is an analysis of the total submissions by each day of the week to help detect any fluctuations in submission patterns for the desired batch control over the desired date range within the snapshot. Ideally the number of submissions should be consistent.
- The Batch Job Submission By Hour Of Day zone is an analysis of the total submissions by each hour for the desired batch control over the desired date range within the snapshot. This zone is provided to help detect any fluctuations in submission patterns. Ideally the number of submissions should be consistent.
- The Top Users Submitted Job zone summarizes the top 20 users that have submitted the desired batch control over the desired date range as expressed in the snapshot. The total number of submissions by those users is shown for the desired batch control over the desired date range within the snapshot. This zone is primarily provided to help detect unusual patterns by individual users as necessary.

Batch Job Analytics - Detail

The Detail section of the analytics lists all batch runs for the desired batch control over the desired date range as expressed in the snapshot. This represents the data that was used as part of the analysis used in the previous sections. It is possible to navigate to the Batch Run Portal for individual runs for further information. This list can be further filtered for status and submission method used.

Batch Thread Analytics

The Batch Thread Analytics portal illustrates a set of metrics aligned with execution of batch processes at the thread. This portal is useful for summarization of the state and work performed by the threads for a batch control over a selected date range. This portal is useful for jobs that are run multi-threaded.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BTANSN - Batch Thread Analytics Snapshot Update.**.

> **Note**
>
> Not all zones in this portal support broadcast as it is not appropriate in all situations. Unless indicated assume broadcast is not available from the zone.

> **Note**
>
> At the top of each tab in this portal, the selected batch control and date range is shown.

Copyright © 2026, Oracle and/or its affiliates.

Batch Thread Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Thread Analytics search from the search dropdown. To help find the necessary batch control, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.

Once the appropriate filters are populated a list of batch controls is returned with the number of executions and the maximum thread limit used in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired batch control from the list.

Batch Thread Analytics - Main

The Main section of the analytics summarizes the state, submission source, performance, and volumes processed by the desired batch control over a date range. This includes analysis in the following areas:

- The Batch Thread Status zone summarizes the state of all executions of the desired batch control over the desired date range within the snapshot. In an ideal situation this graph should show a high percentage of complete (successful) executions.
- The Batch Thread Reliability Summary zone calculates the total reliability rating for the desired batch control over the desired date range within the snapshot.
- The Batch Thread Limit History zone shows the thread limit specified for each batch run for the desired batch control over the desired date range within the snapshot. Ideally the thread limit used should be consistent. Large deviations in values may be due to volume or inefficiencies.

Batch Thread Analytics - Reliability

The Reliability section of the analytics focuses on the reliability of individual job runs for the desired batch control over the desired date range within the snapshot.

The analysis of threads introduces a reliability rating related to the success rate of threads for an execution. The percentage of successful threads over total thread count for each execution is used for the rating using the following criteria:

- If all threads (100%) are successful for a batch run then the rating is **High**.
- If the thread failure rate is less or equal to 1% of the total number of threads then the rating is **Good**.
- If the thread failure rate is less or equal to 5% of the total number of threads then the rating is **Fair**.
- If the thread failure rate is above 5% of the total number of threads then the rating is **Poor**.

The reliability analysis includes analysis in the following areas:

- The Batch Thread Reliability History zone shows the reliability rating for each run for the desired batch control over the desired date range within the snapshot. The color and size of the entries in the graph illustrate the reliability and percentage of successful threads.

Copyright © 2026, Oracle and/or its affiliates.

- The Batch Thread Reliability Details zone lists the individual batch runs with their associated reliability rating for the desired batch control over the desired date range within the snapshot. You can navigate to the Batch Run Portal for additional information.

Batch Thread Analytics - Differential

The Differential section of the analytics shows analysis of submission information periods applicable to the desired batch control over the desired date range within the snapshot.

This includes analysis in the following areas:

- The Batch Thread Time Differentials zone shows the difference between start and end time for the desired batch control over the desired date range within the snapshot as expressed as a wost case differential. Each differential is shown in seconds. If threads start at the same time and end at the same time then the differential is zero (0). Positive differentials mean differences in start and end times of threads.
- The Batch Thread Volume Differentials zone shows the difference between the smallest volume and largest volume for the threads within each run against the average volume. Each differential is shown as number of records that are different compared to the average volume. A reference line is shown to show the relative difference in low and end volumes. In an ideal situation each thread should work on the same amount of records. Variations means variance in processing volume and may result in different thread elapsed times.

Batch Thread Analytics - Retries

> **Note**
>
> If this zone is blank, then there were no automatic retries for the desired batch control over the desired date range within the snapshot.

The Retries section of the analytics lists the total retries per batch run. If the number of retries is above zero, in a consistent way, then there may be some underlying issues that need investigation.

> **Note**
>
> Retries are only performed for specific conditions. Refer to the cloud service documentation for additional information.

Batch Thread Analytics - Zero Processing

> **Note**
>
> If this zone is blank, then there were no zero processing threads found for the desired batch control over the desired date range within the snapshot.

While most threads have records to process, it is possible that a thread is allocated no work to do. This is inefficiency as resources are allocated to start an end the thread at a minimum. This zone highlights the number of threads per batch run with zero processing for the desired batch control over the desired date range within the snapshot. If the number is consistently high then consider altering the thread limits or thread numbers to re-balance the workload.

Copyright © 2026, Oracle and/or its affiliates.

Batch Thread Analytics - Interquartile Range Analysis

Interquartile analysis is a statistical method used to understand the spread and distribution of a set of data by focusing on the middle 50% of values. By dividing the data into four equal parts, it helps identify the range where most data points fall, making it easier to spot any unusual or extreme values (outliers). This approach is helpful for quickly assessing data consistency and detecting trends, enabling you to make more informed decisions based on typical, rather than exceptional, results.

The Batch Thread Inter Quartile Analysis zone shows the inter-quartile analysis for threads across batch runs for the desired batch control over the desired date range within the snapshot.

To understand the analysis there are different metrics calculated to provide reference:

- The **median quartile** represents the median between the worst and best thread elapsed time. This metric is calculated to provide a reference to assess other metrics against. In an ideal situation all the threads would have this elapsed time.
- The **upper quartile whisker** and **lower upper quartile whisker** are calculated at the 75% and 25% confidence interval, respectively, from the median. These whiskers represent the acceptable extremes for elapsed times of threads. There proximity to the median illustrates whether the threads are finishing
- The **high within quartile** and **low within quartile** calculates the thread elapsed time within the whiskers. The proximity of these metrics to the whiskers indicates if threads are ending early and/or late consistently.
- The **worst thread elapsed time** and **best thread elapsed time** are used as reference for all the above metrics. If these values are above their respective whiskers then they are consider outliers
- The difference between the whispers is known as the **Inter Quartile Range (IQR)**. The higher the IQR value the more the variation in the thread elapsed times. If these is little variation, then these values are close to the median.

It is divided into three types of analysis, as different graphs on the zone:

- The first analysis is focusing on the upper quartile against the worst performing threads. The relationships between these metrics will illustrate the extent of any variation in the threading and the potential presence of threads that overrun.
- The second analysis is focusing on the lower quartile against the best performing threads. The relationships between these metrics will illustrate the extent of any variation in the threading and the potential presence of threads that finish too early.
- The last analysis is focusing on the IQR value. If you see a small IQR, it means thread execution times are consistent. A large IQR suggests there is significant variation.

Batch Workflow Analytics

The Batch Workflow Analytics portal illustrates a set of metrics aligned with execution of batch processes at the workload level. This portal is useful for ;looking at workload at the threadpool level to determine load profiles over a particular day for a particular threadpool.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BTANSN - Batch Thread Analytics Snapshot Update.**.

> **Note**
>
> Workload Analytics use a "workload minute" to measure capacity. Each thread running in any minute uses one workload minute. If multiple threads run in the same minute, each one adds to the total. This makes it easy to see how much capacity is used during a day, since the maximum available is the number of allowed threads multiplied by 1,440 minutes (the total minutes in a day) for each thread pool.

Batch Workload Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Workload Analytics search from the search dropdown. To help find the necessary threadpool and date, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.

Once the appropriate filters are populated a list of threadpools and dates is returned with the total workload minutes for the combination of threadpool and date in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired threadpool and date combination from the list.

Batch Workload Analytics - Main

The Main section of the analytics summarizes workload minutes for the desired threadpool and desired date for every two minute interval. This zone shows how busy the desired threadpool was at any point in the desire date. You filter the information with additional filters as necessary.

Clicking on any point in time will broadcast the time interval and the Batch Workload Detail will show the batch jobs and threads that were started, running and/or complete in that interval. You can navigate to the Batch Run Portal for additional information.

Batch Workload Analytics - Summary

The Summary section of the workflow analytics provides summary level analytics related to the total workload minutes in the desired threadpool and desired date. This includes analysis in the following areas:

- The Batch Workload Summary by Category zone summarizes total workload minutes by batch category. This can be used to appreciate the distribution of workload for the desired threadpool and desired date.

Copyright © 2026, Oracle and/or its affiliates.

- The Batch Workload Summary by Submission zone summarizes total workload minutes by submission type. This can be used to appreciate the distribution of how the workload is submitted for the desired threadpool and desired date.

Batch Workload Analytics - Hour Analysis

If the minute by minute analysis is too detailed, the Batch Workload Hour Analysis zone shows workload by hour of the day for the desired threadpool and desired date.

Reporting Tool Integration

This section describes how to configure your third party reporting tool and how to define your reports in the system to enable users to submit reports online.

- The Big Picture Of Reports
- Configuring The System To Enable Reports
- Sample Reports Supplied with the Product
- How To Define A New Report

The Big Picture Of Reports

The topics in this section describe the approach for designing and defining your system reports. Note that the product includes an out-of-the-box integration with BI Publisher. However it is possible to use the reporting objects to integrate with a different third party tool.

- Integration with BI Publisher
- How To Request Reports
- Viewing Reports

Integration with BI Publisher

Your DBMS, your product, and BI Publisher can work together to produce reports. You may choose to use a different reporting tool, but this may not be a trivial effort. This section provides high-level information about some of the business requirements that are being solved with the reporting solution.

Multi-Language and Localization Support

The integration supports a multi-language implementation and supports different localization settings.:

- All labels, headings and messages are defined using field and message meta-data in the application, which support multiple languages.
- The appropriate font, size, and layout are based on the requested report and the user's language.
- Dates and numbers are formatted as per the user's display profile.
- Currency based numbers are formatted as per the currency definition from the product
- Requesting Reports from The System
- Overview of the Data - BI Publisher

Copyright © 2026, Oracle and/or its affiliates.

Requesting Reports from The System

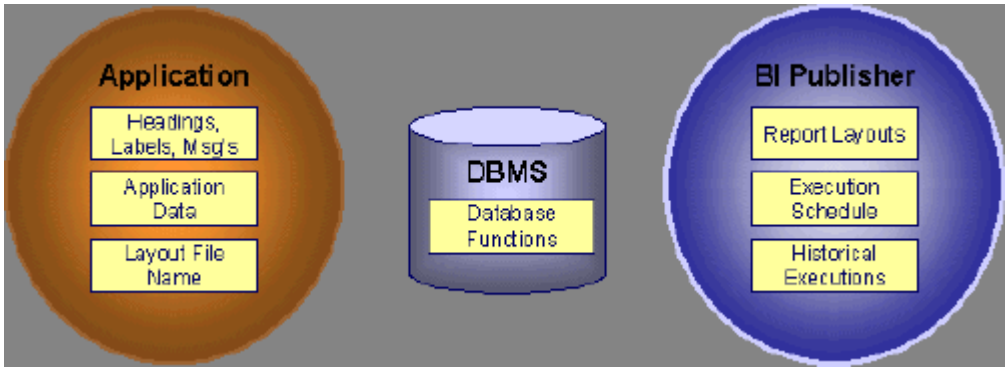
Although reports are rendered in your reporting tool, users must be able to request ad-hoc reports from within the system (assuming users have the appropriate security access).

- The prompts for the input parameters must be shown in the user's language
- Users should be able to use the standard search facilities to find parameter values
- Plug-ins can be optionally used to cross-validate input parameters
- Application security must authorize ad-hoc report requests

Overview of the Data - BI Publisher

The following diagram provides an overview of where data is stored for your reports for integration with BI Publisher.

Figure Application and BI Publisher



The application contains:

- The application data that appears on your reports.
- The language-specific headings, labels and messages on your reports.
- The layout file name to be used for the report.

BI Publisher contains:

- How your reports look.
- Information about scheduled reports and reports that have already run.

The DBMS contains the SQL used to retrieve the data on your reports (residing in database functions).

> **Note**
>
> BI Publisher can be configured to retrieve data via a service call. Because every business object can be read via a service call, elements that reside in an XML based column can be displayed on reports produced using BI Publisher. See your product's *Installation Guide* or *Optional Products Installation Guide* for information on this configuration.

Copyright © 2026, Oracle and/or its affiliates.

How To Request Reports

A user may request an ad hoc report from within your product:

- A report submission page enables a user to choose the desired report and enter the parameter values for the report
- The user must be granted security access to the report
- The request is passed to the reporting tool real time. Refer to Configure The System to Invoke BI Publisher for more information.
- The reporting tool generates the report and displays it in a new browser window

The reporting tools' scheduler creates reports (as per your schedule). This function is entirely within the reporting tool. No scheduling functions reside within your product.

A user can request an ad-hoc report from within the reporting tool. Note, the user's ID must be supplied as a parameter to the report in order for the user's profile to be used to format dates and numbers

Viewing Reports

As described above, ad-hoc reports requested from within your product are displayed immediately after they are generated in a new browser window

If your reporting tools supports it, the Report History page may be configured to open your tool's report execution history page and request a view of this report.

> **Note**
>
> The Report History page currently does not display historical reports for BI Publisher.

Configuring The System To Enable Reports

- Configuring BI Publisher Reports
- Defining Reporting Options
- Defining Report Definitions

Configuring BI Publisher Reports

This section contains topics specific about configuring the product to interoperate with BI Publisher.

- Configure the System to Invoke BI Publisher Real-time
- Batch Scheduling in BI Publisher

Configure the System to Invoke BI Publisher Real-time

The base product provides an installation algorithm plug-in spot called Reporting Tool. This plug-in spot should contain an algorithm that invokes the third party reporting tool real-time.

For BI Publisher, the system provides an algorithm type called `F1-BIPR-INV`, which invokes BI Publisher.

Copyright © 2026, Oracle and/or its affiliates.

These algorithms rely on information defined in the Reporting Options table: the reporting server, reporting folder and the user name and password for accessing the reporting tool. The values in the reporting options should have been set up when the system was installed. Contact your system administrator if there are any problems with the values defined on the reporting options.

To use the algorithm types to invoke BI Publisher, perform the following steps:

- Create an algorithm for the appropriate algorithm type.
- On the installation options, add an entry to the algorithm collection with an algorithm entity of **Reporting Tool** and indicate the algorithm created in the previous step.

Batch Scheduling in BI Publisher

For many of your reports, you probably want the report to be produced on a regular basis according to a scheduler. The reporting solution relies on the BI Publisher software to provide the batch scheduler functionality. Refer to BI Publisher documentation for details about configuring the batch scheduler.

Defining Reporting Options

The reporting options are provided as a mechanism for defining information needed by your reporting solution. The base product uses the reporting options to define information needed to access the reporting tool from within the system using the algorithm defined on the installation option.

Navigate to this page using **Admin**, and then **Reporting**, and then **Reporting Options**.

Description of page

Reporting Folder defines the shared folder where reports are stored.

Reporting Server From Browser defines the URL of the web application where the reporting tool is installed. For example, using BI Publisher, the format is: `http://<BI Publisher Server>:<port>`.

> **Note**
>
> **Customize Options**. The reporting options are customizable using the Lookup table. This field name is `RPT_OPT_FLG`. The reporting options provided with the system are needed to invoke the reporting tool. If your implementation requires other information to be defined as reporting options, use the lookup table to define additional values for the reporting option flag.

Where Used

This information is used by the reporting tool algorithm on the installation option to invoke the reporting tool software.

Implementations may use reporting options to record other information needed for their reporting tool.

Defining Report Definitions

For each report supplied by your installation, use the report definition page to define various attributes of the report.

Copyright © 2026, Oracle and/or its affiliates.

- Report Definition - Main
- Report Definition - Labels
- Report Definition - Parameters

Report Definition - Main

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition**.

> ⚠ **Caution**
>
> Important! If you introduce new report definitions, you must prefix the report code with **CM**. If you do not do this, there is a slight possibility that a future release of the application could introduce a new system report with the name you allocated.

Description of page

Enter an easily recognizable **Report Code** and **Description** for each report. Use the **External Reference ID** to define the identifier for this report in your external reporting tool.

Define an application service to enable users to request submission of this report online or to view report history for this report. Once you define an application service for each report, use Secured Objects and Application Services to define which users may access this report.

> ⓘ **Note**
>
> **Access Mode**. The access mode for application services related to reports must be set to **Submit/View Report**.

If you have more than one parameter defined for your report and you wish to perform cross-validation for more than one parameter, provide an appropriate **Validation Algorithm**.

Enter a **Long Description** to more fully describe the functionality of this report. This information is displayed to the user when attempting to submit the report online or when viewing history for this report.

For BI Publisher, if you want to use one of the sample reports provided by the system, but with a different layout, indicate the layout to use for the report in the **Customer Specific Font/ Layout** field and BI Publisher uses this information instead. The name for base report layout is <report code>_Base. For example, a base layout for CM_TODO is named CM_TODO_Base.

Report Definition - Labels

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition** and go to the **Labels** tab.

> ⓘ **Note**
>
> **Company name and logo.** Note the company name used as a title in the sample reports is defined as a message on the installation options. For information about installing the company logo, refer to your product's *Installation Guide* or the *Optional Products Installation Guide*.

Copyright © 2026, Oracle and/or its affiliates.

Description of Page

In order to provide multi-language capability for each report, the labels used for the report must support multiple language definitions. For each label used by your report, indicate a unique **Sequence** and the Field used to define the **Label**. The label defined here should be the same label that is defined in your report layout defined in the external reporting tool.

When rendering an image of the report, the external reporting tool retrieves the appropriate label based on the language used for the report.

Report Definition - Parameters

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition** and go to the **Parameters** tab **.**

Description of Page

The **Parameters** scroll contains one entry for every parameter defined for the report. The following fields display:

Parameter Code is the identifier of the parameter. This must correspond to the parameter definition in the reporting tool.

Required indicates that a value for the parameter must be defined when submitting the report.

Sort Sequence must match the parameter order defined in the reporting tool's report. It is also used when displaying the list of parameters on the report submission page.

Characteristic Type indicates the characteristic type used to define this parameter.

Default Value is option and if populated is displayed to the user when the report is chosen on the report submission page.

Description is a brief description of the parameter. This description is used when displaying the parameter on the report submission page.

Long Description is a detailed description of the parameter. This description is used on the report submission page when the user requests more information for a given parameter.

Sample Reports Supplied with the Product

Depending on your specific product, there may be sample reports provided that your organization may use as they are or as a starting point for creating a new report. The following sections provide an overview of the sample reports along with instructions on how to use one of the sample reports in your implementation environment.

- How to Use a Sample Report Provided with the System

How to Use a Sample Report Provided with the System

If you would like to use any of the sample reports, you need to perform some steps to be able to execute them in an implementation environment. This section walks you through the steps needed.

- Steps Performed at Installation Time

Copyright © 2026, Oracle and/or its affiliates.

Steps Performed at Installation Time

Refer to the *Installation Guide* or *Optional Products Installation Guide* for instructions for setting up and configuring your product and reporting tool to use the sample reports provided with the system. The following steps are described there.

- Setting up the stored procedures used by the sample reports.
- Defining the company title and logo used by the sample reports. Note the company name used as a title in the sample reports is defined as a message on the installation options.
- Defining a user for integration with your product.
- Publishing the sample reports in BI Publisher.

Contact your system administrator to verify that the above steps have occurred.

How To Define A New Report

- Use a Sample Report as a Starting Point
- Publishing Reports in BI Publisher
- Designing Your Report Definition

Use a Sample Report as a Starting Point

- Make a copy of the report and save it in an appropriate directory. Prefix the new report name with **CM**.
- Review the stored procedure(s) used for this report. Refer to the installation guide for information about where the stored procedures should be defined. If you want to change the data that is being accessed, copy the stored procedure, prefixing the new stored procedure with **CM**. Make the appropriate changes in the new version of the stored procedure. Contact your database administrator to find out the procedure for creating a new stored procedure.

> **Note**
>
> **Performance considerations**. When designing a stored procedure, you must consider the performance of the report when executed. Consult your database administrator when designing your database access to ensure that all issues are considered.

> **Note**
>
> **Defining Messages**. The stored procedures provided with the system use messages defined in message category 30. If your new stored procedures require new messages, use message category 90000 or greater, which are reserved for implementations.

- Review the parameters used by the report. Make appropriate changes to the parameters required by the report. This affects how you define your report. Refer to Designing Parameters for more information.

Copyright © 2026, Oracle and/or its affiliates.

- Determine whether or not you require cross validation for your report parameters. If any cross validation is necessary, you should design an appropriate validation algorithm to be executed when requesting a report in your product. Refer to Designing Validation Algorithms for more information.

> **Note**
>
> **Cross Validation for On-line Submission Only**. The cross validation algorithm is only executed for ad-hoc report submissions via your product. If you submit this report through your reporting tool, this algorithm is not executed.

- Review the labels used by the report. Labels and other verbiage are implemented in the sample reports using a reference to the field table in the system. This enables the report to be rendered in the appropriate language for the user. For any new report label you require, you must define a new field entry. Refer to Designing Labels for more information.
- Review the layout of the report and make any desired changes based on your business needs.

When you have finished designing and coding your new report in your reporting tool, you must do the following in order for it to be usable:

- Publish the report in BI Publisher. Refer to the documentation for this products for details about publishing a report. Refer to Publishing Reports in BI Publisher for configuration information specific to publishing a report for integration with your product.
- Define the report. Refer to Designing Your Report Definition for more information.

Publishing Reports in BI Publisher

Please refer to the documentation for BI Publisher for more information about publishing a report in this system. The remaining topics in this section provide information about settings needed to ensure that the report is accessible using BI Publisher.

- BI Publisher Database Access
- Verify BI Publisher User Access Rights

BI Publisher Database Access

When publishing a report in BI Publisher, you are asked for database logon information. The logon user name and password must be the user name and password that has access to the database functions related to this report in your database.

Verify BI Publisher User Access Rights

To verify the user's access rights to folders in BI Publisher:

- Open the BI Publisher Enterprise Security Center.
- Check that the role for the user has access to the appropriate report folders.

For more information, refer to the "Understanding Users and Roles" section in the Oracle Business Intelligence Publisher User's Guide.

Copyright © 2026, Oracle and/or its affiliates.

Designing Your Report Definition

When adding a new report, you must define it in the system to allow users to request ad-hoc reports from on-line and to take advantage of the multi-language provisions in the system. The following topics illustrate the steps to take to correctly configure your report definition.

- Designing Main Report Definition Values
- Designing Characteristic Types
- Designing Parameters
- Designing Validation Algorithms
- Designing Application Services
- Designing Labels

Designing Main Report Definition Values

Refer to field description section of the report definition main page for information about defining general information about the report.

For the validation algorithm, preliminary steps are required. Refer to Designing Validation Algorithms for more information.

For the application service, preliminary steps are required. Refer to Designing Application Services for more information.

Designing Characteristic Types

The parameter tab on the report definition page uses Characteristic Type to define the report parameters. For each report parameter that you plan to use, you must define a characteristic type.

You do not need a unique characteristic type for each report parameter. For example, if Start Date and End Date are parameters your report, only one **Report Date** characteristic type needs to be defined. This characteristic type would be used on both date parameters.

Each characteristic type to be used as a report parameter must indicate a characteristic entity of **Report**.

To illustrate the characteristic type definitions, let's look at the sample report Tax Payables Analysis. It needs the following parameters: From Date, To Date, GL Account Type Characteristic Type and Account Type value.

> **Note**
>
> **Account Type Parameters**. The tax payables report must find general ledger entries that have posted to a certain distribution code. In order to find the appropriate distribution code, the report expects each distribution code to be defined with a characteristic indicating its GL account type (for example, **Revenue**, **Asset**, etc.) The report needs to know the characteristic type used to define this entry.

To support the required parameters, the following characteristic types are needed.

Char Type	Description	Type	Valid Values	Char Entities

Copyright © 2026, Oracle and/or its affiliates.

CI_DATE	Date Parameter	Ad-hoc	(Uses validation algorithm to validate proper date entry)	Report
CI_CHTYP	Characteristic Type	FK Reference	CHAR_TYP	Report
CI_GLTY	GL Account Type	Pre-defined	A- Asset, E-Expense, LM-Liability/ miscellaneous, LT-Liability/taxes, R-Revenue	Distribution Code, Report

Highlights for some of the above settings:

- We have defined a characteristic type for defining a characteristic type. This is to allow the user to indicate which Char Type on the Distribution Code is used for the GL account type. This is an FK reference type of characteristic.
- The GL account type characteristic type is referenced on both the Distribution Code entity and the report entity.

Designing Parameters

Your report definition parameters collection must define a unique parameter entry for each parameter sent to the reporting tool. The sequence of your parameters must match the sequence defined in your reporting tool.

Continuing with the Tax Payables Analysis report as an example, let's look at the parameter definitions.

Parameter Code	Description	Char Type	Default Value
P_FROM_DT	From Date	CI_DATE	N/A
P_TO_DT	To Date	CI_DATE	N/A
P_CHAR_TYPE	Account Type Characteristic	CI_CHTYP	CI_GLTY
P_TAX_ACCTY_CHAR	Account Type Char Value for Tax Related GL Account	CI_GLTY	LT-Liability/taxes

Highlights for some of the above settings:

- The from date and to date parameters use the same characteristic type.
- The characteristic type parameter is defined with a default value pointing to the GL account type characteristic type.
- The GL account type parameter defines the liability/taxes account type as its default value.

> **Note**
>
> **User Id**. The sample reports provided by the system pass the user id as the first parameter passed to the reporting tool. It does not need to be defined in the parameter collection for the report.

Copyright © 2026, Oracle and/or its affiliates.

Designing Validation Algorithms

When designing your report definition, determine if cross validation should occur for your collection of parameters. In the Tax Payables Analysis report, there are two date parameters. Each date parameter uses the characteristic type validation algorithm to ensure that a valid date is entered. However, perhaps additional validation is needed to ensure that the start date is prior to the end date. To do this, a validation algorithm must be designed and defined on the report definition.

The system provides a sample algorithm `RPTV-DT` that validates that two separate date parameters do not overlap. This algorithm should be used by the Tax Payables Analysis report.

If you identify additional validation algorithm, create a new algorithm type. Create an algorithm for that algorithm type with the appropriate parameter values. Plug in the new validation algorithm to the appropriate report definition.

Designing Application Services

Application services are required in order to allow a user to submit a report on-line or to view history for a report. Define an application service for each report and define the user groups that should have submit/view access to this report.

Update report definition to reference this application service.

Designing Labels

The system supports the rendering of a report in the language of the user. In order to support a report in multiple languages, the verbiage used in the report must be defined in a table that supports multiple languages. Some examples of verbiage in a report include the title, the labels and column headings and text such as "End of Report".

The system uses the field table to define its labels.

> **Note**
>
> **Report Definition.** This section assumes that your new report in the reporting tool has followed the standard followed in the sample reports and uses references to field names for all verbiage rather than hard-coding text in a single language.

For each label or other type of verbiage used by your report, define a field to store the text used for the verbiage.

- Navigate to the field page using **Admin**, and then **System**, and then **Field**.
- Enter a unique **Field Name**. This must correspond to the field name used in your report definition in the reporting tool and it must be prefixed with **CM**.
- Define the **Owner** as **Customer Modification**.
- Define the **Data Type** as **Character**.
- **Precision** is a required field, but is not applicable for your report fields. Enter any value here.
- Use the **Description** to define the text that should appear on the report.
- Check the **Work Field** switch. This indicates to the system that the field does not represent a field in the database.

Copyright © 2026, Oracle and/or its affiliates.

Update the report definition to define the fields applicable for this report in the **Labels** tab.

If your installation supports multiple languages, you must define the description applicable for each supported language.

Health Assessment Options

The product provides functionality to periodically collect health assessment scores of various aspects of the application. Such mechanism has the following benefits:â€‹

- Minimize risks and costs by reporting and addressing issues promptly.
- Proactively identify common implementation issues in customer environments to address potential problems early.â€‹
- Leverage best practices that are based on proven strategies from other Oracle products to ensure consistency and efficiency.â€‹

Following points highlight some of the features of this functionality:

- The health assessment type object defines configuration related to a given assessment rule, including various criteria that determine whether it is applicable to the current environment or not.
- The health assessment object is used to capture a calculated assessment value at a given point in time along with a corresponding pass, fail or warning score.
- The Health Assessment Dashboard portal provides a high level compliance summary.
- The Health Assessment portal is provided to display information about a calculated assessment.
- A monitor batch control is used to create the health assessment records. Every time it's run, it calculates and captures another measurement for each health assessment type that is eligible for the current environment.
- In a cloud environment, the monitor batch control used to create the health assessment records is considered an internal batch job and will be run by the Oracle Utilities Cloud Service.

> **Note**
>
> Although the health assessment functionality is required for cloud implementations, implementations that are on premises may also choose to run the health assessment type's monitor batch control to capture information for viewing in the Health Assessment Dashboard portal if desired.

- Understanding Health Assessments
- Defining Health Assessment Types
- Health Assessment
- Health Assessment Dashboard

Understanding Health Assessments

A health assessment record captures the environment's score for a given condition at a given point in time. Health assessment records are calculated and captured periodically to assess an environment health over time.

Copyright © 2026, Oracle and/or its affiliates.

Health assessment type describes a health condition that should be measured and evaluated over time along with remediation instructions for completeness. The condition may be classified as being critical or not and it is assigned a category for reporting purposes.

The definition specifies the calculation rule for measuring the condition along with settings that determine whether it is eligible for the current environment or not.

> **Note**
>
> Base delivered assessment types cannot be inactivated. They represent conditions that should be checked for product health. Your organization can mark an assessment type as hidden from report for it not to appear on the Health Assessment Dashboard portal by default. The portal allows you to toggle your view to include or exclude those as needed.

This section describes topics related to designing and reviewing health assessments.

Eligibility Rules

A health assessment condition may only be applicable in certain environments and not others. For example, a condition that should only be checked in a production environment or is only applicable in a cloud installation, etc. By default, a health assessment type is considered eligible in any environment unless explicitly defined otherwise.

For a cloud implementation, the system looks at the applicable cloud offerings setting on the health assessment type. If that is populated with specific cloud offerings, then the assessment type is only applicable if your product is in the list. (If no applicable cloud offerings are populated, then it's considered applicable for all cloud offerings.)

The health assessment calculation algorithm, referenced on the health assessment type definition, may further determine that the assessment type is not eligible for the current environment.

Capturing Health Assessments

The health assessment type monitor batch process, which is used to calculate and store assessments, is scheduled as an internal batch process in the cloud.

Implementations that are on premises may also choose to run the health assessment type's monitor batch control to capture information for viewing in the Health Assessment Dashboard portal if desired.

The process of calculating a health assessment of a given type is as follows:

- Basic eligibility rules are evaluated, such as whether this is a cloud or production environment, etc. If the assessment type is not eligible it is skipped.
- The calculation algorithm is called. If the algorithm indicates the assessment type is not eligible it is skipped.
- If an exception occurred while calling the algorithm, i.e. a measured value could not be calculated, the assessment score is set to **Exception** and the assessment record records the exception details.
- If the algorithm did not score the measured value, and a scoring range is defined on the assessment type, the value is scored accordingly. Scoring the measured value is required and if it cannot be scored an exception is issued.

Copyright © 2026, Oracle and/or its affiliates.

- If the algorithm did not provide the percentage to failure value, it is calculated based on the assessment score and the scoring range defined on the assessment type.
- The assessment record is stored along with any optional characteristics provided by the algorithm.

Health Assessment Score

The measured assessment value is scored as either **Pass**, **Warning** or **Fail** to describe the severity of the condition. In addition, a **Percentage to Failure** value is also provided to indicate how close the measured value is from a failure score as a proactive alert mechanism. A pass score is set to 0% and a failed score to 100%.

The algorithm associated with the assessment type is responsible for computing a measured value. It may also provide a score for the measurement and calculate its percentage to failure value.

If the algorithm did not score the value, the system uses the scoring range defined on the assessment type to score the calculated value. A value less than or equal to the pass limit is set with a 'pass' value and if greater or equal to the 'fail' limit then is set to 'fail'. Any value is between is scored with 'warning' value. If a scoring range is not defined the assessment is scored with a 'warning' value.

For example, if an assessment type is defined with a maximum pass value of 7 and a minimum failure value of 11 then a measured value of 5 would be scored as 'pass', a value of 9 as 'waring' and a value of 11 as 'fail'. In the same way, if a percentage to failure is not set by the algorithm, the system calculates it based on the same range. A value scored as warning is set with a higher percentage to failure the close it is to the fail limit. If a scoring range is not defined the default is 0 as percentage to fail.

Assessment Calculation Algorithm

The health assessment type definition references a processing algorithm for calculating the condition's measurement value and scoring it. If the algorithm does not score the value, settings on the assessment type may be used to do so.

Basic eligibility rules are defined on the assessment type definition. For more specific rules, the algorithm may also determine that the assessment is not eligible for the current environment and therefore should be skipped.

The algorithm may provide basic supporting details as characteristics to display along with the assessment.

The product provides generic assessment algorithms that can be used to simplify the configuration of basic assessment types. For example, an algorithm that gets a measurement value labeled as 'count' from an SQL provided as an parameter. The measured value is scored by comparing it against the scoring range defined on the assessment type. Refer to product health assessment types for examples.

Launching A More Information Portal

An assessment type may be associated with an optional portal for reviewing specific details about an assessment. When specified, the health assessment portal provides navigation to the specific portal along with the health assessment identifier as context. This is an optional configuration and most likely needed to clarify complex assessment conditions.

Copyright © 2026, Oracle and/or its affiliates.

Review Assessments

The Health Assessment Dashboard portal provides a high level summary of health assessment conditions. It is recommended to review this portal periodically and address highlighted issues as needed.

Health assessment types provided by the base product cannot be disabled. Customers can mark certain product assessment types to be hidden from this dashboard display. These are still assessed and collected but a user may choose to view them or not on this dashboard. The user may drill into a Health Assessment portal for more information about a given rule over time. â€‹When applicable, the user may also further navigate to the specific portal associated with the rule.

No Custom Business Objects

Health assessment functionality relies on base product business objects for core functionality. There is no support for custom business objects nor customization of the base product's business objects. Customers can add assessment types with custom calculation rules but the assessment type and assessment entities must be based on product business objects.

Defining Health Assessment Types

Health assessment types are mainly defined by the product but allow additional ones to be added by your organization. To view the health assessment types, open **Admin**, and then **System**, and then **Health Assessment Type.**

Use the **Health Assessment Type Query** portal to search for a health assessment type. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Health Assessments for more information.

For information about the data defined on the product metric type, refer to embedded help.

Health Assessment

The Health Assessment portal displays the measurement score and additional information related to a specific health assessment rule.

To view this portal, open **Main**, and then **System**, and then **Health Assessment.**

Refer to Understanding Health Assessments for more information.

The portal includes the following zones:

- **Health Assessment.** This zone displays details about a health assessment record along with guidance on how to address such issue.
- **Assessment History.** This zone lists a history of assessment values prior to the current assessment.

For more information about what each zone on the portal is reporting, refer to the embedded help.

Copyright © 2026, Oracle and/or its affiliates.

Health Assessment Dashboard

The Health Assessment Dashboard portal provides a high level summary of health assessment scores for key aspects of your application. The information visible on this portal will depend on the product or products installed by your organization.

To view this portal, open **Main**, and then **Dashboards**, and then **Health Assessment Dashboard.**

Refer to Understanding Health Assessments for more information.

The portal includes the following zones:

- **Critical Assessments.** This zone summarizes the latest assessment for all critical assessment types by assessment score.
- **Overall Assessments.** This zone summarizes the latest assessments of all types by assessment score.
- **Assessments.** This zone lists latest assessments by assessment score.

For more information about what each zone on the portal is reporting, refer to the embedded help.

Service Health Check

The system provides a plug-in spot on Installation, called Health Check. The algorithms plugged in here are used to check various system conditions and return details to help assess the health of the system.

What Information is Checked?

Each algorithm is responsible for checking one type of health check condition. The condition should be defined as a valid health component type (as defined in the `HEALTH_COMP_FLG` lookup). The algorithm may check conditions across many records (referred to as Health Components) and return a status and details for each health component. For example, if the algorithm is checking the level of service for batch programs in the system, the batch control is the health component in this case. The algorithm should gather the level of service results for each batch program and return the collection of information to the calling program.

The algorithms for the plug-in spot should return the following information for each health component.

- The **Health Component Detail** should be populated with information specific to the object that was checked. For example, if the algorithm is checking batch control level of service, this will have the Batch Control code. In addition, for enabling display for details of each health component type, the algorithm should populate the maintenance object code and primary key field and value details along with the navigation option.
- The **Health Component Status Flag** and **Health Component Status Description** should be populated with a valid status code and description appropriate for the condition being checked. This information could differ based on the health component type as each health component type may have its own unique status values.
- The **Health Component Status Reason** is meant to provide supporting information about the health component's status value. For example, if the component provides an error type of status, this could have the status reason.

Copyright © 2026, Oracle and/or its affiliates.

- The **Health Component Response** is used to map the status value of the health component to a standard value defined in the lookup `HEALTH_RESPONSE_FLG`. The values are a subset of HTTP response codes. The supported values are **All Checks Successful** (200); **Non-Critical Function Degraded** (203), **No Content** (204) and **One or More Critical Functions Degraded** (500).
- The **Response Details** list is provided for health components that check several conditions as part of the health check. Each condition could return its own status. In this situation, the algorithm should use the response details list to record the individual responses and populate the overall Health Component Status Flag, Description and Reason with summary information. The suggestion is to do the following:
  - Set the status flag and response based on the details of the individual responses. Refer to the batch level of service information below for an example of this logic.
  - When the response list is populated, the health check user interface will show the overall Status Reason text along with an icon to expand the details. The suggestion is to populate this field with text like "See Results for Details".

The system provides an algorithm that checks the **Batch Level of Service** health component type. This health component type finds all the batch controls that are configured with at least one level of service algorithm and invokes the algorithms for each batch control. The business service populates the output for this health service for each batch control as follows:

- The **Health Component Detail** is populated with the Batch Control code and description. In addition, the navigation information for being able to drill into the batch control are provided and used to build the column as hypertext.
- The **Status** is populated based on whether the batch control has one algorithm or multiple. If there is one algorithm, the Level of Service lookup value returned by the algorithm is returned. If there are multiple, the system determines an overall status based on the detailed status values from each algorithm. If any algorithm returns Error, that value is returned. Otherwise, if any return a Warning, that value is returned. Otherwise Normal is returned.
- The **Status Reason** is populated based on whether the batch control has one algorithm or multiple. If there is one algorithm, the expanded text of the status reason returned by the algorithm is returned. If there are multiple, the text from message category / message number 11002 / 22001 is returned (**See Results for Details**)
- The **Response** is populated based on the value of the overall Level of Service status. It is set to **All Checks Successful** (200) when the Level of Service is **Normal** or **Disabled**; **Non-Critical Function Degraded** (203) when the Level of Service is **Warning** and **One or More Critical Functions Degraded** (500) when the Level of Service is **Error**.

How are the Algorithms Called?

These algorithms are called by a business service provided by the product `F1-HealthCheck`.

That service calculates an overall Health Response value based on all the details returned by all the algorithms. The values described above for the `HEALTH_RESPONSE_FLG` are the ones used.

Also note that the system provides Inbound Web Services for this business service for both SOAP and REST service calls allowing external systems to use a web service to retrieve this information.

- Health Check Configuration
- Viewing Health Check Results
- Introducing Health Check Conditions

Copyright © 2026, Oracle and/or its affiliates.

Health Check Configuration

To enable your system to display health check information, navigate to the Installation Options - Algorithms collection and add an entry for the **Health Check** system event for each health check algorithm provided by the product that your product would like to include in your system health check assessment.

Viewing Health Check Results

The product provides a portal that allows a user to view the detailed results of the health check business service.

Navigate using **Admin**, and then **System**, and then **Health Check** to view this portal.

The zone on the portal displays the following:

Overall Response is the HTTP response code returned by the business service.

The grid displays the detail of each individual component checked as part of the system health check. Refer to Service Health Check for more information about the details returned.

Introducing Health Check Conditions

If your implementation would like to check a condition for system health that is not already provided by your product, here are the steps required:

- Define a new value in the Health Component Type lookup (`HEALTH_COMP_FLG`).
- Provide a new algorithm type, plug-in script and algorithm for the Health Check system event that checks the condition that you wish to check. Refer to the Service Health Check topic for the information that each algorithm should be supplying for the condition. Note that if you follow the pattern of the system delivered Batch Control - Level of Service algorithm, your algorithm may be designed to check conditions across many records of the same type and gather all the results.
- Plug this new algorithm into the Installation Options - Algorithms collection using the **Health Check** system event.

Measuring Performance

Many implementations need the ability to track and view the performance of key system processes against a defined target level. The framework provides objects to allow an edge product or an implementation to calculate and display performance measures against a desired target for one or more use cases. The topics in this section provide information about what is supplied in the framework and guidelines for implementing a specific use case for batch processes. Your product may supply out of the box support for additional use cases. Refer to your product documentation for more information.

- Understanding Performance Targets
- Setting Up Performance Target Configuration
- Maintaining Performance Targets

Copyright © 2026, Oracle and/or its affiliates.

Understanding Performance Targets

The following are examples of use cases that would be well suited for tracking as performance targets:

- Track and view the duration of key batch processes, either individually or as a group, and how they relate to a defined target.
- Monitor used and free space on a database against critical levels.
- Check the performance of individual user interface zones against a defined performance expectation
- Compare the number of web service requests made to an application against a threshold where performance may be of concern.

Framework provides functionality to define and categorize performance targets and link them to objects such as business services, zones and portals. This supports the calculation and display of the metrics against desired results.

In addition, Framework supplies out of the box support for batch process performance targets. Individual edge applications may supply more specific functionality for other use cases, if applicable.

Ideally, users should have the ability to view these performance targets on a dashboard that groups related measures. Framework provides the necessary components to achieve this for batch process performance targets.

The following sections highlight functionality supported for performance targets in the framework. Refer to the edge application product documentation for more details of other supported use cases.

- Performance Target Objects Overview
- Calculating and Displaying Performance Targets
- Performance Target Metrics and Metric Types
- Performance Target Categories and Types
- Performance Targets Define Specific Metrics
- Objects Linked to a Performance Target
- Creating Performance Target Zones

Performance Target Objects Overview

The setup of a performance target involves a unique combination of configuration data and processing logic that calculates and displays a specific measure.

The framework performance target functionality is supported by a combination of inter-related objects, as shown below. Some of these objects will be generic for use in all performance targets while some are specific to a functional area such as batch processes.

- **Maintenance Objects** for capturing performance target types and performance target instances.
- **Extendable Lookups** to define performance target categories and performance target metrics.
- **Business Services** to calculate known metrics for a group of performance targets, such as for batch processes.

Copyright © 2026, Oracle and/or its affiliates.

- **UI Maps** to interpret the performance calculation results and display them in charts that show the comparison to the target.
- A **Zone** that serves as a template for system duplication to create specific zones related to each performance target. The zone invokes a business service to perform calculations and display performance measures in the related UI map.
- **Business Objects** to capture configuration data for a specific performance target instance and its related objects.
- Functionality to create **Zone instances** for specific performance targets based on the associated template zone.

Each edge application will deliver the following to compliment the objects delivered by the framework:

- **Portals** to group related performance target zones.
- Specific entries for the performance target category **Extendable Lookup** .

The following sections describe the combined use of these objects for performance targets in more detail.

Calculating and Displaying Performance Targets

Performance targets are intended to be displayed in a portal using an explicit object zone. The zone parameters define both the business service used to calculate the performance metrics and the UI map that displays the results. While an individual performance target needs to reference a zone with a unique configuration that calculates a particular metric, those zones will be based on a template zone which defines the core parameters.

Framework provides a base Batch Performance Target Metric Template zone (`F1-PERFBA`) for batch process performance targets. Refer to this zone and its parameters for more information about the batch performance zone configuration and the related business service and UI map.

Performance Target Metrics and Metric Types

The framework supports two types of metrics for performance monitoring:

- **Value based metrics** are used to record results against a specific numeric target.
- **Time based metrics** are used to track the results against a specific date and time target.

The list of valid metrics for a given performance target category and its associated performance target types is maintained using an extendable lookup. Framework uses the base business object Batch Performance Target Metric (`F1-BatchPerfTargetMetric`) to define batch process metric values. Refer to this lookup for the supported batch process metrics.

> **Note**
>
> While users are not prevented from adding new values to the lookup, the list is not intended to be extendable as new values will not be recognized by the business service that performs the base batch performance calculation logic.

Your edge product or implementation may supply other extendable lookup business objects for additional performance measurement use cases, if applicable.

Copyright © 2026, Oracle and/or its affiliates.

Performance Target Categories and Types

There are key configuration details required by all performance targets. These are defined on two related objects.

Performance Target Categories

Target categories define the template zone and security setting for a group of performance measures. The list of valid categories is maintained using an extendable lookup.

The framework product supplies the business object Performance Target Category (`F1-PerformanceTargetCategory`) for this functionality. Refer to the business object description and configuration for more information.

Performance Target Types

Target types define the related performance target business object and the display portal for a group of performance measures. In addition, a target type references a target category which defines the associated zone details.

The framework product supplies the business object Performance Target Type (`F1-PerformanceTargetType`) for this functionality. Refer to the business object description and configuration for more information.

The framework does not deliver any standard type or category values for batch processing performance targets. Refer to your specific edge application products to verify if any standard values are delivered for the batch processes within your applications. Edge applications may also supply standard categories and types for additional performance target use cases.

Performance Targets Define Specific Metrics

Although the types of measures and the business services and UI maps that govern how they are calculated and displayed are defined using separate objects, a **performance target** record defines the additional configuration needed to measure a specific metric and compare the result to a desired value.

There are key configuration details required by all performance target instances. These include a reference to the metric being measured, the desired target value or time, the desired result for the target and the unique zone by which this performance target will be monitored.

> **Note**
>
> The performance target maintenance object has a direct foreign key link to the extendable lookup business object and value that define the performance metric. This is an unusual pattern as extendable lookup values are normally recorded only in an object's schema. The pattern has been adopted to allow the description of the metric to be displayed in the performance target maintenance portal.

A given **performance target type** may require additional details for its calculations. For example, the framework batch process performance target defines additional details to restrict the measurement to specific batch processes that have executed within a given time frame. These details are configured on the performance target business object. Refer to the embedded help text on the business objects supplied by the framework and your edge applications for more information.

Copyright © 2026, Oracle and/or its affiliates.

The **performance target type** also defines the business object to use when creating the resulting performance target record.

Objects Linked to a Performance Target

Some performance measures such as batch process metrics, derive the data for the calculation from objects within the system. The performance target can be linked to one or more related objects to define the specific data sources included in that target metric. For example, when creating a batch process performance target, the framework supports linking specific batch codes to the performance target to indicate the group of batch processes that are included in the measure.

The performance target business object may be configured to allow only relevant objects types to be linked to a performance target record. Refer to the base Batch Performance Target business object (`F1-BatchPerformanceTarget`) for an example.

Creating Performance Target Zones

Once a performance target and its objects have been defined, a unique zone needs to be created to display and monitor the specific target.

The framework functionality for batch performance targets implements the zone creation via the related business object lifecycle. When a performance target is added, status enter plug-ins are responsible for generating the new zone using the template zone and prefix configured on the target category (or the override values) and adding the zone to the portal configured on the target type. When a performance target is inactivated, an enter plug in is responsible for removing the zone from the portal.

> **Note**
>
> While the template zone associated with the performance target category may be overridden, the zone generation algorithm makes certain assumptions about the zone type and parameters. In particular, the logic expects to configure a zone parameter that references the performance target code as input to the business service responsible for calculating the metrics.

Refer to the base Batch Performance Target business object (`F1-BatchPerformanceTarget`) for details of the lifecycle and the enter plug-ins responsible for performance target zone creation.

Setting Up Performance Target Configuration

The following topics highlight the general configuration steps required to use performance target functionality. Your particular product or implementation may supply additional functionality to support specific use cases for performance targets. Refer to your product documentation and the library of business objects supplied for performance target in your system for more information.

- Performance Target Category Lookup
- Defining Performance Target Types

Performance Target Category Lookup

Refer to About Performance Targets for an overview of performance target functionality.

Copyright © 2026, Oracle and/or its affiliates.

Each Performance Target Type is associated with a unique Performance Target Category. The categories are defined as an extendable lookup.

Navigate to the Extendable Lookup portal. Search for and select the **Performance Target Category** business object. Define values that are appropriate to the categories your implementation is assigning for the performance target types in use. In some cases, an edge application may already have delivered the appropriate performance target category for your use

Refer to the embedded help for more information about configuring this object.

Defining Performance Target Types

Refer to About Performance Targets for an overview of performance target functionality.

To maintain the performance target types applicable to your product or implementation, open **Admin**, and then **Reports**, and then **Performance Target Type.**

This is a standard All-in-One portal.

The information captured on the performance target type depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

Maintaining Performance Targets

This section describes the functionality supported for viewing and maintaining performance targets.

Refer to About Performance Targets for an overview of performance target functionality.

Navigate using **Main**, and then **Reporting**, and then **Performance Target**. This is an All-in-One portal and includes the standard List and display zones for a performance target.

For information about the data defined on the performance target, refer to embedded help.

Capturing Statistics

The product provides objects to allow for an edge product or an implementation to periodically calculate and capture statistics for one or more use cases. The topics in this section provide information about what is supplied in the framework and guidelines for implementing specific use cases.

- Understanding Statistics
- Configuring Your System for Statistics

Understanding Statistics

There are two objects that work together to capture the statistics related to a given business use case:

- The statistics control record which defines configuration related to capturing statistics. It also defines a category that is used to group similar types of statistics together.
- The statistics snapshot is the record where the actual calculated statistics are captured.

Copyright © 2026, Oracle and/or its affiliates.

Statistics Control

The statistics control record defines whether or not statistics are automatically calculated and how frequently. In addition, you may control the retention policy of snapshot records. You may indicate that only the most recent snapshot is kept or all snapshot records are kept or that they are retained for a defined number of days.

The system provides a business object for statistics control (`F1-Statistics`) that is expected to be used for most use cases. Its lifecycle includes the support for periodic capturing of statistics snapshot records as well as on demand capture.

Statistics Snapshot

The statistics snapshot object is the object that is responsible for calculating and capturing the statistics details. A separate business object must be defined for each use case. The schema of the business object defines the details that are captured. In addition, it must include an algorithm as an enter plug-in on the **Complete** state that is responsible for capturing statistics.

The system provides a 'root' business object for statistics snapshot (`F1-SnapshotRoot`) which includes the lifecycle that statistics snapshot business objects should follow. No explicit statistics snapshot use case is provided by the Framework. Your specific product may supply some out of the box support for certain use cases, in which case specific statistics snapshot business objects are provided. Refer to your product specific documentation for more information.

Viewing / Reporting Statistics

The product provides the mechanism for defining statistics control and statistics snapshot records and viewing the data in basic format. If your product has provided support for specific use cases, there may also be additional portals used to display the statistics in a meaningful format. For example, the statistics snapshot may be capturing data that can be presented in graphical format. Additionally, if multiple historical statistics snapshot records are retained, a zone may be defined to graph changes over time.

Configuring Your System for Statistics

If your product provides statistics use cases that you are planning to implement, all that you need to do is configure the appropriate statistics control record and define the appropriate configuration for your business requirements. Refer to Defining and Monitoring Statistics for more information.

If your implementation has identified an additional use case where you would like to capture statistics, the following points highlight the steps needed to configure the system to support the use case.

- Determine whether an additional Statistics Category value is needed. This is captured on the statistics category record. Navigate to the Lookup page. Search for and select the `STAT_CATEGORY_FLG` field. Review the values and determine if additional values are needed.
- Define a new statistics snapshot BO. This should be a child BO of the base delivered BO (`F1-SnapshotRoot`). Its schema should define elements for the specific information that is captured by the statistics calculations. The schema should be designed in conjunction with an appropriate Enter Status algorithm for the **Complete** that calculates the statistics as appropriate for the business requirement.
- Once the snapshot business object is designed and implemented, configure the appropriate statistics control record.

Copyright © 2026, Oracle and/or its affiliates.

- Defining and Monitoring Statistics

Defining and Monitoring Statistics

Refer to Understanding Statistics for an overview of the statistics functionality.

To define and maintain statistics control records and view a list of the current snapshot records, open **Admin**, and then **Reporting**, and then **Statistics Control.**

This is an All-in-One portal and includes the standard List and display zones for a statistics control.

For information about the data defined on the statistics control, refer to embedded help.

Statistics Snapshot

If there are any Statistics Snapshot records for the statistics control, the **Statistics Snapshot List** zone displays a list of the most recent records. A user may drill into any of the records to view more detail. You are brought to a maintenance portal where you may view details about the calculated statistics. The information captured will be unique to the particular use case. Refer to the embedded help for more information.

Capturing Product Use Metrics

The product provides functionality to periodically calculate and capture information used to review and confirm licensing / subscription compliance for cloud implementations.

Following points highlight some of the features of this functionality:

- The product metric type object defines configuration related to a given metric, including which cloud offerings it applies to and the product metric snapshot business object to use when capturing the metric.
- The product metric snapshot object is used to capture the calculated metric.
- The Product Use Metric portal is provided to display information about the calculated metrics.
- A monitor batch control is used to create the product metric snapshots. Every time it's run, it creates a snapshot for each applicable product metric type. For on premises, all product metric types are considered applicable. For a cloud implementation, the system looks at the applicable cloud offerings setting on the product metric type. If that is populated with specific cloud offerings, then the product metric type is only applicable if your product is in the list. (If no applicable cloud offerings are populated, then it's considered applicable for all cloud offerings.)
- In a cloud environment, the monitor batch control used to create the product metric snapshot is considered an internal batch job and will be run by the Oracle Utilities Cloud Service.

> **Note**
>
> Although the product use metrics functionality is required for cloud implementations, implementations that are on premises may also choose to run the product metric type's monitor batch control to capture information for viewing in the Product Use Metric portal if desired.

- Defining Product Metric Type

Copyright © 2026, Oracle and/or its affiliates.

- Product Metric Snapshot
- Product Use Metrics

Defining Product Metric Type

Product metric types are defined by the product. To view the product metric types, open **Admin**, and then **System**, and then **Product Metric Type.**

Use the **Product Metric Type Query** portal to search for a product metric type. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

For information about the data defined on the product metric type, refer to embedded help.

Product Metric Snapshot

Product metric snapshots are created whenever the product metric type's monitor batch is run. A snapshot is created for every applicable product metric type. Refer to Capturing Product Use Metrics for more information

To view the product metric snapshots, open **Admin**, and then **System**, and then **Product Metric Snapshot.**

Use the **Product Metric Snapshot Query** portal to search for a product metric snapshot. Once a record is selected, you are brought to the maintenance portal to view the selected record.

For information about the data defined on the product metric snapshot, refer to embedded help.

Product Use Metrics

The Product Use Metrics portal includes a zone (or zones) that are used to perform a business review of your organization's implementation of the product to confirm licensing / subscription compliance. The information visible on this portal will depend on the product or products installed by your organization.

To view this portal, open **Main**, and then **Dashboards**, and then **Product Use Metrics.**

For more information about what each zone on the portal is reporting, refer to the embedded help.

External Messages

This section describes mechanisms provided in the product that enable an implementation to configure the system to communicate with an external application.

- Incoming Messages
- Outgoing Messages
- Web Service Category
- JMS Message Browser
- Oracle Integration Cloud Catalog

Copyright © 2026, Oracle and/or its affiliates.

- XAI Documentation Note

Incoming Messages

This section provides information about support for incoming messages.

- Inbound Web Services
- Guaranteed Delivery

Inbound Web Services

Inbound web service functionality is provided to support receiving web service requests from an external system.

- Understanding Inbound Web Services
- Configuring Inbound Web Service Options
- Deploying Inbound SOAP Web Services

Understanding Inbound Web Services

The system supports communicating with the system via RESTful services or via SOAP services. In both cases, the system uses an object called inbound web service (IWS) to store the configuration. A web service class is used to distinguish whether the IWS is used for REST or SOAP. There is also a distinct business object for each web service class value because there is different configuration for each type of IWS.

For both types of inbound web service, the system supports the configuration of one or more operations per web service. Each operation defines the schema-based object to invoke to perform the desired function. An operation may refer to a Business Service, a Business Object, or a Service Script. If the IWS supports multiple operations, each operation can refer to the same or a completely different schema-based object from other operations within the IWS.

REST APIs may be secured by OAuth client credentials grant in a cloud installation. Refer to the Identity and Access Management chapter in the *Oracle Utilities Cloud Services Administration Guide* for more information.

The following topics provide more information about the different configuration of IWS based on whether it is SOAP or REST.

- Inbound REST Web Services
- Inbound SOAP Web Services

Inbound REST Web Services

This topic provides more information about the product's support of REST services using IWS.

The REST engine checks the Accept-Language header and attempts to find a supported language whose locale matches the referenced language. If found, the web service response returns any translatable data in that language. Note that if no supported language is found for the languages provided in the Accept-Language header, the system uses the language of the user that submitted the request.

IWS REST records are marked with a REST engine version of 1.0 or 2.0. Version 2.0 is the version used for new services. Version 1.0 is introduced for backward compatibility. Some of

Copyright © 2026, Oracle and/or its affiliates.

the system behavior differs based on the version of the REST service. The sections below highlight cases where this occurs.

HTTP Method and Parameters

When defining operations for inbound REST web services, the product supports the HTTP methods of Get, Patch, Post, Put, and Delete. Note that the product's support of these various HTTP methods are a means of communicating the purpose of the web service to the outside world. However, the actual behavior of the REST web service is driven by the behavior of the underlying schema based object (business object, business service or service script). For example, you may configure the HTTP Method of "Put" for an operation that references a service script that just retrieves a record. The product is not able to detect this type of discrepancy. Configuration users should carefully consider the correct method to use based on the logic of that service.

For operations that reference a business object, the transaction type must be provided. The REST syntax doesn't support defining the transaction type at runtime. There are some basic validation checks in this case related to the transaction type and the HTTP method. For example, a Get method only makes sense with the Read transaction type.

> **Note**
>
> Using the transaction type **Change** requires all values to be passed in. Using the transaction type **Update** allows the web service to pass only the primary key and the values to be updated. All other elements will retain their existing values.

You may additionally define parameters. For each parameter, you define an external reference to the element, which is how this parameter is exposed to the external callers and is defined in the API specification. Each of these parameters is mapped to the XPath of the schema element from the underlying business object, business service or service script. For each parameter, you indicate if it is a **Path** parameter or a **Query** parameter.

- Path parameters are parameters that are part of the endpoint and are required. Each path parameter must be included in the operation's URI component surrounded by curly brackets.
- Query parameters are optional. They are not part of the endpoint but rather are included in the endpoint URL after a question mark, followed by name value pairs.

Refer to the URL section below for examples of path and query parameters in the sample URLs.

URL Composition

When building the endpoint URL for a REST service, there are three main parts that make up the full URL.

- The first part is the one that is environment specific. This will differ for an on premise implementation as compared to a cloud implementation. Both will have the host and the port and then additional components that identify the environment.
- The second part is a hard coded path designated by the product, namely "/rest/apis".

  These two parts of the URL are defined in the substitution variable `F1_REST_BASE_URL`. This is defined when initializing your environment. Refer to the Server Administration Guide for more information.

Copyright © 2026, Oracle and/or its affiliates.

The remaining part of the URL is built dynamically based on configuration for each IWS and its operations. The components are "/ownerURIComponent/resourceCategoryURIComponent/iwsURIComponent/operationURIComponent"

- The owner URI component is taken from the owner flag of the inbound REST web service. A special extendable lookup (Owner Configuration for REST Services) defines this component for each owner flag.
- Each Inbound REST Web Service must reference a resource category. This category is used to associate related web services to a common category of resources. For multiple IWS records linked to the same resource category, external catalogues can use this information to group together related web services. The resource category is an extendable lookup and the URI component is an attribute of this record.
- Each REST IWS record defines a URI component that serves as an external identifier of this IWS record. The value must be unique within a given owner flag.
- Each operation must define the HTTP Method and optionally a URI component. When defining path parameters, the path parameters must be included in the URI component using curly braces.

In all cases, the URI component must begin with a slash ('/').

The following are some examples of the dynamic portion of the URL for a REST service. The last example illustrates the use of a query parameter.

IWS Name URI Component	Owner URI Component	Resource Category URI Component	Operation HTTP Method URI Component	Dynamic URL Component	Run Time Example
c1rateCalculation / rateCalculation	C1 /customer	C1-RATES /rates	Post /	../customer/rates/rateCalculation/	../customer/rates/rateCalculation/
w1WorkActivity / workActivity	W1 /asset	W1-WORK /work	Get /{activityId}	../asset/work/workActivity/{activityId}	../asset/work/workActivity/5798165498
w1WorkActivity / workActivity	W1 /asset	W1-WORK /work	Patch /scheduleWindow/{externalSystem}/{activityId}/{windowStartDateTime}	../asset/work/workActivity/scheduleWindow/{externalSystem}/{activityId}/{windowStartDateTime}	../asset/work/workActivity/scheduleWindow/MY-COMPANY/5798165498/20190101
CM-AccountActivityHistory / accountActivityHistory	CM /cm	CM-ACCT-INFO / accountInformation	Get /{accountId}	../cm/accountInformation/accountActivityHistory/{accountId}	../cm/accountInformation/accountActivityHistory/123456789?activityId=5468976

Copyright © 2026, Oracle and/or its affiliates.

Payload Format

REST services support receiving a request payload in XML or JSON format and returns the payload in either XML or JSON. The default format returned depends on the value of the REST Engine Version.

- Version 2.0 services assume JSON format as the default. The default can be overridden by providing an accept header of `application/XML`.
- Version 1.0 services assume XML format as the default. The default can be overridden by providing an accept header of `application/JSON`.

Root Node for JSON Format

Version 2.0 services with the JSON formatted payload do not accept in the request or return in the response any root node around the payload. The following is an example of the response of a REST call for a version 2.0 service:

```
{
   "batchJobId": "string",
   "requestSuccessful": "string"
}
```

Version 1.0 services with the JSON formatted payload expect a root node in the request and return one in the response. The following is an example of the response of a REST call for a version 1.0 service:

```
{
  "F1CnclBatJob": {
    "batchJobId": "string",
    "requestSuccessful": "string"
 }
}
```

The Open API Specification visible on the inbound web service maintenance page displays the expected format based on the record's REST engine version when viewing the specification.

Data Type Format in JSON

In JSON format, strings are surrounded by quotes and numbers and Boolean data do not have quotes. All services for REST engine version 2.0 follow this standard. Originally services that are REST engine version 1.0 were incorrectly treating numbers and Boolean data as strings and returning the data with quotes. This has been corrected.

To accommodate any integration that may have worked around the behavior for version 1.0 services, the system provides the ability to identify inbound web services that are exceptions to this rule. For any inbound web service identified as an exception, the system continues to treat numbers and Boolean data as strings in JSON responses. To add one or more IWS records to the exception list:

- Go to Feature Configuration.
- Look for an existing feature configuration with the feature type **External Messages**. If one does not exist, create one.
- Add an option for the option type **IWS JSON Data Type Exceptions**.

Copyright © 2026, Oracle and/or its affiliates.

- In the option value, indicate the IWS record that is an exception. Note that multiple options for the option type may be added. Additionally, the option value supports a comma delimited list.

External Facing Schema

By default the schema of the underlying service (for example, service script, business object or business service) is also the schema for the REST inbound web service operation and acts as both its Request and the Response schema. The product provides the ability to define an explicit schema for a REST inbound web service (IWS) operation, allowing a user to adjust the schema for an external facing consumer. The IWS operation schema also supports some special configuration that allows additional features to be defined only for the IWS operation schema.

The operation schema supports the following capabilities:

- Declaring whether an element belongs only to the request schema, only to the response schema, both (default) or excluded altogether. This is used to establish a clear separation between request and response schema definitions.
- Assign a different element name to an internal element. This allows the internal service element name to more closely align to internal references to the element, if needed. The external facing element name can be different allowing for a more readable schema.
- Introduce specialized elements that support links and other types of structural standards in a way that does not impact internal service processing.

Refer to Web Service Schema Nodes and Attributes for more information.

The REST servlet uses the request and response schemas to map elements to and from the internal schema.

> **Note**
>
> While the caller should only provide elements defined by the request schema, the application does not filter out extraneous elements. Assumption is that the internal service is designed to ignore such elements. However, only elements defined by the response schema are included in the response.

Dynamic Links

There are use cases where the published API includes a "_self" element that contain the endpoint URL of the GET operation related to the data returned in the response. Additionally, response payloads may include foreign keys and for those entities, the response includes a "_link" element that contains the endpoint URL of the GET operation for that entity (if it exists).

Syntax is provided in the REST inbound web service operation schema to support building the runtime endpoint URL for the _self and _link elements. Besides dynamically building the static portion of the URL based on the current environment details, it also builds the dynamic portion of the URL, substituting the URL components for the operation and substituting the path parameters. The syntax allows you to define a specific IWS Operation or allows you to reference a maintenance object and at runtime, the system determines the GET operation for the maintenance object and builds the URL accordingly.

Example of the syntax:

```
<_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>
```


Copyright © 2026, Oracle and/or its affiliates.

Example of the runtime endpoint URL:

```
_link: "http://.../common/toDos/toDoEntries/28937296450934"
```

A GET operation may be associated with a maintenance object to denote that this is the default GET operation for the entity. A **GET Operation** option at the Maintenance Object and Business Object level may be used to override this default. If the entity is not associated with a business object or the latter is not associated with such option then the operation associated with the maintenance object (if any) is used.

API Specification Documentation

The following documentation information may be provided for an Inbound Web Service:

- Short and detailed description for each operation.
- Help text may be provided for individual elements. The same text can be shared across multiple elements across all operations of the Inbound Web Service. If an element is not associated with help text then its internal field's label is used instead.
- Sample request and response documents may be provided for each operation.
- Operations may be associated with a sequence number that controls the order in which they appear in the Open API Specification.

> **Note**
>
> Detailed descriptions and element related help text are not translated since the published catalogue for base product APIs is in English language only.

Customization Options

Some base product REST web services may explicitly indicate that they allow customization. When allowed, your implementation may customize the following aspects of a service to meet integration requirements:

- Add more elements to the request and response schemas
- Implement custom logic to populate additional elements or enforce custom business rules.
- Provide help text for the additional elements and corresponding request and response examples when generating an API specification.

When an IWS allows customization, a **Custom Script** may be specified on any of its IWS operations to override the operation's underlying base service.

> **Note**
>
> The custom script must adhere to the base service schema it overrides and can only extend it. For that reason, the custom script's schema must at least include the operation's base service schema and may include additional custom elements as needed.

Copyright © 2026, Oracle and/or its affiliates.

Customizing an IWS operation does not change the URL used to access the web service. If the operation is customized then internally the custom script is called instead of the base service.

In the same way, the API specification of the operation is adjusted to use the custom script if any. If a custom web service schema is also provided then it is used instead of the base web service schema if any. If sample request and response sample documents are provided as options on the customized operation they are used instead of base samples. You may also provide custom help text for your custom elements by adding corresponding entries to the IWS help text list.

Inbound SOAP Web Services

For inbound SOAP web services, by default the system uses the Schema Name to dictate the Request and Response for the service. The API can be overridden with custom formats by specifying Request and Response XSLs to transform into the relevant schema formats. In addition, if desired, the Request and Response Schemas that document the expected message may be provided.

> **Note**
>
> The Request and Response Schema fields are not supported in a cloud implementation.

> **Note**
>
> The system supports defining the XSLs as a managed content record. For backward compatibility, the system supports defining the XSL as a file in the file system. This is a system wide setting, defined using a feature configuration option. The feature type is **External Messages** and the option type is **XSL Location**. Set the value to `F1FL` to support XSL in the file system. The file location is configured using a system property. To support the XSL in managed content, no option is needed as this is the default. You may explicitly define the value of `F1MC` if desired.

In addition, note that for business object based operations, when invoking the web service an action is required. This may be passed into the web service as part of the invocation or alternatively, the action may be defined when configuring the operation using the transaction type.

> **Note**
>
> Using the transaction type **Change** requires all values to be passed in. Using the transaction type **Update** allows the web service to pass only the primary key and the values to be updated. All other elements will retain their existing values.

Annotations Used for Security

When preparing to deploy inbound SOAP web services, the security aspects of the service must be decided. The product provides a default security policy that is applied when no other policy is defined: **@Policy(uri="policy:Wssp1.2-2007-Https-BasicAuth.xml", attachToWsdl=true)** which requires HTTP Basic over SSL and a WS-Security Timestamp.

Copyright © 2026, Oracle and/or its affiliates.

If a different security policy is desired, the following options are available:

- Security policies may be attached to the Inbound Web Service via the Java Enterprise Edition (Java EE) Web Application Server. This allows for multiple policies to be attached as supported by the Java EE Web Application Server. In order to enable this capability, explicit system configuration is required so that the product does not assume the default security policy. See the subsequent bullets for more information.
- Define a system wide security policy using a feature configuration option. Find the Feature Configuration record for the **External Messages** feature type. (It may need to be defined if it does not exist). Choose the option type **Default security policy** and define an appropriate value. If your implementation wishes for the policies to be attached at the Java EE Web Application Server, define this option type with an option value of `<none>`.
- Attach a security policy to the IWS via a Web Service Annotation. The base product provides annotation types that support the standard WS-Policy (`F1POLICY`) and OWSM Security Policy (`F1-OWSM`). No base annotation is supplied by the product for either annotation type.

  If your implementation wishes for the policy of a particular IWS to be attached at the Java EE Web Application Server, define a special annotation for the `F1POLICY` annotation type and configure the **uri** parameter value to `<none>`.

> **Note**
>
> Refer to WebLogic documentation for more information on supported security policies.

> **Note**
>
> In order to use the OWSM Policy, additional system configuration is necessary. Contact your system administrator to confirm if your implementation supports OWSM.

Inbound SOAP Web Service Deployment

A Inbound SOAP Web Service must be deployed to the Java EE Web Application Server in order for it to be available to the Web Service Clients to access the system. Refer to Deploying SOAP Web Services for more information.

- Deploying XAI Inbound Service via IWS

Deploying XAI Inbound Service via IWS

For implementations using XAI inbound services for external messages, the product recommends moving to the inbound web service mechanism, which uses the Java EE Web Application Server to communicate with the product rather than the XAI servlet.

For XAI inbound services that use the **Business Adapter**, it is straight forward to move to IWS because the configuration is similar. In both cases, the service is configured to reference a business object, business service or service script. The associated WSDL for each record is similar. Changing the interface for the incoming message to use IWS instead of XAI inbound services is similar.

However, for XAI inbound services that use the **Core Adapter**, these services reference an underlying "page service" in the product. For these services, the Request and Response schemas for the XAI inbound service were created using the Schema Editor. In order to support calling an underlying "page service" in IWS, first a business service must be created to

Copyright © 2026, Oracle and/or its affiliates.

reference the page service (if one doesn't already exist). However, the resulting schema for the business service is different from the Request and Response schemas related to the XAI inbound service. Moving this functionality to IWS using business services requires changes to the format of the incoming messages.

Moving all incoming messages over to use IWS instead of XAI is the product recommendation. However, to aid in implementations that have many integrations in place using the XAI inbound services that use the **Core Adapter** (or any adapter whose message class is **BASEADA**), the product provides the ability to deploy these types of XAI inbound services to the Java EE Web Application Server along with the Inbound Web Services.

To take advantage of this capability, you must define a feature configuration option. Under the **External Messages** feature configuration type, the **Support XAI Services via IWS** is used to indicate if this feature is supported. Setting the value to **true** turns on the feature. If no option is defined for that option type, it is equivalent to setting the value to **false**.

When the system is configured to support XAI services via IWS, the Inbound SOAP Web Service deployment includes XAI inbound services (that are configured with an Adapter that references the **BASEADA** message class). The deployment portal will also include a zone showing the deployment status of these XAI Inbound Services.

> **Note**
>
> There is no support for XAI inbound services via REST, only via inbound SOAP web services.

Configuring Inbound Web Service Options

This topic describes the configuration needed for using inbound web services.

- Configuring SOAP Inbound Web Service Options
- Configuring REST Inbound Web Service Options
- Maintaining Inbound Web Services
- Maintaining Inbound Web Service Operations

Configuring SOAP Inbound Web Service Options

This topics in this section describes the configuration needed for using inbound SOAP web services.

- Technical Configuration
- Maintaining Web Service Annotation Types
- Maintaining Web Service Annotations
- Maintaining XAI Inbound Services

Technical Configuration

In order to use inbound web services, there are tasks a system administrator must perform.

Refer to the Server Administration Guide for technical details of each of these processes.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Web Service Annotation Types

The product provides some base annotation types. Refer to the metadata for more information. If your implementation wishes to define additional annotation types, use the Web Service Annotation Type portal. Open this page using **Admin**, and then **External Message**, and then **Web Service Annotation Type.**

You are taken to the query portal where you can search for an existing web service annotation type. Once an annotation type is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> Use of custom policies should only be considered if the policies supplied by the Java EE Web Application Server are not sufficient for your implementation's needs.

> **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service annotation type code. Refer to System Data Naming Convention for more information.

The **Web Service Annotation Type** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

The system supports the ability for an IWS record to refer to multiple policies. In this situation, the annotation type for the policy should include a reference to a parent annotation type so that the system can properly build the array of annotations.

Maintaining Web Service Annotations

If your implementation wishes to define annotations, use the Web Service Annotation portal. Open this page using **Admin**, and then **Integration**, and then **Web Service Annotation.**

This is a standard All-in-One portal.

> **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service annotation code. Refer to System Data Naming Convention for more information.

Maintaining XAI Inbound Services

XAI Inbound Service is a legacy mechanism for exposing system services to external systems as web services. The current supported mechanism for defining web services is Inbound Web Services, which should be used for any new web service.

Copyright © 2026, Oracle and/or its affiliates.

Some framework based products and existing implementations may have existing XAI inbound services that may need to be viewed. For these services, the product supports deploying them as SOAP inbound web services for execution. Refer to Deploying XAI Inbound Service via IWS.

The following sections describe basic information about the maintenance pages. Note that some of the information is related to legacy functionality that is no longer supported.

XAI Inbound Service - Main

To view an inbound service, open **Admin**, and then **XAI**, and then **XAI Inbound Service**.

Description of Page

XAI In Service Name is used in the system to identify the service. The service name is also the first XML element after the <Body> element in the XML request/response document. The **XAI Service ID** serves as the primary key.

Owner indicates if this XAI inbound service is owned by the base package or by your implementation (**Customer Modification**). This information is display-only.

The **Adapter** defines the interface with the target application server.

If adapter for this service invokes a system service, then the appropriate **Service Name** is visible.

If adapter is the **Business Adapter** then **Schema Type** and **Schema Name** reference the object to invoke.

Web Service Category is visible if the XAI inbound service is linked to one or more web service categories.

Use the **Description** and **Long Description** to describe the service.

Check the **Active** switch if this service is enabled and available for execution.

Post Error is not applicable.

Trace is not applicable.

Debug is not applicable.

Schema Definitions

> **Note**
>
> Request Schema and Response Schema are not applicable to services invoking schema-based objects. They do not appear when the **Business Adapter** is used.

The next two properties define the request and response XML schemas. The schemas are SOAP compatible. The schema XML files are expected to be stored in the Schemas Directory on the Web server running the XAI server.

The **Request Schema** is the XML schema defining the service request. The request sent to the server must adhere to the schema definition.

The **Response Schema** is the XML schema defining the service response. The response generated by the XAI server corresponds to the response schema definition.

Copyright © 2026, Oracle and/or its affiliates.

The same service may perform several actions on a business object. Use the **Transaction Type** to define the default action performed by a service. The transaction type can be provided when invoking a service, by dynamically specifying a transaction type attribute on the Service element of the XML request. This field may take the following values: **Read**, **Add**, **Change**, **Update**, **Delete**, **List** and **Search**.

> **Note**
>
> The difference between **Change** and **Update** is that for **Change**, all field values must be passed in with the request. Field values that are not passed in to the request are set to null. For **Update**, you need only pass the primary key field values and the values of the fields to be updated. All other fields retain their existing values.

Services, which perform a Search, may allow searching based on different criteria. When the Transaction Type value is **Search**, use the **Search Type** to define the default search criteria. The possible values are **Main**, `Alternate1`, `Alternate2`, `Alternate3`, `Alternate4`, `Alternate5` and `Alternate6`.

> **Note**
>
> This is a default definition only and it may be overridden at run time when the service is invoked. To override the search type at run time, you should specify the searchType attribute on the Service element of the XML request.

XSL Transformation Definitions

Sometimes, the XML request document does not conform to the request schema, or the response document expected by the service requestor is not the one generated by the adapter. In such cases the request and/or the response documents must be transformed. The XAI server supports transformation through XSL transformation scripts. Transformation scripts may be applied to the request before it is passed to the adapter or applied to the response document before it is sent to the service requestor.

The **Request XSL** is the name of the XSL transformation to be applied to the request document before processing it. The transformation is usually required when the incoming document does not correspond to the XAI service request schema therefore it has to be transformed before it can be processed by the adapter.

The **Response XSL** is the name of the XSL transformation to be applied to the response document when the requester of the service expects the response to have a different XML document structure than the one defined by the response schema for the service.

Click the **WSDL URL** hyperlink to launch a separate window that contains the WSDL definition for the inbound service. Note that the server name and port number for the URL are built using a setting in the common properties file using the XAI HTTP Caller URL setting.

Configuring REST Inbound Web Service Options

This topics in this section describes the configuration needed for using inbound REST web services.

- Owner Configuration Lookup
- Resource Category Lookup

Copyright © 2026, Oracle and/or its affiliates.

Owner Configuration Lookup

Refer to Inbound REST Web Service for an overview of REST IWS functionality.

When generating the URL for a REST web service, part of the URL is taken from the owner of the Inbound REST Web Service record. The component to add to the URL is defined in this lookup using the URI Component field. The values are provided by the product.

To view the values, navigate to the Extendable Lookup portal. Search for and select the **Owner Configuration for REST Services** business object.

Resource Category Lookup

Refer to Inbound REST Web Service for an overview of REST IWS functionality.

Each Inbound REST Web Service is associated with a Resource Category. The categories are defined as an extendable lookup. The REST URL is built using a component from the IWS record's resource category.

Navigate to the Extendable Lookup portal. Search for and select the **Resource Category** business object. Define values as needed. Typically the category is related to a logical grouping of entities, or maintenance objects, referred to as a 'resource' in REST nomenclature. In some cases, an edge application may have already delivered appropriate resource categories for your use.

Populate the appropriate URI component to include when building the URL for IWS records linked to this resource category.

Maintaining Inbound Web Services

Inbound Web Services are used to define a specific message that your implementation will receive from an external system and provides configuration needed to process the inbound message.

The product provides support for REST and SOAP services and provides several inbound web services out of the box.

For inbound SOAP web services, by default no annotations are defined for the base inbound web services. You may modify the message options or the annotations for any base IWS record. In addition, you may define additional IWS records for other incoming messages supported by your implementation.

To view an inbound web service, navigate using **Admin**, and then **Integration**, and then **Inbound Web Service.** You are brought to a query portal with options for searching for inbound web services.

Once an inbound web service has been selected, you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the inbound web service code. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

Copyright © 2026, Oracle and/or its affiliates.

- **Inbound Web Service.** This zone displays the configuration information for the record and its operations.
- **Operations.** This zone lists the operations defined for the selected inbound web service. The zone is only visible for a REST inbound web service. You may navigate to the Inbound Web Service Operation portal to further describe the operation's web service schema.
- **Help Text Fields.** This zone maintains help text information for fields across all operations of the selected inbound web service. The zone is only visible for a REST inbound web service.

Refer to Inbound SOAP Web Services for specific information about Operations for SOAP records.

Refer to Inbound REST Web Services for specific information about Operations for REST records.

Note that in addition to standard actions available on this portal, there is also a special **Add to Category** button in the page action area. Click this button to link an inbound web service to one or more web service categories.

Maintaining Inbound Web Service Operations

This portal is used to maintain a REST inbound web service operation. You can access the portal from the **Operations** zone on the Inbound Web Service portal for a REST service.

Refer to Inbound REST Web Services for specific information about Operations for REST records.

The following zones may appear as part of the portal's **Main** tab page

- **IWS Operation.** This zone provides information about the selected operation.
- **Schema.** This zone describes the request and response schema as defined by the operation's web service schema. The zone is only visible when an operation schema is explicitly defined.
- **Schema Extensions.** This zone manages web service schema extensions that correspond to extensions made to the operation's internal schema. This zone is only visible if the underlying schema was extended using dynamic data area extension functionality. Each owner that extended the internal schema may provide web service schema information for their extension elements.
- **Options.** This zone lists various options for the operation such as request and response samples. The zone is visible when such options exist for the operation. Note that options are maintained as part of operation record.

The following zones may appear as part of the portal's **References** tab page

- **IWS Operation References.** This zone lists references to the current web service operation. The list includes other inbound web service operation records that reference the current operation via their web service schema. For a GET operation, the list also includes the maintenance object or business object that explicitly references the operation using a **GET Operation** option.
- Web Service Schema Nodes and Attributes

Web Service Schema Nodes and Attributes

A REST inbound web service (IWS) operation invokes a single schema based object, either a Business Object, a Business Service or a Service Script. The object has one schema which by default acts as both the Request and the Response schema. The product provides the ability

Copyright © 2026, Oracle and/or its affiliates.

to define an explicit schema for a web service operation that represents a view of the underlying internal schema for an external facing consumer. Note that the operation schema is a strict view of the internal schema and as such does not introduce structural changes.

While optional, defining an operation schema has the following benefits:

- Declaring the **usage** of an element as to whether it is part of the Request schema, Response schema, both or neither.
- Assigning an **external element name** to an internal element. Useful to avoid cloning of internal services for this purpose only.
- Better management and visibility of **GET operation** references.
- Support HATEOAS style structures outside of the internal schema. This allows for seamless internal processing.
- Providing help text for individual elements.

The following documentation is a complete list of the attributes available to you when constructing an IWS operation schema.

Operation Schema Definition

The following attributes defines the external facing operation schema that establishes the request and response schema for the operation.

Mnemonic	Valid Values	Description	Examples
usage=	"REQ"	Indicates that the element should be included in the request schema only. When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default.	`<customerName usage="REQ" />`
	"RESP"	Indicates that the element should be included in the response schema only. When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default.	`<customerId usage="RESP" />`
	"BOTH"	Indicates that the element should be included in both request and response schema definitions. This is the default value When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default.	`<customerName usage="BOTH" />`
	"EXCL"	Indicates that the element should not be included in either the request or the response schema definitions. When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default.	`<httpMethod usage="EXCL" />`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples

mapTo=	"internal element name"	Maps a web service element to an internal element. By default the internal element name is assumed to be the same as the web service element name. Use this attribute to assign an external facing element name in the operation schema and map it to a corresponding internal element.
		
		Note that this is not an XPath expression but rather a reference to a valid element name that corresponds to the structure of the expanded internal schema.
		
		The owner of the element must be the same as the owner of the web service schema. If an operation's underlying schema was extended (using dynamic data area extension) then the extending owner may only reference their elements on their web service schema extension for the operation.
			Internal schema element:
			
			`<toDoPriority .../>`
			
			Operation schema element:
			
			`<priority mapTo="toDoPriority"/>`

role=	"FKGP"	Establishes a group structure that represents a foreign key to an entity.
		
		The element itself does not exist in the internal schema. It groups together the foreign key value element(s) along with a corresponding `_link` element that at run time provides the endpoint URL of that entity's GET operation. Consumers of the web service may use the link to get the data for the entity.
			Internal schema element:
			
			`<user .../>`
			
			Operation schema group:
			
			`<user role="FKGP">`
			`  <user>`
			`  <_link getOperation="mo:'USER';pk1:user;"/>`
			`</user>`

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"COLL"	Establishes a group structure that represents a collection of entities that corresponds to an internal service list. The element itself does not exist in the internal schema. It groups together the internal list mapped to a _data element along with a corresponding _link element that at run time provides the endpoint URL for the collection's GET operation. Consumers of the web service may use the link to get a more detailed set of information for each entity in the list. By default all the elements in the referenced list are included in the operation schema. You may explicitly reference and define attributes for specific elements as needed.	Internal schema element: <alerts type="list"> <alert/> <startDate /> <endDate /> <version /> </_data> </alerts> Operation schema group: <alerts role="COLL"> <_link getOperation="..."/> <_data mapTo="alerts"> <version usage="EXCL"/> </_data> </alerts>
getOperation=	"expression" Expression components: • mo:'<maintenance object>'; • pk1:<relative xpath to an element in the internal schema>; • pk2:...; • pk3:...; • pk4:...; • pk5:...; Parameters pk2–5 are required to match the maintenance object's prime key definition.	Only valid on _self and _link elements. The attribute is used to reference the default GET operation for an entity as defined by its maintenance object or business object. The application determines the GET operation to use as follows: • If the entity's business object references an operation using the **GET Operation** option then it is used. • If the entity's maintenance object references an operation using the **GET Operation** option then it is used. • If a GET operation is found that references the maintenance object then it is used. If a GET operation cannot be determined for the entity a standard text indicating that a link is not available is returned instead.	<_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Valid Values	Description	Examples
	"expression" Expression components: • `iws:`'<inbound web service name>'; • `operation:` '<operation name>'; • `parms:` [<external reference as defined on the GET operation record>:<relative xpath of an element name in the internal schema that holds the value>; <another external reference name>:<another relative xpath>;...];	Only valid on _self and _link elements. The attribute is used to reference a specific GET operation along with the information needed to compose its path parameters.	`<_self getOperation="iws:'F1-ToDoEntry'; operation:'getToDoEntry'; parms:[toDoEntryId:toDoEntryId;]"/>`

Specialized Elements

The operation schema definition makes use of specialized element names in the style of the Hypermedia as the Engine of Application State (HATEOAS) standard. These elements should only be defined by the operation schema as they are not relevant to nor should be managed by the internal operation service.

The following list describes each element and how it is used.

Element Name	Description	Examples
`<_self ... />`	Response payloads may include a "_self" element that contains the endpoint URL of the GET operation related to the entity returned in the response. This is an optional element and only one may exist in an operation schema. Refer to `getOperation=` schema definition attributes for more information.	`<_self getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>`
`<_link ... />`	Response payloads may include foreign keys and for those entities, the response includes a "_link" element that contains the endpoint URL of the GET operation for that entity (if it exists). Refer to `role=` and `getOperation=` schema definition attribute for more information.	`<_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>`

Copyright © 2026, Oracle and/or its affiliates.

Element Name	Description	Examples
`<_data ... />`	Provides a standard reference to a list of entities in the internal schema. Refer to `role="COLL"` schema definition attribute for more information.	`<alerts role="COLL"> <_link getOperation="... "/> <_data mapTo="alerts"> </_data> </alerts>`

Documentation Attributes

The following attributes provide additional documentation for individual elements that is included as part of the web service's Open API specification.

Mnemonic	Valid Values	Description	Examples
`helpText=`	"help text field name"	Associates an element with a specific help text record defined for the Inbound Web Service. If the element is included in both request and response schema the same help text is used in both. If this attribute is not specified, and a single Help Text Field record exists for the Field associated with the internal element, that help text record is used. If more than one such Help Text Field record exists the one where the help text name is the same as the Field name is used. Note that an internal element is associated with a Field name via the mdField= attributes, if any, else via the mapField= attribute. No valid for _self and _link elements.	`<toDoEntryId usage="BOTH" helpText="TD_ENTRY"/>` In this example the inbound web service defines a help text field name "TD_ENTRY" that provides help text for this element. The same help text is included for the element in both request and response schema.
`helpTextResponse=`	"help text field name"	Associates an element that is included in both request and response schema with a specific help text record to be used in the context of the response schema. Follows same rules as the helpText= attribute.	`<toDoEntryId usage="BOTH" helpText="TD_ENTRY" helpTextResponse="TD_ENTRY_RESP"/>` In this example the inbound web service defines the following help text records: • Help text field name "TD_ENTRY" describes the element sent in the request schema. • Help text field name "TD_ENTRY_RESP" describes the element populated in the response schema.

Copyright © 2026, Oracle and/or its affiliates.

Deploying Inbound SOAP Web Services

Once an Inbound SOAP Web Service is defined it is not automatically available to the Web Service Clients to access the system. The Deployment Status and the Active flag (set to true) indicate whether a Web Service is available or not. The last step is to deploy the Inbound SOAP Web Services to the Java EE Web Application Server. This deployment phase has a number of steps that are automatically performed when a deployment is initiated:

- The Web Service files are generated and policies are attached.
- The WSDL is generated with appropriate annotations and enumerations.
- The necessary Java stub code to implement the Web Service in the Java EE Web Application Server is generated and compiled.
- The Web Services are built into a valid Web Application Archive (WAR) file.
- Optionally, the newly created Web Services WAR file is deployed to the Java EE Web Application Server. This can also be done manually for clustered deployments, if desired.

There are two methods available for deploying inbound SOAP web services:

- Deployment at the command line using the **iwsdeploy[.sh]** command as outlined in the Server Administration Guide. This method is recommended for native installations and production implementations.
- Deployment using the Inbound Web Service Deployment portal. This method is only supported in development (non-production) environments.

Inbound Web Service Deployment Portal

To use the online Inbound Web Service Deployment portal, navigate using **Admin**, and then **Integration**, and then **Inbound Web Service Deployment**.

The following sections describe the base zones that are provided on the portal.

Deploy Inbound SOAP Web Services

The Deploy Inbound SOAP Web Services zone provides information about the last deployment. If the region is a development (non-production) region you may use the **Deploy** button to deploy or re-deploy inbound web services. All inbound web services whose Active switch is Yes will be deployed. All whose active switch is No will be undeployed.

> **Note**
>
> When an Inbound SOAP Web Service is deployed, the value of its service revision field is captured. Certain changes to configuration will require re-deployment to take effect. When any of the following changes occur, the IWS service revision value is incremented. This will cause the deployment status to show **Needs Deployment**.
>
> - Active switch is changed
> - An Annotation is added or removed
> - An Operation is added or removed.
> - The Operation Name, Schema Type / Schema Name, Request or Response Schema, Request or Response XSL for an Operation is changed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In addition, if the implementation supports XAI services deployed through IWS, the appropriate XAI inbound services will also be deployed or undeployed as required.

Deployment Status

The Deployment Status zone displays a list of inbound SOAP web services in the product, including the deployment status.

The deployment status is determined by comparing the internal Service Revision field on each IWS against the value captured at the time of deployment.

- **Deployed**. Indicates that the IWS has been deployed and no changes have been detected to the configuration.
- **Needs Deployment**. Indicates that the IWS has never been deployed or has been deployed but in the meantime, changes have been detected to the configuration that require redeployment.
- **Undeployed**. Indicates that the IWS is marked as inactive and the IWS is not found to be deployed at this time.
- **Needs Undeployment**. Indicates that the IWS is marked as inactive but the IWS is found to be deployed at this time.

If the IWS has been deployed, the View column will include a **WSDL** link allowing you to launch a separate window to view the WSDL definition.

> **Note**
>
> For cloud installations, the system supports defining the WSDL URI via a substitution value defined for the token URI: @F1_BASE_IWS_URI@. The system will use this value per the functionality described in URI Substitution. When no value is defined, the system uses a default URL, which is the appropriate URL for on-premise installations.

Use the broadcast button adjacent to any of the inbound web services listed in the zone to view the details of the IWS record. This causes the **Inbound Web Service** zone to appear. It is the same zone that appears on the Inbound Web Service maintenance portal.

XAI Inbound Service Deployment Status

The XAI Inbound Service Deployment Status zone is only visible if the feature configuration option **Support XAI Services via IWS** is configured on the **External Messages** feature type or if the system detects that there are XAI inbound services that have been deployed. (The latter condition is checked for the case where an implementation has XAI inbound services deployed and then chooses to discontinue using this functionality. After changing the feature configuration option to false, one more deployment is required to "undeploy" the XAI services.) The zone displays a list of XAI inbound services in the product that are related to page services. Refer to Deploying XAI Inbound Service via IWS for more information.

The deployment status is determined by comparing the record's Version field against the value captured at the time of deployment.

- **Deployed**. Indicates that the XAI inbound service has been deployed and no changes have been detected to the configuration.

Copyright © 2026, Oracle and/or its affiliates.

- **Needs Deployment**. Indicates that the XAI inbound service has not been deployed or has been deployed but in the meantime, changes have been detected to the configuration.
- **Undeployed**. Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** and the XAI inbound service is not found to be deployed at this time.
- **Needs Undeployment**. Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** but the XAI inbound service is found to be deployed at this time.

XAI inbound service does not have the equivalent of a Service Revision field that inbound web service has, which is only incremented when changes are made to the record that impact deployment. For XAI inbound service, the version number on the record is used. This field is incremented when any changes are made, even ones that may not impact deployment. As a result, some XAI Inbound Services may indicate "Needs Deployment" in cases where a redeployment may not be necessary. The recommendation when this occurs is to simply Deploy again to be safe.

If the IWS has been deployed, the View column will include a **WSDL** link allowing you to launch a separate window to view the WSDL definition.

Guaranteed Delivery

There are alternatives for sending messages to the system besides using inbound web services. An external system may be able to send messages to the system in a generic manner where a new web service does not need to be defined for every new type of message. These types of messages may provide a payload (the message) and the service script or business service to invoke. An example of this type of communication is a message sent from a mobile application using RESTful operations.

The external system may have no mechanism for retrying failed messages. For this situation, the product provides an algorithm that may be used to capture incoming messages that should "guarantee delivery". A servlet processing this type of message may invoke the installation algorithm - Guaranteed Delivery, passing the details of the message and an indication if a response should be returned. The algorithm is responsible for storing the message information in a table so that it can be subsequently processed.

Outgoing Messages

"Outgoing messages" is the term used to describe messages that are initiated by our system and sent to an external system. Messages may be sent real time or near real time. The system provides the following mechanisms for communicating messages to external systems.

- **Outbound Messages.** This method allows implementers to use configurable business objects to define the message format and to use scripts to build the message. If sent near real-time the message is posted to the outbound message table waiting for Oracle Service Bus to poll the records, apply the XSL and route the message. If sent real time, the message dispatcher routes the message immediately.
- **Web Service Adapters.** Using a web service adapter, an implementation can consume a WSDL from an external system and create an "adapter" record that references the URL of the external system and creates appropriate request and response data areas to expose the payload information in a format understood by configuration tools. A script may then be written to build the request information and initiate a real-time web service call from within the system.

Copyright © 2026, Oracle and/or its affiliates.

- **Send Email.** The system supplies a dedicated business service that may be used to send an email real-time from within the application.

All these methods are described in more detail in the following sections.

- Outbound Messages
- Web Service Adapters
- Sending Email

Outbound Messages

Outbound messages provide functionality for routing XML messages to an external system real-time or in near real time. In addition the functionality supports collecting related messages into a batch to then be sent to an external system as a consolidate XML message.

For each outbound message that your implementation must initiate you define a business object for the outbound message maintenance object. Using the business object's schema definition, you define the fields that make up the XML source field. These are the fields that make up the basis for the XML message (prior to any XSL transformation).

Each outbound message requires the definition of its schema by creating a business object whose schema describes the information that is provided to the external system. An XSL transformation may then be performed when routing the message to an external system.

For each external system that may receive this message, you configure the appropriate message XSL and routing information.

Because the outbound message type is associated with a business object, your implementation can easily create outbound message records from a script using the **Invoke business object** step type. Such a script would do the following:

- Determine the appropriate outbound message type and external system based on business rules
- Access the data needed to populate the message detail
- Populate the fields in the schema and use the **Invoke business object** script step type for the outbound message type's business object to store the outbound message.
- The resulting outbound message ID is returned to the script and the scriptwriter may choose to design a subsequent step to store that ID as an audit on the record that initiated this message.

The following topics provide more information about functionality supported by outbound messages.

- Polling Outbound Messages Using OSB
- Batch Message Processing
- Real Time Messages
- Designing the System for Outbound Messages
- Configuring the System for Outbound Messages
- Managing Outbound Messages

Polling Outbound Messages Using OSB

If the outbound message that needs to be sent to an external system can be sent as an asynchronous message (in 'near real time'), the process initiating the outbound message

Copyright © 2026, Oracle and/or its affiliates.

should create a record in the outbound message staging table. Oracle Service Bus (OSB), is the recommended tool to use to process outbound messages in near real-time.

Outbound messages that should be processed by OSB should be configured with a processing method defined as **SOA** for the external system / outbound message type. No other information is required for defining outbound message types that are processed by OSB.

For the OSB part of the processing, the product provides a custom transport: OUAF Outbound Message that may be used by an implementation to define messages to process and how to process them. This transport processes outbound messages in order of the priority defined on the outbound message type.

This section provides an overview of steps required to develop OSB integrations for outbound messages created by your product.

Before developing OSB integrations, a developer should be familiar with OSB development such as creating proxy services, business services, and message flow/routing. These terms are defined as follows:

Proxy Service: In OSB, a Proxy Service is the entity that processes a given type of message and routes it to a Business Service. A separate proxy service should be defined for each type of outbound message. If a given outbound message type may be routed to different external systems, it is the responsibility of the proxy service to query the external system defined on the outbound message and invoke the appropriate business service (see below). If any transformation is required prior to routing a message to a business service, it is the proxy service's responsibility to perform the transformation.

Business Service: In OSB, a Business Service is an entity that receives a message from OSB and routes it to the appropriate destination. This should not be confused with the Business Service object provided in the product in the configuration tools.

NOT_SUPPORTED

Refer to the technical reference paper *OSB Integration* for more information.

Batch Message Processing

Your implementation may be required to send messages to the same destination as a single XML file with multiple messages include. The following points describe this logic:

- The individual messages that should be grouped together must have a processing method of **batch** on the external system / outbound message type record. The appropriate batch code that is responsible for grouping the messages must also be provided.
- A separate "consolidated message" outbound message type should be configured for the external system with a processing method of **SOA**.
- When outbound message records are created for the individual messages, the batch code and current batch run number are stamped on the record.
- When the batch process runs it is responsible for building the XML file that is a collection of the individual messages. This batch process should include the following steps:
  - Format appropriate header information for the collection of messages
  - Apply the individual message XSL to each message before including the message

Copyright © 2026, Oracle and/or its affiliates.

  - Insert a new outbound message for the external system with the "consolidated message" outbound message type.
- The consolidated message is ready to be processed by Oracle Service Bus.

> **Note**
>
> **No process provided**. The system does not supply any sample batch job that does the above logic.

Real Time Messages

The system supports the ability to make web service calls, i.e. sending real time messages, to an external system.

The system supports special functionality for sending an Email message real-time. Refer to Sending Email for more information.

For other types of real-time messages, the system also uses outbound message type and external system configuration to format and route the message. When defining the configuration for real time messages, an additional step is required to define the mechanism for routing the message using a message sender. The system supports routing messages via HTTP and via JMS. Note that for HTTP routing, the system also supports sending the message using a JSON format.

Just like near real-time messages, initiating a real-time outbound message may also be done from a script. When a real time message is added, the system immediately routes it to the external system. If the external system provided a response message back, the system captures the response on the outbound message. If the outbound message type for the external system is associated with a response XSL it is applied to transform the response. In this case the system captures the raw response as well on the outbound message. Note that the outbound message BO should be configured to capture a response XML in its schema.

Any error (that can be trapped) causes the outbound message to be in a state of **Error**. It is the responsibility of the calling process to check upon the state of the outbound message and take a programmatic action. When the outbound message state is changed back to **Pending** the message will be retried.

The base package provides two business services: Outbound Message Dispatcher (`F1-OutmsgDispatcher`) and Outbound Message Mediator (`F1-OutmsgMediator`) that further facilitate making web service calls. Both business services are similar, allowing the calling script to configure the following behavior (with differences noted):

- Whether or not exceptions encountered while sending the message are trapped. Trapping errors allows the calling script to interrogate any errors encountered and take some other programmatic action.
- Whether or not the sent message is persisted as an actual outbound message record.
  - If a persisted message is desired, the recommendation is to use the Outbound Message Dispatcher. This business service creates the message using standard BO processing, relying on the outbound message logic to route the message and store the record. The message is routed after the BO pre-processing algorithm and after the record is persisted but before the BO post-processing and audit plug-ins are executed. If you need to send the outbound message ID as part of the message, refer to Capturing the Outbound Message ID in the Message for more information.

Copyright © 2026, Oracle and/or its affiliates.

- If the message should not be persisted, then the recommendation is to use the Outbound Message Mediator. As mentioned, the Outbound Message Dispatcher creates the outbound message record, relying on the outbound message logic to route the message. If it should not be persisted, it is subsequently deleted. In contrast the Outbound Message Mediator executes the BO pre-processing algorithms and then routes the message explicitly without creating a message record. It is more efficient for scenarios that don't require persistence. Note that the Outbound Message Mediator also supports persistence, but it does so by creating the records without using BO processing. This is not recommended. The Dispatcher is the better option if persistence is desired.

Refer to the descriptions of the two business services for more information.

- OAuth Configuration

OAuth Configuration

This section describes the various OAuth Grant Types supported by the system and highlights any special steps needed when using the grant type for a given use case. The information here is used to correctly configure the message sender context for senders that need this authorization type.

The following context entries apply to all OAuth grant types.

- **OAuth Access Token URL**
- Either the **OAuth Client Credentials Key Ring** or the **OAuth Client ID** and **OAuth Client Secret**. Refer to OAuth Configuration for more information about defining OAuth key rings.
- **OAuth REST API Scope**
- **OAuth Token Request Details in Body**

Client Credentials

The client credentials grant type requests the access token using the client ID and secret credentials as defined above.

JWT Assertion

The system supports both the JWT client assertion grant type and the JWT user assertion grant type. Client assertion proves the identity of the application making the request. User assertion, is for use cases where the specific application end-user is making the request.

Besides the context values applicable for all OAuth grant types, the JWT grant types also requires the following:

- **OAuth Signature Key Ring**. This defines the public / private signature key pair used to sign the assertion. Note that a prerequisite step required for contacting Oracle Identity Cloud Service is to upload a certificate based on this key.
- OAuth JWT Audience

Designing the System for Outbound Messages

The following sections describe the setup required when using Outbound Messages to communicate with an external system. The configuration walks you through the steps to configure a single external system and all its messages.

- Define the Outbound Message Business Object and Type

Copyright © 2026, Oracle and/or its affiliates.

- Real-Time Message Configuration
- Define the External System and Configure the Messages
- Outbound Message Schema Validation

Define the Outbound Message Business Object and Type

The product supplies many outbound message Business Objects along with Outbound Message Types for out of the box functionality.

In addition, implementations may need to define configuration for custom outbound messages. For each outbound message that must be sent to an external system, create a business object for the outbound message maintenance object. Using the business object's schema definition, your implementation defines the elements that make up the XML Source field (XML_SOURCE). These are the elements that are the basis for the XML message. XSL transformations may be applied to this XML source to produce the XML message.

If your integration is real-time and a response is expected, the Outbound Message business object should also map to the XML Response field (XML_RESPONSE).

- You may decide to capture the response as is and define the element as “raw”. For example

  ```
  <responseDetail mapXML="XML_RESPONSE" type="raw"/>
  ```

  In this scenario, a Response XSL may or may not be needed.
- Alternatively, if the details of the response are needed, you may define specific elements for the response. For this option, depending on how the integration is designed, a Response XSL may be needed to transform the response into the expected XML format.

Once you have your business object and schema, define an outbound message type for each unique outbound message.

Referencing an XSL

The system supports defining the XSLs as a managed content record. For backward compatibility, the system supports defining the XSL as a file in the file system. This is a system wide setting, defined using a feature configuration option. The feature type is **External Messages** and the option type is **XSL Location**. Set the value to `F1FL` to support XSL in the file system. The file location is configured using a system property. To support the XSL in managed content, no option is needed as this is the default. You may explicitly define the value of `F1MC` if desired.

Capturing the Outbound Message ID in the Message

If your integration would like to use the system generated Outbound Message ID as a unique identifier with the external system, the following configuration is needed:

- Define an element within the XML Source that should be populated with the system generated outbound message.
- Configure a BO Option on the outbound message’s business object using the Option Type **Outbound Message ID XPath** and set the Option Value to the XPath of the element defined to capture the ID.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This functionality is only applicable if the outbound message is persisted.

Support for Dynamic URLs

The product supports the ability to build a dynamic URLs. These are cases where the URL requires information determined at runtime. This is supported with a combination of BO schema definition, URL configuration and appropriate code when creating the outbound message. The following points highlight the steps needed to support this functionality.

- When defining the Message Sender's URL, use the syntax `${pathParms}` in the location of the URL where runtime information must be inserted. For example: `http://[SERVER_NAME]:1000/rest/services/${pathParms}`
- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes elements for **pathParms** and **queryParms**.
- In the code that creates the outbound message, populate the **pathParms** and if applicable **queryParms** elements with the appropriate information. The system will build the URL by plugging in the **pathParms** element value followed by a question mark, followed by the **queryParms** element value into the `${pathParms}` location in the URL.

The following is an example of run time values.

```
<dynamicConfiguration>
 <pathParms>job/1234</pathParms>
 <queryParms>firstName=John&lastName=Doe</queryParms>
<dynamicConfiguration>
```

Dynamic HTTP Headers

HTTP headers are typically static and the HTTP message senders support defining this static information as sender context. However, it is possible to support HTTP header information that may differ based on the individiual message and can only be provided at runtime. The following points highlight the steps needed to support this functionality.

- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes the list element **httpHeaders**, which supports a list of **httpHeader** elements.
- In the code that creates the outbound message, populate each 'header-key:header-value' pair that you want included in the HTTP header in the outgoing message. For example:

  ```
  move 'Lang:CHN' to "CM-RealTimeMessage/dynamicConfiguration/+httpHeaders/
  httpHeader";
  ```

Add SOAP Header Parameters at Runtime

The product supports the ability to add SOAP header parameters to an external message at runtime. The following points highlight the steps needed to support this functionality.

- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes the element **soapHeaders** .

Copyright © 2026, Oracle and/or its affiliates.

- In the code that creates the outbound message, populate the **soapHeaders** element with the self contained XML to add to the SOAP header section of the outgoing SOAP request.

Real-Time Message Configuration

Messages are routed to an external system real-time using the outbound message dispatcher or using the real-time send email business service. The system supports routing messages using HTTP and routing messages using JMS. In addition, there is a special type of message sender used for sending emails. The following sections highlights the supported real-time communication and the configuration needed for each.

Email Messages

For sending emails, the following configuration is needed:

- Define a message sender configured for email. Senders of this type should be configured with the **RTEMAILSNDR** class. The sender context is used to configure the connection information for the connecting to the SMTP server.
- This sender may be defined as the default email sender on the message option table. Alternatively, the message sender can be provided to the business service as input. Refer to Sending Email for more information.

Outbound Messages

For other outbound messages that are routed using the real-time outbound message dispatcher, a message sender must be configured to define how to route the message. The following points highlight more detail related to this configuration.

Determine the communication mechanism prior to configuring the sender.

- When routing the message using JMS, the following configuration must be defined
  - Define an appropriate JNDI Server that indicates where the JMS resources are located.
  - Define a JMS Connection to define additional configuration needed for the connection.
  - Define the JMS Queue or JMS Topic to define the queue or topic to use.
- When communicating using a JSON format, determine the method to use to convert the XML to JSON. The desired method is driven by how the request should be sent.
  - If choosing the **Base JSON Conversion** method, if XSL transformation needs to be applied prior to the conversion to JSON, then the target XML Request Schema must be defined (using a data area) so that the conversion logic knows the format of the XML it is converting. The XSL is applied to the outbound message's XML Source, resulting in the defined XML Request Schema, which is then converted to JSON. If XSL transformation is not required, then the outbound message's XML Source is converted to JSON.
  - If choosing the **Rootless JSON Conversion** method, the group element mapped to the XML source field is removed by the conversion, resulting in a rootless JSON request document.
  - If the XML source on the outbound message can be converted to JSON using an XSL, then the **XSL Transformation** method may be chosen.
  - You may also choose to convert the XML Source to JSON via the **Standard API Conversion** method (using a Jettison library). With this method, an XSL may optionally be provided. The conversion will be performed on the transformed XML.

Copyright © 2026, Oracle and/or its affiliates.

  - For the response, if the outbound message BO defines detailed elements for the XML Response field, then the JSON should be converted to this format.
    * If your conversion method is **Base JSON Conversion**, then if the JSON response cannot be converted directly to the XML Response elements on the outbound message BO, then define a Response Schema (data area) that represents the results for the base JSON conversion. In addition, define an XSL that can transform the response from the converted XML to the XML format expected on the BO.
    * If your conversion method is **Rootless JSON Conversion**, the response document is assumed to be rootless. The group element mapped to the XML response field is added by the conversion process, resulting in a well formed XML response document.
    * If the conversion method is **Standard API Conversion** or **XSL Transformation**, the standard API is used to convert JSON to XML. An XSL may be defined to convert the response to the XML Response if needed.
  - If the outbound message BO defines a "raw" element to capture the response, then a response schema and XSL are not necessary. In this case, the system will perform a JSON to XML conversion using the **Standard API Conversion** method (regardless of the conversion method defined) and the result is captured in the XML response.

- For HTTP senders including JSON senders, the system provides the following support for sending messages secured by OAuth authentication:
  - Using Oracle Web Services Manager (OWSM). The system provides a pre-configured set of policies for OAuth (`F1-0AUTH`) using a special extendable lookup. Note that the values of this policy set defines a specific CSF key repository that the implementation should use for capturing its CSF keys. In addition, there is a substitution value defined for the token URI: @F1_OAUTH2_URI@. Configure the appropriate URI for this implementation as described in URI Substitution. By default the system does not support additional policy sets to be defined. If your implementation requires a different policy set, contact support.
  - You may provide OAuth related settings as part of the message sender configuration for a REST API.

Define a message sender configured for each appropriate routing method. The invocation type should be configured as `Real-time`. For routing via HTTP, use the **RTHTTPSNDR** - HTTP sender class. For routing via HTTP with SOAP format automatically applied, use the **SOAPSNDR** - HTTP SOAP sender class. For routing via HTTP using JSON format, use the **RTJSONSNDR** - JSON sender class. For routing via JMS, use the **RTJMSQSNDR** - JMS queue sender class or **RTJMSTSNDR** - JMS topic sender class and configure the JMS Connection and JMS Queue or JMS Topic. Use the sender context to configure the required values for connecting to the appropriate destination.

> **Note**
>
> Refer to Support for Dynamic URLs for configuration needed to support dynamic URLs when sending an outbound message. There is specific configuration expected when defining the URL on the Sender in order to support dynamic URLs.

Configure the external system / outbound message type collection. The processing method defined for the external system / outbound message type must be `Real-time`.

Copyright © 2026, Oracle and/or its affiliates.

Define the External System and Configure the Messages

Define an external system and configure the valid outgoing messages and their method of communication (processing method). Refer to Outbound Messages for more information.

Outbound Message Schema Validation

The outbound messages that are generated by the system should be well formed and valid so that they do not cause any issues in the external system. To ensure this validity you may configure the system to validate messages before they are routed to their destination. Note that the validation is applied just before communication with the sender and therefore after any Request XSL has been applied.

- Define a directory where the valid W3C schemas are located using the Message option **Outbound Message Schema Location**.
- Each external system message must indicate the appropriate W3C schema to use for validation.

You may turn on or off this validation checking using the Message option **Schema Validation Flag**.

This feature is not supported for cloud implementations.

Configuring the System for Outbound Messages

The following sections describe the setup required when using Outbound Messages to communicate with an external system.

- JNDI Server
- JMS Connection
- JMS Queue
- JMS Topic
- Message Sender
- Defining Outbound Message Types
- External Systems
- Message Option

JNDI Server

If using JMS to communicate outbound messages, define a new JNDI Server. Open **Admin**, and then **Integration**, and then **JNDI Server**.

Description of Page

Enter a unique **JNDI Server** and **Description**.

Indicate the Provider URL to indicate the location of the JNDI server. A variable may be used in place of all or part of the URL. The variable must be predefined in the substitution variable property file. The value here should enclose the variable name with @. Refer to Referencing URIs for more information.

Indicate the **Initial Context Factory**, which is a Java class name used by the JNDI server provider to create JNDI context objects.

Copyright © 2026, Oracle and/or its affiliates.

JMS Connection

To define a JMS Connection, open **Admin**, and then **Integration**, and then **JMS Connection**.

Description of Page

Enter a unique **JMS Connection** and **Description**.

Indicate the **JNDI Server** to be used. Refer to JNDI Server for more information.

Use the **JNDI Connection Factory** to indicate the lookup keyword in the JNDI server used to locate the JMS connection.

Where Used

JMS Queue

To define your JMS Queue values, open **Admin**, and then **Integration**, and then **JMS Queue**.

Description of Page

Enter a unique **JMS Queue** and **Description**.

Enter the **Queue Name** as defined in the JNDI server. This is the JNDI lookup name identifying the queue.

Use the **Target Client Flag** to indicate whether or not the target client is **JMS** or **MQ**.

Select the **JNDI Server** where the queue is defined. Refer to JNDI Server for more information.

JMS Topic

To define your JMS Topic values, open **Admin**, and then **Integration**, and then **JMS Topic**.

Description of Page

Enter a unique **JMS Topic** and **Description**.

Select the **JNDI Server** where the topic is defined. Refer to JNDI Server for more information.

Enter the **Topic Name** as defined in the JNDI server. This is the JNDI lookup name identifying the topic.

Message Sender

The topics in this section describe the maintenance of a message sender

- Message Sender - Main
- Message Sender - Context

Message Sender - Main

To define a new sender, open **Admin**, and then **Integration**, and then **Message Sender.**

Description of Page

Enter a unique **Message Sender** and **Description**.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new records, carefully consider the naming convention of the message sender code. Refer to System Data Naming Convention for more information.

Set **Invocation Type** to `Real-time`.

Indicate the **Message Class** for this sender, which indicates the method used to route the message. The valid sender classes are as follows:

Message Class	Description
RTEMAILSNDR	Email sender.
RTHTTPSNDR	HTTP sender.
RTJMSQSNDR	JMS queue sender.
RTJMSTSNDR	JMS topic sender.
RTJSONSNDR	HTTP JSON sender.
SOAPSNDR	HTTP SOAP sender.

> **ⓘ Note**
>
> For backward compatibility, the sender class HTTPSNDR is also supported, but should only be used if indicated by a specific integration configuration.

Indicate whether or not this sender is currently **Active**.

For JMS related senders, indicate whether the **Message Encoding** is **ANSI message encoding** or **UTF-8 message encoding**. (Note that for all other types of senders, the Character Encoding context type is used to configure encoding.)

If the Message Class is one that connects to a JMS Queue or JMS Topic, indicate the appropriate **JMS Connection**

> **NOT_SUPPORTED**
>
> Refer to JMS Connection for more information.

If the Message Class is one that connects to a JMS queue, indicate the name of the **JMS Queue** to define where the message is to be sent.

> **NOT_SUPPORTED**
>
> Refer to JMS Queue for more information.

If the Message Class is one that connects to a JMS topic, indicate the name of the **JMS Topic** to define where the message is to be sent.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to JMS Topic for more information.

Message Sender - Context

The sender may require context information to define additional information needed by the system to successfully send outgoing messages. Open **Admin**, then **Integration**, then **Message Sender** and navigate to the **Context** page.

Description of Page

Define the **Context Type** and **Context Value**, which contain parameters for senders when more information is required. See below for the supported context values for different types of senders.

Email Sender Context

The email sender is used by the business service that sends email messages real-time.

An email sender must point to the Message Class **RTEMAILSNDR**. In addition, the following context records should be defined for senders of this type.

Context Type	Description
SMTP Host name	The SMTP server host name.
SMTP Username	The user ID used to access the SMTP server.
SMTP Password	The password used to access the SMTP server.
SMTP From Address	The email address to use in the "from" field when sending the email.
SMTP TLS Enable	Set to 'true' or 'Y' to send emails using the Transport Layer Security (TLS) protocol.
Response Time Out	The amount of time the system should wait for a real time response.

Once you have your email senders defined, configure on Message Option the default email sender to use when an explicit email sender is not defined.

HTTP Sender

An HTTP sender is one that sends messages to an HTTP server using the HTTP protocol. HTTP senders should reference a Message Class of **RTHTTPSNDR**, **RTJSONSNDR** or **SOAPSNDR**. (Note that your implementation may use a specific integration that requires the **HTTPSNDR** message class. It should only be used if indicated that the integration's documentation.)

Various parameters are required to establish a session with the target HTTP server. You specify these parameters by defining a collection of context values for the sender. A set of context types related to HTTP variables is provided with the product. The following section describes the context types and where appropriate, indicates valid values.

Before defining the HTTP sender, you need to find out how the HTTP server on the other side expects to receive the request, and in particular, to answer the following questions:

- What is the address of the HTTP server?
- Is the HTTP server using a POST or GET HTTP method?

Copyright © 2026, Oracle and/or its affiliates.

- If the server is using POST, how are message contents passed? Does it use an HTTP FORM or does it pass the data in the body of an XML message?

Context Type	Description	Values
HTTP URL1 - URL9	Used to construct the URL of the target HTTP server. Since the URL may be long and complex, you can break it into smaller parts, each defined by a separate context record. The full URL is built by concatenating the values in URL1 through URL9. You may use substitution variables when entering values for URL parts. Note that the substitution string `@XMLMSG@` may be used for **GET** calls if an XSL has been applied to convert the message into HTTP GET parameters. It is useful if the HTTP Form is not applicable to the type of message. Refer to Support for Dynamic URLs for configuration needed to support dynamic URLs when sending an outbound message.	
HTTP Method	The HTTP method used to send the message. **Note** The SOAP sender message class **SOAPSNDR** only supports the **POST** method.	**POST** or **GET**

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
HTTP Transport Method	Specifies the type of the message. You can either send the message or send and wait for a response. For either method, be sure to set the HTTP Timeout parameter.	**Send** or **sendReceive**
HTTP Form Data	Used when the message is in the format of an HTML Form (`Content-Type: application/x-www-form-urlencoded`). This context specifies the form parameters (data) that should be passed in the HTTP message. Since a form may have multiple parameters, you should add a context record for each form parameter. The value of a form parameter takes the format of *x*=*y* where *x* is the form parameter name and *y* is its value. If *y* contains the string @XMLMSG@ (case sensitive) then this string is replaced by the content of the service response XML message. The @XMLMSG@ string can be used in the HTTP Form Data or in the HTTP URL, but not in both. If a context record of this type is defined for a sender, the sender uses the HTML Form message format to send the message even if @XMLMSG@ is not specified in one of the context records. If a context record of this type is not defined for a sender, then the XML is sent with `Content-Type: text/plain`. When using POST it is put in the HTTP message body. Always required when using the GET method. If you are using the GET method and do not specify a Form Data context record, no message is transferred to the HTTP server. You may use substitution variables when entering values for Form Data.	

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
HTTP Login User	The HTTP server may require authentication. Add a context record of this type to specify the login user to use.	
HTTP Login Password	The HTTP server may require authentication. Add a context record of this type to specify the login password to use.	
HTTP Header	Sometimes the HTTP server on the other side may require the addition of HTTP headers to the message. For each HTTP header that has to be specified you should add a context record with a value having the following format *x:y* where *x* is the header name and *y* is the value for the header **Note** The product also supports Dynamic HTTP Headers that may be included in the message at runtime.	

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
HTTP Time Out	Indicates the amount of time to wait for a connection to be established with the remote system. Be sure to indicate something here, otherwise users may wait a long time if an external system is not responding.	
Character Encoding	Indicates if the message should be encoded. The sender will add to the HTTP's content type header the string `;charset=x` where *x* is the value of this context and when sending the message it will encode the data in that encoding.	`UTF-8` or `UTF-16`
Response Time Out	The amount of time the system should wait for the remote system to send a response.	

When sending messages, authorization is important. The following context types are related to authorization.

Context Type	Description	Values
Sender Security Type	Indicate the desired security type to apply. (Note that additional security types are applicable only to the SOAP Sender. See below for more information.)	**BASIC** (HTTP Basic), **TEXT** (Username Token plain text), **DIGEST** - Username Token Digest, **OWSM** - OAuth Security via OWSM, **OAUTH** - OAuth Security.
OWSM Policy Set	Applicable only if the Sender Security Type is **OWSM**. Defines the policy set to apply.	Enter a valid value for the extendable lookup Set of Policies (`F1-SetOfPolicies`). The product provides the value `F1-OAUTH` that may be used here.
OAuth Grant Type	Applicable only if the Sender Security Type is **OAUTH** . Defines the OAuth specification method for acquiring an access token. Refer to OAuth Configuration for more information on defining OAuth security types.	`client_credentials`, `jwt_user_assertion`, `jwt_client_assertion`
OAuth Access Token URL	Used to construct the URL of the target authorization server. You may use substitution variables to construct the URL.	
OAuth Client ID	The generated identifier for the current application in the external system's authorization server.	

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
OAuth Client Secret	The secret paired with the OAuth Client ID as provided by the external system's authorization server.	
OAuth Client Credentials Key Ring	Defines the key ring used to store the OAuth Client ID and secret provided by the external system's authorization server. If a key ring is specified, the OAuth Client ID and OAuth Client Secret context entries are not allowed.	
OAuth JWT Audience	This is applicable when the OAuth Grant Type is one of the JWT assertion types. If populated, it is used as the JWT audience (aud) claim. If not provided, the OAuth Token URL is used as the audience.	Example: `https://identity.oraclecloud.com/`
OAuth REST API Scope	The authorization level requested from the token end point.	
OAuth Signature Key Ring	This key ring is applicable when the OAuth Grant Type is one of the JWT assertion types. It is used to sign the JWT request sent to the Identity Cloud Service.	
OAuth Token Request Details in Body	By default, the authorization token request includes the Client ID and Secret information in the message header. This context type may be used to indicate that this information should be sent in the body of the request instead.	Set this to 'Y' to send the Client ID and Secret information as part of the body of the authorization token request.

Real-time HTTP Sender

The following context type is only applicable to senders with the **RTHTTPSNDR** message class.

Context Type	Description
Content Type	Populate a value here to override the `Content-Type` attribute in the HTTP header, which defaults to `text/xml` .

SOAP Sender

A SOAP sender is an HTTP sender that automatically adds support for the SOAP format. For this type of sender (message class of **SOAPSNDR**), besides the context values listed above, the following context entries may also be supplied.

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
Message Namespace URI	Used to indicate a specific namespace to be included in messages for this sender. Note that this value is used only when the External Message link to this sender is configured with a **Namespace Option** of **Configured on Sender**.	
SOAP Insert Timestamp	Indicate whether a timestamp should be added. Default value is 'N'.	Y or N
SOAP Expiration Delay (in seconds)	Indicate an expiration delay to add to the timestamp. The default value is 60.	
Sender Security Type	Additional security types are supported for SOAP senders for specific third party integrations.	**CIM** - Uses an appropriate namespace and adds security information in the SOAP header specific to Landis+Gyr, MS_V30 - Uses an appropriate namespace and adds security information in the SOAP header specific to MultiSpeak v3.0, MS_V41R - Uses an appropriate namespace and adds security information in the SOAP header specific to MultiSpeak v4.1.**ITRON** - Uses appropriate SOAP header information security information in the SOAP header specific to Itron.

> **Note**
>
> Refer to Add SOAP Header Parameters at Runtime for information about dynamically including SOAP Header parameters when sending a message.

JMS Sender

A JMS sender is one that sends messages to a JMS queue or JMS topic. JMS senders should reference a Message Class of **RTJMSQSNDR** or **RTJMSTSNDR**, respectively.

The following parameters are used to connect to the JMS resource.

Context Type	Description	Values
JMS Message Type (Bytes(Y)/ Text(N)	Indicates whether the data is sent as a bytes message or as a text message.	Y or N

Copyright © 2026, Oracle and/or its affiliates.

Context Type	Description	Values
JMS User Name	Enter the user name to connect to the JMS resource.	
JMS User Password	Enter the password to use to connect to the JMS resource.	
JMS Header	If JMS header values are required for the message, use this context type. For each JMS header that has to be specified, add a context record with a value having the following format *x:y* where *x* is the header name and *y* is the value.	

Application Variables

Some APIs may require application identifying variables to be included in the message payload. These values are typically available at provisioning time and need to be captured in relation to these API configurations.

> **Note**
>
> This type of context is not involved in the process of orchestrating and routing the API call. It needs to be incorporated into the payload by the application logic that composes the message using the "Get Application Variables (F1MsgVars)" service script.

The following parameters may be used to capture application variables.

Context Type	Description	Values
Application ID	Identifies the current environment for this API.	Determined at provisioning time by the API provider.
Tenant ID	Identifies the current tenant for this API.	Determined at provisioning time by the API provider.

Defining Outbound Message Types

Refer to Outbound Messages for an overview of this functionality.

Use this page to define basic information about an outbound message type. Open this page using **Admin Menu**, then **Integration**, then **Outbound Message Type.** You are brought to a query portal with options for searching for outbound message types.

> **Note**
>
> This page is not available if the **External Message** module is turned off.

Once an outbound message type has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page.

- **Outbound Message Type.** This zone provides basic information about an outbound message type. Besides the code and description, the outbound message type captures the outbound message business object and the relative priority of the outbound message. The priority is only applicable if the outbound message is processed using OSB. In addition, if the outbound message type is linked to any web service categories, they are displayed.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the outbound message type code. Refer to System Data Naming Convention for more information.

- **External System Links.** This zone is visible if the outbound message type is linked to any external systems.

Note that in addition to standard actions available on this portal, there is also a special **Add to Category** button in the page action area. Click this button to link an outbound message type to one or more web service categories.

The following zones may appear as part of the portal's **Help Text** tab page.

- **Help Text Fields.** This zone maintains help text information for fields associated with the outbound message business object.

> ⓘ **Note**
>
> Help text is not translated.

External Systems

Use this page to define an external system and define the configuration for communication between your system and the external system.

- External System - Main
- External System - Template Use

External System - Main

Open this page using **Admin**, and then **Integration**, and then **External System.**

> ⓘ **Note**
>
> This page is not available if both the **External Message** and the **Open Market Interchange** modules are turned off.

Description of Page

Enter a unique **External System** and **Description**.

Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new records, carefully consider the naming convention of the external system code. Refer to System Data Naming Convention for more information.

Use the field **Our Name In Their System** to specify the identity of your organization (i.e., your external system identifier) in the external system.

> **ⓘ Note**
>
> The workflow process profile and notification download profile are only applicable to products that support workflow and notification. They and are not visible in the product if the **Open Market Interchange** module is turned off.

If this external system sends inbound communications through notification upload staging, the type of workflow process that is created is controlled by the sender's **W/F (Workflow) Process Profile**.

If you send notifications to this external system, select a **Notification DL** (download) **Profile** that is used to define the configuration of the outgoing messages.

> **ⓘ Note**
>
> The remaining fields are not visible if the **External Message** module is turned off.

Set **Usage** to **Template External System** for external systems whose outbound message type configuration is inherited by other external systems.

If the outbound message type configuration should be inherited from a template external system, define the **Template External System**. If this field is specified, the outbound message type collection displays the data defined for the template system as display-only.

The **Outbound Message Type** accordion contains an entry for every type of outbound message defined for this external system. For each type of outbound message identify its **Outbound Message Type**.

Define the **Processing Method** for messages of this type. Valid values are **Batch**, `Real-time`, **SOA** and **XAI**.

Define an appropriate **Message Sender** if the processing method is **XAI** or `Real-time`.

Refer to Exchanging Date / Time Information With External Systems for more information about date/time formats.

Namespace Option is used when your message should include a namespace in the resulting XML. The valid option are **Standard Namespace** and **Configured on Sender**. If the value is Standard Namespace, the system will generate a namespace for the resulting WSDL with the following value: `http://ouaf.oracle.com/outbound/boName`, where boName is the business object code of the outbound message (configured on the outbound message type defined for this profile). If the value is Configured on Sender, then the value of the namespace is taken from the sender context with context type **Message Namespace URI**.

Define an appropriate **Batch Control** if the processing method is **Batch**.

Copyright © 2026, Oracle and/or its affiliates.

If the message sender is one with a message class of **RTJSONSNDR**, indicate the **JSON Conversion Method**. The valid values are **Base JSON Conversion**, **Standard API Conversion** and **XSL Transformation**. Refer to Real Time Message Configuration for more information about these methods and the additional configuration that may be applicable.

If the **JSON Conversion Method** is **Base JSON Conversion**, the **Request Schema** is enabled. Populate a data area that defines the schema for the XML format to convert the outbound message's BO schema to prior to performing the JSON conversion. Refer to Real Time Message Configuration for more information.

The **Message XSL** is the schema used to transform information from the format produced by the system to a format understood by the sender, who receives a message of this type. This is not applicable for Processing Method of **SOA**. Refer to Referencing an XSL for information about configuring where the XSL is defined.

Enter the file name of the appropriate **W3C Schema** if you want to validate the message built for outbound messages for this external system / outbound message type prior to being routed to their destination. Refer to Outbound Message Schema Validation for more information. This is not applicable for Processing Method of **SOA**. This feature is not available for cloud implementations.

If the **JSON Conversion Method** is **Base JSON Conversion**, the **Response Schema** is enabled. Populate a data area that defines the schema for the XML format that the JSON message is initially converted to. The XML is then converted to BO XML. Refer to Real Time Message Configuration for more information.

Response XSL will only be displayed when the processing method is `Real-time`. Refer to Real Time Messages for more information on how it is used. Refer to Referencing an XSL for information about configuring where the XSL is defined.

External System - Template Use

If you are viewing an external system whose usage is a **Template External System**, use this page to view the other external systems that reference this one. Open this page using **Admin**, and then **Integration**, and then **External System** and then navigate to the **Template Use** tab.

Description of Page

The tree shows every external system that references this external system as its template.

Message Option

The Message Option page defines various system settings used by the system when processing external messages.

To define options for your environment, open **Admin**, and then **Integration**, and then **Message Option**.

Description of Page

The following options are supported.

Option	Description	Option Name
Default Email Sender	This is the default Message Sender used for sending e-mails when no explicit Message Sender is specified.	`defaultEmailSender`

Copyright © 2026, Oracle and/or its affiliates.

Option	Description	Option Name
Default User	The default user is used by the system to access your product when no other user is explicitly specified.	`defaultUser`
Email Attachment File Location	This is the default location of e-mail attachment files. If not specified, the e-mail service provided with the product assumes a full path is provided with each attachment file.	`emailAttachmentFileLocation`
Email XSL File Location	This is the default location of e-mail XSL files. If not specified, the e-mail service provided with the product assumes a full path is provided to an XSL file as part of an e-mail request.	`emailXSLFileLocation`
Messages Language	The default language to use for the messages.	`language`
Outbound Message Schema Location	Enter the full path of the virtual directory where valid W3C schemas are stored if your implementation wants to validate outbound message schemas. For example: http:/localhost/cisxai/schemas.	`xaiOuboundSchemaLoc`
Schema Validation Flag	Enter `Y` to turn on schema validation for outbound messages. Enter `N` to turn this off.	`xaiSchemaValidationCheck`

Note that the following options are no longer applicable.

Option
To Do Type for Inbound JMS Message Errors
To Do Type for Outbound Message Errors
XSD Compliance

Managing Outbound Messages

Use this page to view information about outbound messages.

- Outbound Message - Main
- Outbound Message - Message
- Outbound Message - Response

Outbound Message - Main

Open this page using **Menu**, and then **Integration**, and then **Outbound Message**.

Copyright © 2026, Oracle and/or its affiliates.

Description of Page

Outbound Message ID is the system-assigned unique identifier of the outbound message. These values only appear after the outbound message is added to the database.

The **Processing Method** indicates whether this record will be processed by a **Batch** extract process or `Real-time`. (The value of **XAI** is no longer supported). The value defined on the external system / outbound message type collection populates this value.

When records are created with a processing method of **Batch**, the system sets Extract to **Can Be Extracted**. Change the value to **Not to be extracted** if there is some reason that processing should be held for this record.

For records with a processing method of **Batch**, **Batch Control** indicates the process that will extract this record. This value is populated based on the on the external system / outbound message type's value. **Batch Number** indicates in which batch run this record was extracted or will be extracted.

The **Retry Count** is no longer applicable..

The **Creation Date** indicates the date that this record was created.

The status is no longer applicable.

Outbound Message - Message

Use this page to view the XML source used to build an outbound message. Open this page using **Menu**, and then **Integration**, and then **Outbound Message** and then navigate to the **Message** tab.

Description of Page

The **XML Source** is displayed.

If a message XSL is defined on the external system / outbound message type record linked to this outbound message, the **Show XML** button is enabled. Click this button to view the XML that is a result of applying the Message XSL to the XML source.

Outbound Message - Response

Use this page to display the XML response. Open this page using **Menu**, and then **Integration**, and then **Outbound Message** and then navigate to the **Response** tab.

Description of Page

The **XML Response** and optionally the **XML Raw Response** is displayed.

XML Response displays the response data from the system called by the real-time message. If a response XSL is defined on the external system / outbound message type record linked to this outbound message, a transform is performed and the XML Raw Response displays the original, unchanged response.

Web Service Adapters

The base product provides a configuration object called Web Service Adapter that is used to help build configuration objects to allow for functionality in the system to initiate a web service call from within the system. A Web Service Adapter provides the following functionality:

- WSDL (web service description language) import. An implementer can use the WSDL import functionality to read the details of a WSDL into the system

Copyright © 2026, Oracle and/or its affiliates.

- Internal API generation. The system generates internal data areas that have two main purposes: they provide the API for custom code to define the appropriate input and they provide output data for the web service call using Oracle Utilities Application Framework schema language. In addition, the web service dispatcher uses element mapping defined in the data areas to transform the internal XML into the structure expected by the external system as described in the WSDL.
- Defines the URL needed to perform the web service call at runtime.
- Understanding Web Service Adapters
- Setting Up Web Service Adapters

Understanding Web Service Adapters

The following topics describe the system functionality in more detail.

Importing a WSDL

Configuring a Web Service Adapter starts by identifying the WSDL (the web service description language document used to define the interface) that will be provided by the external system. The following steps describe the base product functionality provided to allow a user to import a WSDL.

- Navigate to the **Web Service Adapter** page in add mode and select the appropriate base business object.
- Enter a meaningful Web Service Name and appropriate descriptions.
- Provide the URL of the given WSDL.
- Click **Import** to retrieve the details of the WSDL. The system then parses the WSDL details and populates the WSDL Service Name, WSDL Source, WSDL Port, URL and a list of Operations (methods) defined in the WSDL.
- Determine which Operations should be **active** based on the business requirements for invoking this web service. **Active** operations are those that the implementation is planning to invoke from the system. These require appropriate request and response data areas generated for them. The following section provides more information about that.
- Specify the appropriate Security Type to configure the type of security to use when invoking this web service.
- Click Save.

At this point, a web service adapter record is created in pending status. The next step is to generate the request and response data areas for the operations configured as active.

Generating Request and Response Data Areas

Each **active** operation for the web service adapter requires a pair of data areas, request and response, that represent the request and response XML messages for the operation.

The base product provides steps to generate the data areas as follows:

- As described in the Importing a WSDL section above, the operations listed in the WSDL are generated for the web service adapter and the implementer should indicate which operation to activate.
- After saving the **pending** web service adapter, the display lists all the active operations and for each one includes a **Generate** button.

Copyright © 2026, Oracle and/or its affiliates.

- After clicking **Generate** for an operation, a window appears where the names of the new Request and Response Data Areas may be defined. Click **Save** to generate the data areas.

The generated data areas provide the API for the implementer to use when implementing the web service call in an appropriate algorithm or service in the system. The data areas contain the appropriate mapping from the elements that the implementer works within the code that invokes the web services and the WSDL definitions.

To facilitate generating the request and response data areas, the base product invokes a special business service used to create the appropriate mapping. The business service is defined as a BO option on the Web Service Adapter business object. This allows an implementation to provide a custom business service to further enhance the request and response mapping where appropriate.

> **Note**
>
> **Generated data areas.** It is possible to edit and modify the generated data areas after they are created. An implementer can change element names or remove unneeded elements if desired. Manually changing the generated data areas must be done only when absolutely necessary. This is because the system is not able to validate manual changes and issues with the data areas would only be detected at run time.

Activating Web Service Adapters

The business objects provided by the base package for web service adapters include a simple lifecycle of **Pending** and **Active**. Configure the web service adapter and its data areas while in **Pending** status and activate it when it is ready to be implemented in the appropriate system functionality.

Invoking Web Services

To make a call to a web service using a web service adapter, the system has provided a Web Service Dispatcher business service (**F1-InvokeWebService**) to submit a web service call. The calling program is responsible for retrieving all the information to correctly populate the request data required by the web service call before invoking the business service.

> **Note**
>
> Refer to the detailed description of the business service for more information.

Limitations

The following points highlight limitations associated with the types of web services that the system supports:

- It is possible for one WSDL document to contain definitions for several web services. The system currently supports only one port or service per WSDL document.
- It is possible for a WSDL to support multiple message patterns. The system currently supports only request / response.

Setting Up Web Service Adapters

Copyright © 2026, Oracle and/or its affiliates.

Use the Web Service Adapter portal to define the configuration needed to communicate with an external system using a web service call. Open this page using **Admin**, and then **Integration**, and then **Web Service Adapter.** You are brought to a query portal with options for searching for web service adapters.

Once a web service adapter has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Web Service Adapter** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

> **NOT_SUPPORTED**
>
> Refer to Understanding Web Service Adapters for information about common web service adapter functionality.

Sending Email

The framework provides the ability to initiate an email from within the system. The following topics highlight the functionality available.

- Sending email "real time" using a specific business service. The framework provides a business service `F1-EmailService` that supports sending an email. The schema supports elements for all the information required to create an email real time. The SMTP information (host, user name, password and 'from' address) may be provided or may be defined on a message sender, that may be provided as input. In addition, the business service supports using a default message sender defined as a message option. Review the business service schema for information about the input elements.

  > **Note**
  >
  > **Retry setting**. An option in the system properties file allows your implementation to configure the number of times to retry (if any) if the SMTP server is unavailable. Refer to the server administration guide for more information.

- The business service supports sending attachments. The may either reference attachments from the file system (file name and content ID) or reference attachment records captured in the system's attachment table. Note that attachments in the file system are not supported in a cloud implementation.
  - For references to the attachment records, refer to Emailing Attachments for information about the location and size of attchment content and how that affects how attachments are included in the email.
  - For referencing attachments from the native file system, an entry must be defined for the file path in the File Storage Configuration extendable lookup. Refer to URI Substitution for more information on defining this configuration. The extendable lookup records are acting as a type of allowlist.
    The system supports defining a 'parent' directory in the file storage extendable lookup. When invoking F1-EmailService, providing a directory within the registered 'parent' directory is supported. For example, if you register the path D:\systemFiles\, at runtime, when invoking F1-EmailService, you can provide a file in the path D:\systemFiles\CMA.

Copyright © 2026, Oracle and/or its affiliates.

You may also use the `file-storage://XXXX` syntax described URI Substitution when providing the file location to the F1-EmailService.

> **Note**
>
> Validating attachments. If a Validate Email Attachment algorithm is plugged into the installation record, it is called to validate the attachments supplied, if applicable.

- Using an outbound message to send an email. This option allows for different variations as described in Outbound Messages.
  - Some emails may be created en masse (for example a large group of emails routed to users for a given set of To Do entries). In this case, the records can be created in the staging table for processing using OSB.
  - Messages may still be sent real time using one of two business services described in Real Time Messages. This option is an alternative to the dedicated email service described above when aspects of the outbound message functionality are needed, such as the ability to instantiate a record as an audit or to include additional logic via BO plug-ins as part of sending the email.

Web Service Category

The product provides the ability to categorize web services that are defined in the system. The term web services refers to Outbound Message Types, Inbound Web Services, and for those implementations where they are applicable, XAI Inbound Services.

A given web service may be linked to more than one category. The product supplies web service categories and most base delivered web services are configured with appropriate categories. Implementations may define new categories and may link custom web services to base delivered categories or to custom categories. In addition, implementations may link additional categories to base delivered web services. However, implementations may not remove base delivered web services from base delivered categories.

If your implementation uses the Integration Cloud Service Catalog, the integration provides the categories for each web service that is sent to the catalog.

The web service category portal supports adding and removing web services from the category. In addition, the Outbound Message Type portal and Inbound Web Service portal each provide a page action button to **Add to Category**.

- Defining Web Service Categories

Defining Web Service Categories

Refer to Web Service Category for an overview of this functionality.

Use this page to define basic information about a web service category. Open this page using **Admin**, and then **Integration**, and then **Web Service Category.** You are brought to a query portal with options for searching for web service categories.

> **Note**
>
> This page is not available if the **External Message** module is turned off.

Copyright © 2026, Oracle and/or its affiliates.

Once a web service category has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Web Service Category** zone provides basic information about a web service category.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service category code. Refer to System Data Naming Convention for more information.

Once a web service category exists, an **Add Web Services** zone is provided to find and select web services to link to the web service category.

When viewing a category that already has web services linked to it, they are visible in the zone **Included Web Services** . This zone allows a user to remove a customer owned web service from the list.

JMS Message Browser

The JMS Message Browser portal allows you to select a JMS queue and view messages currently in the queue.

In order for a JMS queue to be available on the portal, a message sender must be defined that is configured for the appropriate JMS queue with the credentials to connect to the queue.

- If your organization sends real-time outgoing messages to a JMS queue, this configuration would exist as per the details in Real-Time Message Configuration.
- If inbound web service messages are routed to the system via a JMS queue, no configuration is needed in the system. However, if you would like to view the messages in the queue in the JMS message browser portal, configuration for the JMS queues as described for outgoing messages is required.

Navigate to the portal using **Main**, and then **Integration**, and then **JMS Message Browser.**

The JMS Senders zone provides a list of configured message senders eligible for selection.

The JMS Message List zone is visible for the JMS Sender broadcast from the first zone. This zone supports selecting one or more records to delete from the queue. Use the message selector to limit the results to messages that satisfy the message selector. This uses standard JMS API message selector functionality. Refer to the zone embedded help for information about the supported syntax.

The JMS Message Details zone displays a message broadcast from the list zone.

Oracle Integration Cloud Catalog

Oracle Integration Cloud (OIC) is an offering that serves as integration infrastructure for Oracle cloud solutions. The product provides an adapter for OIC to streamline integration between your edge application and OIC.

The product provides a mechanism (referred to as an adapter) for OIC to retrieve the REST catalog and another to retrieve the SOAP catalog from a given product. The expectation is that a given implementation would use either REST or SOAP and not both, but for those implementations that do support both classes of web services, OIC can retrieve both catalogs.

Copyright © 2026, Oracle and/or its affiliates.

For the SOAP catalog, the adapter retrieves the name, the source system, the WSDL location and namespace for each web service.

For the REST catalog, the adapter retrieves the name, the source system and the OpenAPI Specification for each web service.

It is possible that not every web service supported by an edge product is managed by OIC. In order to only include the appropriate web services in the adapter, configuration is needed to identify which web services to include.

The web service catalog is used to identify the records that should be retrieved by the adapter. Separate catalogs exist for REST and for SOAP

- For SOAP inbound messages, the system supports both the use of inbound web services and XAI inbound services that are deployed via IWS. Each IWS or XAI inbound service that should be included in the catalog must be flagged in the SOAP catalog. Note that only deployed services are returned to the catalogue.
- For REST inbound messages, each IWS that should be included in the catalog must be flagged in the REST catalog.
- For outbound message, the system requires the creation of an External System that includes each outbound message type that the external system receives. For outbound messages that are integrated through OIC, the external system itself will represent OIC. Rather than identifying each outbound message type to include in the catalog, only the external system needs to be flagged. The adapter will return web service information for all the outbound message types configured for the external system. If an implementation supports both REST and SOAP, separate external systems should be created to separately group the REST and SOAP outbound messages.

> **Note**
>
> The subordinate catalog functionality is no longer applicable and will be removed in a future release.

> **NOT_SUPPORTED**
>
> Refer to Maintaining the Web Service Catalog for more information.

- Web Service Catalog Configuration

Web Service Catalog Configuration

The topics in this section describe the configuration needed in your edge application to integrate with OIC.

- Web Service Catalog Master Configuration
- Maintaining the Web Service Catalog

Web Service Catalog Master Configuration

The **Service Catalog Configuration** (master configuration) record defines several system wide settings related to integrating with the service catalog.

Copyright © 2026, Oracle and/or its affiliates.

It includes configuration for the WSDL base URL for the SOAP catalog. Note that this element supports the functionality described in Referencing URIs. Note that for the REST catalog, the system will use the URL defined in the substitution variable `F1_OPEN_API_BASE_URL`

The subordinate server section is no longer applicable and will be removed in a future release.

For more information about specific fields in the master configuration, refer to the embedded help.

Maintaining the Web Service Catalog

Refer to Oracle Integration Cloud Catalog for an overview of web service catalog functionality.

To add or remove services that are reported to the OIC catalog, navigate to the portal using **Admin**, and then **Integration**, and then **Web Service Catalog.**

The Catalog by Class zone provides the list of classes (REST and SOAP). Broadcast the web service class that you want to work with.

The Catalog zone provides a list of services that are currently in the catalog based on the web service class chosen. Users may use this zone to remove services from the catalog.

The Candidate Services zone provides a list of external systems and inbound web services that are not currently linked to the catalog. Users may use this zone to add services to the catalog. The inbound web services will be limited to those with the same web service class as the one broadcast. All external systems are shown regardless of the web service class because there is no web service class configuration on external systems.. The expectation is that implementers will create the external system to include either only REST or only SOAP services and will know which external system to include in which catalog. For the SOAP web service class, if XAI inbound services are applicable, they are also visible.

> **Note**
>
> For inbound SOAP web services and XAI inbound services, they may be selected for the SOAP catalog at any time. However, only deployed services are returned to the catalog by the OIC adapter.

XAI Documentation Note

The XAI functionality is legacy functionality and not recommended for new implementations. The help topics are no longer provided.

Integrations

The primary integration methods supported by products built on the Oracle Utilities Application Framework are (a) inbound and outbound files, and (b) inbound and outbound web services.

File Based Integrations

Files are uploaded to Object Storage or your local file system and processed via scheduled batch processes. Your specific product provides many out of the box upload processes. In addition, the system provides a plug-in driven upload process if you have specific upload requirements that are not met by base product processes. Refer to Uploading Records for more information.

Copyright © 2026, Oracle and/or its affiliates.

File-based extracts can be generated and made available for download and further processing. Several out of the box extract processes are provided by your specific product. Implementation-specific file processing and generation logic can be introduced using the plug-in driven Extract Record Processing.

Web Services Integration is supported through Inbound Web Services and Outbound Messages. Refer to those respective chapters for more detail.

This remaining topics in this chapter provide high level information about product integrations supported for all products that use Oracle Utilities Application Framework.

- Creating Application Links in External Applications
- LDAP Integration
- Oracle Identity Manager Integration
- Provision Users with SCIM Open Standard
- User Redirection
- Data Synchronization
- Operational Analytics
- Analytics Integration
- Business Flags

Creating Application Links in External Applications

When integrating Oracle Utilities applications with other external applications, you can create links from external applications to specific portals and pages within the Oracle Utilities applications (also known as "deep linking"). These links typically leverage Business Process Assistant (BPA) scripts to help navigate the user to the appropriate target portal or page and to make provided parameter values available for use on the portal or page.

> **Note**
>
> This technique may also be used within the product to launch a new browser tab from within the application when it's important to maintain the current screen. For example, launching the GIS Viewer in Oracle Utilities Work and Asset Management by creating a new browser tab.

To do this, parameters are added to the application URL by appending a question mark (?) to the end and using the syntax "key=value". If you require more than one parameter, use an ampersand (&) to separate each key=value pair. There are several predefined parameter key values recognized by the application. In addition, specific parameters understood by the script may be provided. The following list highlights the predefined parameters.

- `location`=<navigation key> - If the exact page to launch is known, the navigation key value of that page can be part of the URL using 'location='. For example, if an approval process has sent an email to a user to approve an adjustment, the link configured in the email may include the adjustment's navigation key.
- `locationNavOpt`=<navigation option> - If the exact page to launch is known, the navigation option value of that page can be part of the URL using 'locationNavOpt='. For example, if an approval process has sent an email to a user to approve an adjustment, the link configured in the email may include the adjustment's navigation option. Assumption is

Copyright © 2026, Oracle and/or its affiliates.

that the navigation option provided is for a portal unless the URL explicitly defines "portal=false".

- `script=`<BPA script name> - In almost every use case, a script is needed to help navigate to the correct page and display the correct information. Even if a location is known and provided in the URL, if there is context data provided as a parameter, it is the responsibility of a script to move that parameter data from the URL into the page data model, so that the page can access the information and display the correct record. In cases where the target page is dynamic based on information provided by parameters, the script may be used to determine what page to launch.
- `portal=true` - When using the location parameter where the target page is a portal page and passing context data, use this parameter. The allows the system to know that the parameter data must be pushed into portal context.
- `initNav=false` - By default, if a script is provided but the location attribute is not provided, the system navigates to the user's home page prior to executing the script. If the script itself includes a step to navigate to a target page as one of its initial steps, the navigation to the home page is unnecessary and may degrade performance. This optional attribute may be included in the URL to bypass the home page.

> **Note**
>
> The system still requires a page to be launched for technical reasons. A blank portal with no zones is used for this purpose. Users may see this portal (called **Launching Application**) briefly before the navigation initiated by the script. In addition, this is the portal that the user will remain on if there are any errors in the script or if the script does not navigate anywhere.

- `minSidebar=true` - This option minimizes the sidebar area as part of launching the application. (The user will be able to maximize the sidebar themselves if needed). You would use this option to potentially improve the speed at which the application is launched. Whether this is useful will depend on the use case. If the reason for launching the application is to bring the user to a specific record to view details about that record, the sidebar may be providing valuable information about the record. In this case, it may be correct to display the sidebar by default (and not include this parameter in the URL). On the other hand if the use case is that a user is already in the application and clicks a link that opens a new browser tab to show the user specific detail about a chosen record, while keeping the original page open on the first tab, the sidebar may contain the same information already visible on the original tab. In this case, minimizing the sidebar may be beneficial.

> **Note**
>
> For backward compatibility, the system still supports the original parameter `minDashboard=true`, which has analogous functionality.

For example, the following URL illustrates launching the script `CM-StopAutoPay` for a given account, assuming the application URL for launching the system is http://system-server:1234/cis.jsp:

- **http://system-server:1234/cis.jsp?script=CM-StopAutoPay&ACCT_ID=1234512345&location=accountMaint**

Copyright © 2026, Oracle and/or its affiliates.

It doesn't matter in which order the parameters are provided. The system processes them in the correct order. For example, the following examples are processed by the system in the same way:

- **http://system-server:1234/cis.jsp?ACCT_ID=1234512345&script=CM-StopAutoPay&location=accountMaint**
- **http://system-server:1234/cis.jsp?ACCT_ID=1234512345&location=accountMaint&script=CM-StopAutoPay**

These parameters are kept in a common area accessible by any script for the duration of the session. To use these parameters on a script you may reference the corresponding `%PARM-<name>` global variables. In this example, after the system is launched any script may have access to the above account ID parameter value by means of the `%PARM-ACCT_ID` global variable. Also note, these parameters are also loaded into temporary storage (to continue the example, there would also be a temporary storage variable called `ACCT_ID` that holds the passed value).

Determining the Application URL

When building the link from within application logic, the service script `F1-EnviURL` may be used the retrieve and build the application URL information.

Navigate to a Given Record's Maintenance Portal

The product provides a script `F1-GotoPrtl` (Navigate to portal for an MO and key values) that may be used for any use case that simply navigates to the maintenance page for a given record and displays that record. This script is only applicable to records that are governed by a business object (and define a navigation option as a BO option). It requires the parameters `mo=` with the maintenance object code of the record and the parameters `pkValue1=` through `pkValue5=` for providing the key values. It looks up the appropriate portal navigation option from the record's BO, populates keys into the 'page data model' and navigates to the portal.

It is also recommended to include the `initNav=false` attribute when building a URL with this script. The example below navigates to the appropriate Migration Data Set portal for the given ID's business object.

- **http://system-server:1234/cis.jsp?script=F1-GotoPrtl&pkValue1=1234512345&mo=F1-MIGRDS&initNav=false**

LDAP Integration

Organizations commonly use a Lightweight Directory Access Protocol (LDAP) security repository as a source of security credentials. The system provides support for importing users and groups from an external LDAP repository into the product to populate Users and User Groups in the system. Once imported, all user and group functions are available. You can resynchronize your LDAP users and groups at any time.

> **Note**
>
> **Import only**. The system currently supports importing LDAP information. Exporting users and groups from the system to LDAP is not provided.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional configuration. When importing new users and / or groups, additional configuration is needed in the base product. For example, after importing a new user group and its users, the user group configuration should be updated to define the valid application services for the user group. After importing a new user, additional configuration may be needed on the user such as valid To Do Roles, valid Home Page, etc.

> **NOT_SUPPORTED**
>
> Refer to Users and User Groups for more information about the application security and what it controls.

This section the functionality provided in the framework application that supports LDAP. Refer to the *LDAP Integration* technical reference paper for more information about typical steps related to the full integration.

- LDAP Integration Overview
- Configuring LDAP Integration

LDAP Integration Overview

This topic provides a high level overview of the integration process.

At a high level, the base product provides a process to import user group and / or user definitions from and LDAP repository. This is a one way integration.

- When importing a user, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user group, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user, its user group links will be updated as per the information in the import file. In addition, if there are any user groups linked to the user that are not found in the system, they will be added (however, the other users linked to that group in the LDAP repository will not be added as part of this step).
- When importing a user group, its user links will be updated as per the information in the import file. In addition, if there are any users linked to the user group that are not found system, they will be added (however, the other user groups linked to that user in the LDAP repository will not be added as part of this step).
- The import will not cause any deletions of the User or User Group to occur.

A Batch Process Initiates the Import

A batch process is used to initiate the import of information from the LDAP repository. F1–LDAP may be submitted ad hoc or may be set up in a scheduler to periodically re-sync the information from the LDAP repository into the application.

The batch process uses parameters to define how to connect to the LDAP repository. In addition, parameters are used to indicate which user or group is being imported.

Copyright © 2026, Oracle and/or its affiliates.

Adjusting Data to Import

The system provides several mechanisms for adjusting data that is being added to the system:

- There is an **LDAP Import Preprocess** algorithm plug-in spot on the installation record. Algorithms plugged in here are called by the batch process prior to the add or update of any records. It may be used to make adjustments to the data before doing updates in the application.
- Specifically for creating or updating Users, the `F1-IDMUser` business object is used to add and create the user. The standard BO Preprocessing algorithm plug-in spot may be used to adjust data prior to creation. Note that the system does not support defining a characteristic to support a template user that can aid in populating configuration for a new user. However, this integration can take advantage of using a default template user as an algorithm parameter. Refer to Template User for more information.
- The LDAP mapping file supports some attributes to perform simple modifications to data.
  - The **transform** attribute supports values to truncate values or to convert data to upper case.
  - The **autoGenerate** attribute is specific to the User ID field. Setting this to true will trigger code that will automatically populate the User ID based on the user's name. Refer to LDAP Mapping for more information.

Performing Additional Processing After Import

The system provides a plug-in spot on the installation record called **LDAP Import**. Algorithms plugged into this spot are called after users or user groups have been added or updated. It may be used to perform any extra processing that may need to be executed.

Configuring LDAP Integration

To interface the LDAP based security repository with the authorization component of the Oracle Utilities Application Framework product the following must be performed:

- The location and port number of the LDAP based security repository must be defined to in the JNDI Server.
- • The LDAP based security repository must be mapped to the Oracle Utilities Application Framework security model. This mapping is expressed as an XML file containing the LDAP query, rules and defaults used in the transformation.
- • The mapping file must be configured on the `F1-LDAP` batch job.

Define the JNDI Server

The first step in the configuration process is to define the location of the LDAP based security repository server so that the interface can connect to the physical attributes of the interface. This is done by creating a JNDI Server.

> **Note**
>
> The LDAP server is strictly not a JDNI source but is treated as a JNDI source for the integration.

Enter a reasonable JNDI Server name and description.

Copyright © 2026, Oracle and/or its affiliates.

Populate the **Provider URL** using the format `ldap://<hostname>:<portnumber>` where `<hostname>` is the host of the LDAP server and `<portnumber>` is the port used for the interface.

For the **Initial Context Factory**, the interface uses the standard `com.sun.jndi.ldap.LdapCtxFactory` provided with java for the LDAP interface. If your vendor supplies a custom context factory it may be used. Refer to the documentation provided with your LDAP based security repository for further information.

Define Mapping

The critical component of the interface is a file that describes the mapping between the LDAP based security repository and the system's security model. This file contains the mapping, rules and queries used by the LDAP batch program to provide the interface. The LDAP batch job includes the reference to the mapping file as a parameter. Refer to LDAP Mapping for more information on defining the mapping file.

Configure LDAP Batch Process

At this point, many parameters for the `F1-LDAP` batch control can be updated with system wide configuration.

- JDNI Server, User and Password may all be configured appropriately. Note that it is recommended that the **Security** setting for the Password be set to **Encrypt**.
- The **LDAP Configuration File** should be populated with the name and location of the LDAP Mapping file.
- If the LDAP service has any limitation to the number of objects that may be imported, configure the **LDAP Query Page Size** parameter to enable querying.

> **Note**
>
> **Group and User Parameters.** The assumption is that the Group or User input parameters are specific to a given import request and as such would not be populated as part of a configuration step.

> **Note**
>
> **L2 Cache**. The LDAP Import batch process requires the L2 Cache to be disabled since it needs to perform some updates in the outside of the worker threads. Any environment using LDAP Import must set `spl.runtime.batch.L2CacheMode=OFF` in the `threadpoolworker.properties` file. It is recommended to run the LDAP import in its own dedicated threadpoolworker.

- LDAP Mapping

LDAP Mapping

An LDAP repository consists of multiple entries. Each entry represents an object in the directory that is identified by a Distinguished Name (DN) and may contain one or more attributes. In a typical LDAP repository there is usually an entry for users and an entry for groups. The connection between users and groups may be implemented in two different ways:

Copyright © 2026, Oracle and/or its affiliates.

- The users belonging to a group are defined in a special multiple-value attribute on the Group entry.
- The groups to which a user belongs are defined in a special multiple-value attribute on the User entry.

The mapping between LDAP security objects and base security objects is stored in an XML document that can be processed by the LDAP import batch job. As part of setting up your system for LDAP import, you need to define this mapping. The base package provides a sample mapping file called `ldapdef.xml` that can be used as a starting point and changed per your business requirements and your particular LDAP repository.

Once you have defined the mapping XML document, this is configured as a parameter in the `F1-LDAP` batch job.

The XML structure:

- The **LDAPEntry** element maps the LDAP entries to system objects (User or Group). The mapping file must contain one and only one LDAPEntry element for User and one for Group.
- The **LDAPCDXAttrMapping** element within the LDAPEntry element maps attributes in the LDAP entry to attributes in the system object.
- The **LDAPEntryLinks** element describes objects linked to the LDAP entry. When mapping the user entity you need to describe how the groups the user belongs to are retrieved. When mapping the group entity you need to describe how the users contained in the group are retrieved.

The following table describes the attributes to define for each element.

Element	Attribute	Description
LDAPEntry	name	The name of the LDAP entry: - **Group** - **User**
	baseDN	The base distinguished name in LDAP for this entry.
	cdxEntity	The name of the base product entity to which the LDAP entry is mapped: - **Group** - **User**
	searchFilter	An LDAP search filter that is used to locate LDAP entries. A %searchParm% string in that filter is replaced by the value from the user or group parameter from the `F1-LDAP` batch job submission.
	Scope	Sets the scope of the search. Valid values are: - onelevel (the value normally used) - subtree
LDAPCDXAttrMapping	ldapAttr	The name of the LDAP attribute to be mapped. Note that this may be referenced more than once to allow one LDAP element to map to multiple base product elements. For example, if an email address should be used both for the Login ID and the Email Address.

Copyright © 2026, Oracle and/or its affiliates.

Element	Attribute	Description
	cdxName	The name of the base product attribute to be mapped. For User, this is the element within the F1-IDMUser business object. For Group, this is either the ‘group’ or the ‘description’.
	default	The default value that will be assigned to the element referenced in the cdxName attribute when one of the following occurs: - The LDAP attribute contains a null or empty value - The LDAP attribute does not exist or is not specified. Default values are applied only when creating a new entity and are not applied to updated entities.
	autoGenerate	Set this to **true** in order to turn on auto generation of the user ID. If this is true, the system will define the user id as <first initial of first name>+<last name> all uppercase, to a maximum of 8 digits. If an existing user is found for the generated ID, a number will replace the eight digit (or be appended to the end). The system will increment the number until a unique ID is found.
	transform	Use this attribute to indicate if a transformation of the data should occur. Valid values: `uppercase|truncate`. Note that this attribute should not be used in conjunction with the autoGenerate attribute.
LDAPEntryLink	linkedToLDAPEntity	The name of the linked entity (**User** or **Group**). Use User when describing the Group entity. Use Group when describing the User entity.
	linkingLDAPAttr	The multiple-value attribute name on the LDAP entity that contains the linked entity.
	linkingSearchFilter	The search filter to be applied to retrieve the list of linked objects, for example: (&(objectClass=group)(memberOf=%attr%)) The search filter may contain the string % attr % that acts as a substitution string and is replaced at run time by the value of the attribute named "attr" of the imported entity. If the LDAP entry you are describing is a Group and the string is %name%, it is replaced by the value of the "name" attribute of the group you are importing. If the LDAP entry you are describing is a User and the string is %dn%, it is replaced by the "dn" attribute of the User you are importing.
	linkingSearchScope	Sets the scope of the search. Valid values are: - **onelevel** (the value normally used) - **subtree**

Copyright © 2026, Oracle and/or its affiliates.

Sample Mapping

The following XML describes a sample mapping. The example makes the following assumptions:

- The base product attribute **displayProfileCode**is defaulted to "NORTHAM" when adding a new user.
- The LDAP Group entry contains the list of users belonging to the group in the **departmentNumber** attribute.
- The groups to which a user belongs are retrieved by applying a search filter.

```
<LDAPEntries>
  <LDAPEntry name=" User" baseDN="ou=people,dc=example,dc=com" cdxEntity="
   user" searchFilter=" (&amp;(objectClass=inetOrgPerson)(uid=%searchParm%))">
    <LDAPCDXAttrMappings>
      <LDAPCDXAttrMapping ldapAttr="uid" cdxName=" user" />
      <LDAPCDXAttrMapping ldapAttr="cn" cdxName="externalUserId" />
      <LDAPCDXAttrMapping cdxName="language" default=" ENG" />
      <LDAPCDXAttrMapping ldapAttr="givenName" cdxName="firstName"/>
      <LDAPCDXAttrMapping ldapAttr="sn" cdxName= "lastName"/>
      <LDAPCDXAttrMapping cdxName="displayProfileCode" default="NORTHAM" />
      <LDAPCDXAttrMapping cdxName="toDoEntriesAge1" default="30" />
      <LDAPCDXAttrMapping cdxName="toDoEntriesAge2" default="90" />
      <LDAPCDXAttrMapping cdxName="userEnable" default="ENBL"/>
    </LDAPCDXAttrMappings>
    <LDAPEntryLinks>
      <LDAPEntryLink linkedToLDAPEntity="Group"
linkingLDAPAttr="departmentNumber" />
    </LDAPEntryLinks>
 </LDAPEntry>
  <LDAPEntry name="Group" baseDN="ou=people,dc=example,dc=com" cdxEntity="
   Group" searchFilter=" (&amp;(objectClass=organizationalUnit)
(ou=%searchParm%))">
    <LDAPCDXAttrMappings>
      <LDAPCDXAttrMapping ldapAttr="name" cdxName="Group" />
      <LDAPCDXAttrMapping ldapAttr="description" cdxName=" Description"
default="Unknown" />
    </LDAPCDXAttrMappings>
    <LDAPEntryLinks>
      <LDAPEntryLink linkedToLDAPEntity="User" linkingSearchFilter="
      (&amp;(objectClass=inetOrgPerson)
(departmentNumber=%distinguishedName%))"
     linkingSearchScope="onelevel" />
    </LDAPEntryLinks>
  </LDAPEntry>
</LDAPEntries>
```

Oracle Identity Manager Integration

The *Oracle Identity Manager* product allows a site to centralize their user definitions and password rules to manage and deploy across the enterprise set of products. When an employee joins an organization, changes their name or departs an organization their security

Copyright © 2026, Oracle and/or its affiliates.

presence across an enterprise must be appropriately managed. Oracle Identity Manager allows for users to be provision and managed in a central location.

An integration is provided to allow the ability to create, maintain and remove users in the identity management product and sync those changes to the users defined in the application. The following sections provide additional details about the integration with respect to configuration steps required in an Oracle Utilities Application Framework based product. For more information about the configuration required in the identity management product, refer to the *Identity Management Suite Integration* technical reference paper.

In order to use this functionality, feature configuration options for the **External Messages** feature type must be configured.

- Set option type **Support SPML Deployment in IWS** to **true**.
- Set option type **Default SPML service security policy** to an appropriate value per your implementation rules.

Template User Functionality

The user object in this product captures configuration used to control access but also preferences. The identify management product allows for extending the configuration to capture user configuration that is specific to this product. However, it does not support providing searches or dropdowns to select valid values. For example, to define the user's Home Page requires the reference to a navigation option. To set up your business process such that the home page is configured when defining the user in the identity management product dictates that the security user types in the correct navigation option reference.

On the other hand, to define a minimal amount of user information in the identity management product may result in a two step process for defining users: first define them in the identity management product with the basic authentication details and setting system defaults for some important fields, then after submitting the new user to be added to this product, navigate to the user page in this product and fill in all the configuration that is specific to this product.

The product provides support for defining a template that can facilitate the definition of users and reduce some of the challenges listed above. Refer to Template User for information about defining template users along with other details around this functionality.

Once you have template users defined, you should look at extending the configuration in the identity manager product.

- Map the information that is unique to a user and in addition, define a field for the template user. For example, you may choose to only capture the Name (first and last), Email address and User IDs for the user along with its Template User (which is mapped to a user characteristic). Additional fields may be included for capture in the identity management product when defining new users as per an implementation's business needs. For example, if the organization covers multiple time zones, perhaps it is easier to define the user's time zone when defining the user in the identity management product.
- When the new user is uploaded to the system, the algorithm used by the system to create the user looks for the existence of a template user sent as a characteristic of type `F1-TMUSR`.
- Once the new user is created, its configuration can now be adjusted, if applicable.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> There is configuration needed in Oracle Identity Management to capture the template user and any other information that the implementation has chosen to define in the identity management product when provisioning a new user. Refer to the *Identity Management Suite Integration* technical reference paper for more information.

Provision Users with SCIM Open Standard

The product provides a REST web service `F1-SCIMUser` (SCIM User Provisioning) that supports adding, changing or viewing details of a user where the API follows the SCIM standard.

The following points highlight some of the mapping between the SCIM API and the application's user record.

- The user record in the application supports only first name and last name. The SCIM standard supports additional detail such as middle name, suffix and prefix. These elements are ignored when adding or updating a user.
- The SCIM standard supports a collection of email addresses. The application only supports one email address. As such, only the first email address is used when adding or updating a user.
- The integration relies on the use of a Template User so that the user is created with appropriate application configuration associated with each user. The system copies application settings from the template user to the new user being provisioned. If the external system is able to support referencing a template user such that it can be provided with the payload, the integration expects the user type element in the API to reference the template user. If it's not possible to provide a template user ID in the user type element, you should define a default template user using an algorithm parameter. Refer to Template User for more information.

Refer to the web service definition for more information about the supported integration.

Template User Functionality

The user object in this product captures configuration used to control access but also preferences. The identify management product allows for extending the configuration to capture user configuration that is specific to this product. However, it does not support providing searches or dropdowns to select valid values. For example, to define the user's Home Page requires the reference to a navigation option. To set up your business process such that the home page is configured when defining the user in the identity management product dictates that the security user types in the correct navigation option reference.

On the other hand, to define a minimal amount of user information in the identity management product may result in a two step process for defining users: first define them in the identity management product with the basic authentication details and setting system defaults for some important fields, then after submitting the new user to be added to this product, navigate to the user page in this product and fill in all the configuration that is specific to this product.

The product provides support for defining a template that can facilitate the definition of users and reduce some of the challenges listed above. Refer to Template User for information about defining template users along with other details around this functionality.

Once you have template users defined, you should look at extending the configuration in the identity manager product.

Copyright © 2026, Oracle and/or its affiliates.

- Map the information that is unique to a user and in addition, define a field for the template user. For example, you may choose to only capture the Name (first and last), Email address and User IDs for the user along with its Template User (which is mapped to a user characteristic). Additional fields may be included for capture in the identity management product when defining new users as per an implementation's business needs. For example, if the organization covers multiple time zones, perhaps it is easier to define the user's time zone when defining the user in the identity management product.
- When the new user is uploaded to the system, the algorithm used by the system to create the user looks for the existence of a template user sent as a characteristic of type `F1-TMUSR`.
- Once the new user is created, its configuration can now be adjusted, if applicable.

> **Note**
>
> There is configuration needed in Oracle Identity Management to capture the template user and any other information that the implementation has chosen to define in the identity management product when provisioning a new user. Refer to the *Identity Management Suite Integration* technical reference paper for more information.

User Redirection

The product allows an implementation to support the ability for a customer to authorize a customer service representative (CSR) to view the customer's account information in a customer facing portal (Digital Self Service), to see exactly what the end customer is seeing/ experiencing.

When the request for the CSR to view the customer information is received, the following steps occur:

- The security server for the customer facing portal is contacted to obtain a temporary grant code.
- The redirection response returns the grant code and the URL of the target customer facing portal.
- The browser redirects to the user to this URL with the grant code, which is further authenticated and automatically exchanged for an authorization token, allowing the CSR to log in.

Standard security measures with private/public key encryption using JSON Web Tokens (JWT) ensures that only the implementation's enterprise application is allowed to communicate with the implementation's customer facing portal server. The system also provides a plug-in spot to allow for additional information, such as the customer's name and email address, to be included in the call to the security server.

This feature works in conjunction with functionality provided by the enterprise customer application. Refer to the documentation for that product for information about how this feature is supported.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The product has been designed to potentially support other types of redirection besides the mentioned use case by adding another entry in the Redirection Target lookup (F1_REDIRECTION_TARGET_FLG). However, the internally provided services work specifically for the Digital Self Service use case. There is no guarantee that they would work for a different redirection use case. Please contact product support if your implementation would like to implement a different redirection scenario.

Redirection Security Context Plug-in

The product supports a plug-in spot to allow for the system to populate additional keys and values to add to the JWT (JSON Web Token) when calling the external security service.

In addition, the product provides an Extendable Lookup BO **Redirection Security Context (F1-RedirectionSecurityContext)** to define the key mnemonic values. For an enterprise product or implementation to populate additional keys, use the extendable lookup to define the key mnemonics. Then provide an algorithm that uses the available information about the customer and populates the security context keys and values by determining the related data in the system.

Master Configuration

The master configuration **Redirection Targets Configuration (F1-RedirectionTargetsConfig)** defines the information needed to contact the security server to obtain a grant code. It also defines the redirection URL.

Build an Anchor Hyperlink

The service script **Return a Hyperlink to the Redirect Servlet for a given Target (F1TargetLink)** is provided to allow for a link to be placed on a UI map or in a zone to initiate the redirection request. Refer to the service script detailed description in the metadata for more information.

Data Synchronization

Your implementation may need to communicate certain data to external systems. This may be part of a data warehousing requirement or an integration effort. The synchronization process has two main parts. First, the change to the data must be detected and captured. Once that is accomplished, the next step is to manage the communication of that change to the external systems involved. The changes must be captured in chronological order so as to avoid systems going out of sync.

- Understanding Data Synchronization
- Maintaining Sync Requests

Understanding Data Synchronization

The product supports various options for managing an ongoing synchronization of an entity to an external system. While each integration involves unique business rules, in many cases the process is quite similar and involves the following overall steps:

- When an entity is added, changed or deleted logic determines whether the change is eligible for synchronization and if so a sync request record is initiated to manage the

Copyright © 2026, Oracle and/or its affiliates.

synchronization process. The base product uses the **Audit** plug-in spot on the maintenance object to allow for such logic to be performed when a change is detected to a record for that MO. The framework calls the algorithm defined on this plug-in spot in the event a change to the MO has been detected. Refer to the description of the plug-in spot on Maintenance Object - Algorithms for more information about when this plug-in is called.

- The processing of the sync request is deferred to batch so that the communication with the external system does not impact the updating transaction. It also ensures that the data being shared with the external system is fully committed to the database.

> **Warning**
>
> Do not remove the deferred batch contol from business object status definitions of such synchronization requests. These flows must remain deferred as designed for data integrity reasons.

- When processed, sync request logic manages the following aspects of the process:
  - Determines whether the synchronization is still needed and if not discards the request.
  - Prepares the outbound message with the entity's synchronization details and sends it to the external system.
  - Manages errors and retry attempts as needed.

The following sections describe the various data synchronization options supported by the product.

- Accumulative Data Synchronization
- Generic Data Synchronization

Accumulative Data Synchronization

The product supports a data synchronization method by which several changes made to an entity may be accumulated and sent together as a single message to the external system.

The following points describe the functionality at a high level:

- When an entity is added, changed or deleted, a snapshot of the entity’s data from before the change is captured on one or more sync requests, one for each external system the data needs to be interfaced to. The before-update snapshot is used as a base line for detecting all changes made since then at the time of sending the message to the external system.
- Sync requests are processed by the next monitor batch process.
- The processing of the sync request typically involves business rules that compare the before-update snapshot with the current content of the entity to detect all change made. If changes were made, logic may decide to proceed with the sending the synchronization message otherwise the request is discarded. These rules are also responsible for handling any errors encountered while communicating with the external system and manage retry attempts as needed.

This functionality is supported by logic supplied in the base business object Sync Request (**F1-SyncRequest**). Note that each edge application delivers an appropriate child business object for this BO for each specific synch scenario supported in that product. Some of the functionality below is accomplished using configuration on the parent BO delivered by the framework while other functionality may be delivered by the child BO. In addition, there may be more complex use cases supported by your specific product integration. Refer to your specific application’s

Copyright © 2026, Oracle and/or its affiliates.

library of Sync Request business object along with the documentation related to your specific product integration for more information.

The following sections describe key aspects of this data synchronization approach.

Capturing the Change

The base product provides a change data capture (MO Audit) algorithm `F1-GCHG-CDCP` that may be used by maintenance objects that need to be synchronized by this method. This algorithm creates a Sync Request record for the changed record, capturing the MO code and the primary key, if it doesn't find an existing sync request for the same record (and the same business object) in the initial state. The sync request business object used is the one defined in the **Sync Request BO** option on the MO for the record that was changed. If more than one such option exists then multiple sync requests are created.

Your specific product may also introduce additional Audit algorithms to cater for more sophisticated examples.

When creating the sync request record, typically the Sync Request BO will have a pre-processing plug-in that captures a snapshot of the record's data prior to its change. This will be used in subsequent steps to verify that the external system needs to be notified of the change.

Confirming that a Sync is Needed

Once a sync request is captured, there are several steps performed prior to any information being sent to the external system.

> **Note**
>
> This section only highlights key steps. Please refer to the business object configuration, its lifecycle and algorithms for a thorough picture of the full functionality..

- When a Sync Request record is created, its initial state (**Pending**) is configured to be processed by a batch monitor. That way, records are added to the sync request table throughout the day but all are processed together. The MO audit algorithm ensures that a new synch request is not created if a Pending record already exists for a given MO / PK combination (for the same business object). However, it is possible that a record for that MO / PK exists in a subsequent "non-final" (such as **Awaiting Acknowledgement**). This state includes a monitor algorithm to check for that condition and to skip transitioning if another record exists. This is done to ensure that the existing record is fully processed before this new record is processed.
- The next state of the lifecycle is **Determine if Sync Needed**. This step uses an algorithm to take a snapshot (called the 'final snapshot') of the data and compare it against the initial snapshot taken when the record was created. Based on the logic of the algorithm, it may decide to proceed (transition to **Send Request** or to discontinue (transition to **Discarded**.

Communicating to the External System

Once it is confirmed that the sync should occur, a message must be sent to the external system. The following points highlight the basic functionality.

- An algorithm linked to the **Send Request** state. The expectation is that this algorithm creates an outbound message that routes the information to the external system appropriately. The algorithm must determine the external system and outbound message type to use. Business objects for Sync Request support BO options to define the external system and outbound message type to use for this algorithm.

Copyright © 2026, Oracle and/or its affiliates.

- Once the outbound message is triggered, the record transitions to the **Awaiting Acknowledgement** status. This state is used to hold the sync request from further state transitions until an acknowledgement is received from the external system. Note that this step relies on implementation of a response mechanism from the external system. It is recommended to implement a response as this helps control the chronological flow of information. The product supplies the business service `F1-UpdateSyncRequest` that transitions the sync request to either the next default state (in this case the **Synchronized** state) if a positive acknowledgement is received; or the state associated with the Rejection transition condition (in this case the **Error** state) if a negative acknowledgement is received. In addition, this state may be configured with a monitor algorithm that detects that a timeout limit has been reached.
- For records that enter the **Error** state, it is recommended to configure an algorithm that creates a To Do entry to alert someone of the problem. Refer to the integration documentation for more information. The state is already preconfigured with an algorithm to complete To Dos when exiting the state.
- The final state **Synchronized** is used to mark the successful synchronizations. However, for more complicated use cases, this state may be used to trigger some additional action. Refer to the documentation for your specific product integration for more information.

Generic Data Synchronization

The product supports a generic data synchronization method by which each change to an eligible entity triggers a synchronization flow that communicates details about the entity to one or more external systems.

The following points describe the functionality at a high level:

- The integration of a specific entity (maintenance object) and an external system is enabled and configured using a **Data Synchronization Control** (F1-DataSyncControl) extendable lookup record.
- Rules supporting the integration reside in a **synchronization script** referenced on the control record.
- When an entity is added, changed or deleted, a sync request may be initiated for each integration according to business rules defined by the corresponding synchronization script.
- Sync requests are processed by the next monitor batch process.

> **Note**
>
> In a cloud installation, near real-time processing is supported for time depenent synchronizations. Refer to Deferred Requests Queue for more information.

- When the sync request is processed, data is synchronized to the external system according to business rules defined by the corresponding synchronization script.

This functionality is supported by logic supplied in the base business object Generic Data Sync Request (**F1-GenericDataSync**). This business object does not need to be extended to support specific integration rules as these rules reside in the synchronization script associated with each integration's data synchronization control record.

The following sections describe key aspects of this data synchronization approach.

Copyright © 2026, Oracle and/or its affiliates.

Data Synchronization Control

The **Data Synchronization Control** (F1-DataSyncControl) extendable lookup defines the configuration needed to support an ongoing data synchronization of an entity to an external system. The configuration defines the entity's maintenance object, the external system it is synchronized with and references a synchronization script that manages the specific synchronization rules for this integration. An integration is considered enabled when its corresponding control record is in active status.

The data synchronization script is called in the following modes:

- When the entity is added, changed or deleted, the script is called in **Check** mode to determine whether a sync request should be created for the changed entity and a specific external system.
- When the sync request is processed, the script is called in **Process** mode to prepare and send the outbound message to the external system. Alternatively the script may request to discard the request or indicate that it should be processed later. The latter may be used when there is a need to wait until certain conditions are met before sending the message to the external system.

Note that the configuration for a maintenance object and a specific external system may be overridden by another configuration defined with a higher sequence number.

Confirming that a Sync is Needed

The base product provides a change data capture (MO Audit) algorithm `F1ONGDATASYN` that may be used by maintenance objects that need to be synchronized by this method. The algorithm consults the synchronization script associated with each active integration control record configured for the entity as to whether the external system needs to be notified for this change. Once confirmed, the algorithm creates a Sync Request record for the changed record, capturing its MO code and the primary key, along with the corresponding external system.

Note that the decision as to whether to create a sync request or not is controlled by the integration specific synchronization script while the creation of the sync request is the responsibility of the audit algorithm. As such, the same sync request BO is used for all integrations, with no provision for capturing integration specific details.

It is recommended to design the synchronization script to only approve the creation of a synchronization request when it is needed. This prevents unnecessary creation of sync requests that would otherwise be discarded when eventually processed.

Managing the Synchronization Process

Once a sync request is captured, there are several steps performed prior to any information being sent to the external system.

> **Note**
>
> This section only highlights key steps. Please refer to the business object configuration, its lifecycle and algorithms for a thorough picture of the full functionality..

- When a Sync Request record is created, its initial state (**Pending**) is configured to be processed by a batch monitor. That way, the changes that triggered the request are fully committed before being communicated to an external system.

Copyright © 2026, Oracle and/or its affiliates.

- When the sync request record is processed, a monitor algorithm calls the synchronization script associated with the corresponding integration's control record to manage the synchronization process. The synchronization script may indicate either of the following:
  - The request is not yet ready to be processed, i.e. it should remain in its current status and get processed again by the next batch run.
  - The request should be canceled. This indication results in the transition of the record to the **Canceled** status.
  - The request was successfully processed. The synchronization script has prepared the outbound message and routed the information to the external system appropriately. This indication results in the transition of the record to the **Sent** status.
- If an error occurs while processing the request, a To Do Entry is raised to capture the error details and the record is set to the **Error** status. Retry processing of the sync request continues with each batch process until the request is either sent or canceled.

Communicating to the External System

Typically the communication with an external system is finalized when the message is successfully sent and successfully processed by the external system. Depending on the specific integration, the message may not be processed right away by the external system but rather queued and processed later. In this situation, the sync request may need to be further acknowledged as successfully processed (set to the **Acknowledged** status) upon an explicit response mechanism from the external system. The product supplies the web service `F1-UpdateAndTransitionSyncRequest` that may be used to process such acknowledgements.

Maintaining Sync Requests

The system provides a Sync Request portal that is used to view the in progress or completed sync request records.

The menu location of the portal depends on your specific edge product. It may be in a Data Synchronization menu or perhaps in the Batch menu. You are brought to a query portal with options for searching. The options may differ based on your specific product.

Once a sync request has been selected, you are brought to the maintenance portal to view and maintain the selected record.

An **Actions** zone may appear to display specific actions. Alternatively, the actions may be displayed directly in the display area of the **Sync Request** zone.

The **Sync Request** zone provides basic information about the sync request record.

Depending on your specific product additional zones may appear.

Operational Analytics

The framework provides several building blocks and tools that the edge applications may use to implement analytic reports directly from the operational application using Oracle Utilities Analytics Visualization product (referred to in this section as the analytics visualization product). The following sections provide more information about this functionality.

- Understanding Operational Analytics
- Calendar and Time Dimensions
- Maintaining Bucket Configurations

Copyright © 2026, Oracle and/or its affiliates.

- Defining Analytics Options
- Defining Analytics Tables
- Defining Analytics Attribute Mapping

Understanding Operational Analytics

The following sections describe the type of configuration supported in your product to integrate with the analytics visualization product. Refer to the Oracle Utilities Analytics Visualization documentation for more information.

Direct Data Access

The analytics visualization product's canvases report directly off of the operational system (with the exception of batch run analytics) thus eliminating the need of an ETL process. Selected tables and views have been designated as dimensions and facts for the purpose of generating the Start Schemas used by these canvases.

Batch Run Analytics

There is a high volume of batch run data in the operational system that changes frequently and is recorded in a complex set of relationships. The product provides snapshot tables to capture only the batch run and thread data required for analytics. Using these tables for batch subject area views both improves performance and supports extracting the data for upload to other environments.

The snapshot tables are populated in increments. The **Batch Run Analytics Snapshot Update** and the **Batch Thread Analytics Snapshot Update** processes are run on an ongoing basis; each snapshot update selects only the batch runs and threads not yet included in the snapshot. Parameters are supplied to restrict the selected data based on a range of days from the current date, to support the initial uptake of data. It is recommended that batch run data is updated before the batch thread data, as only threads with corresponding records in the batch run snapshot will be added. Refer to batch controls `F1BRANSN` and `F1BTANSN` for more information.

The **Batch Run Analytics Data Export** and **Batch Thread Analytics Data Export** processes are provided to extract the batch data in increments for upload to other environments. These processes assume that a **Data Export Control** exists for each of the snapshot maintenance objects, to capture the date of the previous extract as a reference point for selecting the next set of records. Refer to batch controls `F1BRANEX` and `F1BTANEX` for more information.

Calendar and Time Dimensions

Analytic reports rely on calendar and time dimensions to support various hierarchical grouping by date and time.

The Calendar dimension provides a level-based definition of the standard calendar and fiscal calendar in a flattened representation that is commonly used in data warehouses. This is necessary to enabling customers to group by calendar week, months, quarter and fiscal calendar period, quarter and fiscal year etc. Note that fiscal information about a specific date is optional and sourced from your edge application specific accounting calendar

In the same way, the Time dimension provides a level-based definition of each minute in a day, supporting reports that group by hours, AM/PM etc.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Calendar dimension records are generated by a batch process; one record for each date in a specified period of time. You need to run the **Generate Calendar Dimension** batch process on an ongoing basis to cover the standard and accounting calendar days needed to support your business. Refer to `F1-BICDD` batch control for more information.

> **Note**
>
> Time dimension records are also generated by a batch process; one record for each minute in a day. You need to run the **Generate Time Dimension** batch process once to generate all the records. Refer to `F1-BITMD` batch control for more information.

Bucket Configuration

The analytics visualization product provides support for defining a set of ranges, each representing a bucket for which extracted measures can be grouped and classified under the relevant bucket. The framework product provides support for viewing and defining the buckets. Refer to Bucket Configuration for more information.

Analytics Attribute Mapping

The product supports mapping of characteristics to user defined fields associated with dimensions in the analytics visualization product. Each characteristic table that is associated with a dimension table is provided with pre-generated user defined fields identified by unique column sequence numbers in the analytics visualization product. The mapping of characteristics to dimension user defined fields is maintained directly on the dimension portal. You may also review these mappings using the Analytics Attribute Mapping portal.

Logical Dimension

The Element Name of a foreign key field to a dimension represents a distinct logical version of the dimension. This allows for different logical dimensions to be based on the same dimension table. For example, a "Location" dimension may be referenced as a "Main Location" on one fact and also as an "Alternate Location" on the same or other fact. While both elements refer to the same dimension as foreign keys, each represent a distinct logical dimension via their element names. In the same way, while each date field is implicitly considered as a foreign key to the Calendar dimension, each distinct element name establishes a unique logical dimension. For example, a "Start Date" element name and an "End Date" element name are distinct logical dimensions of the underlying Calendar dimension.

Some logical dimensions may represent an aggregated or summarized form of another logical dimension. They are also known as "shrunken" dimensions. For example, the "Calendar Month Dimension" view is an aggregated dimension of the Calendar dimension. Establishing a link between a summarized dimension and its detailed dimension allows the analytics visualization tool to drill from one fact to another for performance reasons when they are indirectly linked via the same logical dimension but at different aggregation levels. For example, "Fact A" represents daily records and "Fact B" aggregates the same measures at a monthly level. Associating the "month" field in "Fact B", which is a foreign key to the "Calendar Month Dimension" view, with the logical dimension name of the "date" field in "Fact A" allows analytic queries to use "Fact B" for queries at the monthly level or above which perform better than performing them on the more granular "Fact A".

Copyright © 2026, Oracle and/or its affiliates.

Analytics Configuration

Depending on which product or products you have installed, there may be some configuration needed for the analytics visualization product. Refer to Defining Analytics Options for more information.

Calendar and Time Dimensions

This portal is used to view generated calendar and time dimension records.

Refer to Understanding Operational Analytics for an overview of operational analytic functionality.

Navigate using **Admin**, and then **Analytics**, and then **Calendar and Time Dimensions**.

The following zones may appear as part of the portal's **Main** tab page:

- **Calendar Dimension.** This zone lists most recent calendar dimension records. You may filter the list to review records generated up to a specific date.
- **Calendar Date.** This zone provides information about a selected calendar date.
- **Time Dimension.** This zone lists all time dimension records.

Maintaining Bucket Configurations

Several key performance indicators in the analytics product look at measurement values (for example: the age of an asset or the age of debt) classified into a number of pre-defined groups also known as buckets. The overall metric can then be reported by the different buckets and allow various analyses.

For example, the age of an asset can be classified into the following buckets:

- Less than 6 Months
- 6-12 Months
- One Year and Older

The age of debt, also known as arrears can be classified onto the following buckets:

- 0-30 Days
- 30-60 Days
- 60-90 Days
- 90+ Days

The definition of the buckets is extracted to the Business Intelligence data warehouse, to be used as dimensions.

Bucket Definition Considerations

Each type of bucket is defined using a bucket configuration Business Object. The bucket definition considerations and/or rules will vary based on the bucket configuration business object used. The business objects available are driven by your specific product. For a list of available bucket configurations business objects, navigate to the business object page and view the business objects for the Bucket Configuration maintenance object.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Bucket Configurations

To maintain the bucket ranges for the bucket configuration(s) applicable to your product, open **Admin**, and then **Analytics Configuration**, and then **Bucket Configuration.**

You are taken to the query portal where you can search for an existing bucket configuration. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> Your specific product may also include an Analytics Configuration portal that displays the list of existing and potential bucket configuration records, allowing you to drill into this page to view the record in detail.

The **Bucket Configuration** zone provides basic information about the bucket configuration.

For more information about the elements supported refer to the zone help or to the relevant analytics integration documentation for your product.

Defining Analytics Options

Refer to Understanding Operational Analytics for an overview of the operational analytics functionality.

Your specific product may require some system wide configuration to be defined when using operational analytics.

Navigate to the Extendable Lookup portal. Search for and select the **Analytics Options** business object.

Use the embedded help to review information about the options to define.

Defining Analytics Tables

This portal is used to search for Analytics Table definitions. An Analytics Table record exists for each table or view in the operational system that represent dimension and fact tables in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a record has been selected you are brought to its maintenance portal to view and maintain the selected record.

- Defining Analytics Dimensions
- Defining Analytics Facts

Defining Analytics Dimensions

Copyright © 2026, Oracle and/or its affiliates.

This portal is used to maintain metadata for Analytics Dimension tables. A dimension record exists for each table in the operational system that represents a dimension table in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a dimension record has been selected you are brought to this portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Analytics Table.** This zone provides information about the selected dimension record.
- **Fields.** This zone lists all the fields that define the dimension table. Fields may be selected from the dimension's underlying operation table using the **Excluded Fields** zone or added as computed fields.
- **Dimensions.** This zone lists dimension tables that this dimension references via foreign key constraints.
- **Referenced By.** This zone lists dimension and fact tables that reference this dimension via foreign key constraints.
- **Characteristic Mapping.** This zone manages characteristic mapping extensions for the dimension. Refer to analytics attribute mapping for more information.
- **Excluded Fields.** This zone lists fields from the dimension's operational table that are not included in the dimension table.

> **Note**
>
> The ability to introduce custom analytics dimensions is currently not supported.

Defining Analytics Facts

This portal is used to maintain metadata for Analytics Fact tables. A fact record exists for each table or view in the operational system that represents a dimension table in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a fact record has been selected you are brought to this portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Analytics Table.** This zone provides information about the selected fact record.
- **Fields.** This zone lists all the fields that define the fact table. Fields may be selected from the fact's underlying operation table or view using the **Excluded Fields** zone or added as computed fields.
- **Dimensions.** This zone lists dimension tables that this fact references via foreign key constraints.

Copyright © 2026, Oracle and/or its affiliates.

- **Excluded Fields.** This zone lists fields from the fact's operational table or view that are not included in the fact table.

> **Note**
>
> The ability to introduce custom analytics facts is currently not supported.

Defining Analytics Attribute Mapping

This portal maps extensions to user defined fields provided for tables associated with an analytics dimension table that capture well defined extension data. A common example of this type of data are characteritics.

These mappings may be directly maintained on the operational Analytics Dimension portal. Refer to Understanding Operational Analytics for more information.

If your system is integrated with the Oracle Utilities Analytics product refer to Understanding Analytics Integration for more information about ETL based characteristic mapping.

You can access the portal from the **Admin Menu**, and then **Analytics Configuration**, and then **Analytics Attribute Mapping.** You are brought to a query portal with options for searching for a specific mapping record. Once a record has been selected you are brought to the this portal to review the information for the selected record.

The **Analytics Attribute Mapping** zone on the portal's **Main** tab page provides information about the mapping record.

> **Note**
>
> For characteristic mapping, only **Adhoc** and `Pre-defined` characteristic types are supported.

> **Note**
>
> For the product provides support for identifier mapping for operational analytics in this release but only the metadata definition is supported. The support for these types of attributes in the operational analytics product is in a future release.

Analytics Integration

The framework provides several building blocks and tools that the edge applications may use to implement integration with the Oracle Utilities Analytics product (referred to in this section as the analytics product). The following sections provide more information about this functionality.

- Understanding Analytics Integration

Understanding Analytics Integration

The following sections describe the type of configuration supported in your product to integrate with the analytics product. Refer to the Oracle Utilities Analytics documentation for more information.

Copyright © 2026, Oracle and/or its affiliates.

Master Configuration

Edge applications that include an integration to the analytics product typically include a master configuration record that captures information needed for the extract, load and transformation step, such as extract parameters. These records are provided by the specific edge products and my be viewed and maintained on the master configuration portal.

Note that your specific edge application may deliver an **Analytics Configuration** portal that displays the information from the master configuration record along with other analytics related configuration.

Bucket Configuration

The analytics product provides support for defining a set of ranges, each representing a bucket for which extracted measures can be grouped and classified under the relevant bucket. The framework product provides support for viewing and defining the buckets. Refer to Bucket Configuration for more information.

ETL Based Characteristic Mapping

The product provides objects to allow mapping configuration of characteristics in the product to user defined fields on dimensions in the analytics product. The mapping is used by the extract process to populate target user defined columns with corresponding characteristic data.

This type of mapping relies on configuration in the Allowed Target Dimensions extendable lookup. The extendable lookup is used to define each Target Table in the analytics product that has one or more user defined fields that may be populated with characteristic values. It also defines the valid characteristic entities that may act as the source for the characteristic data.

> **Note**
>
> The framework does not provide any values in this lookup, but edge products that support mapping provide values in this lookup. Please refer to your specific product's integration chapter and refer to the Oracle Utilities Analytics documentation for more information.

Refer to Defining Analytics Attribute Mapping for more information.

Change Data Capture Using Sync Request

Depending on the specific edge application and version you are using, there may be components of the integration that use Sync Request for the change data capture step. If that functionality applies to your implementation, the following points highlight how to get more information:

- Refer to the administration guide for Oracle Utilities Analytics to confirm if your product integration is using Sync Request for any change data capture functionality.
- Review the Sync Request Business Objects provided by your product for analytics integration.
- Refer to Data Synchronization for a high level understanding of the process.

Copyright © 2026, Oracle and/or its affiliates.

Business Flags

It is possible that information detected in one product may be useful or even critical to share with another product. The framework provides functionality for receiving information from an external system that acts as a type of flag or alert that may need investigation. This allows any system to store detected business flags in a common way and share that information with one or more other systems.

- Understanding Business Flags
- Setting Up Business Flag Configuration
- Maintaining Business Flags

Understanding Business Flags

The following is an example of a use case for business flags. Imagine that DataRaker highlights potential theft of service at a certain location. That product may initiate a business flag alert to various products owned by the implementation with a recognized "standard name" for the business flag, such as "TAMPER".

- If Oracle Utilities Meter Data Management receives this business flag, it may initiate a service investigation monitor.
- If Oracle Utilities Mobile Workforce Management receives this business flag, it may initiate a service investigation activity.
- If Oracle Utilities Customer Care and Billing receives this business flag, it may initiate a hold on billing for that location.

Note that the framework product supplies basic functionality to support logic that is common to all edge applications that implement business flag functionality. However it is the individual edge applications that supply more specific functionality (business objects and algorithms) for specific use cases, if applicable.

The following sections highlight functionality supported for business flags in the framework. Refer to the edge application product documentation for more details for supported use cases.

- Standard Name
- Business Flag Type Defines Behavior for a Standard Name
- Business Flag Type Algorithms
- Objects Linked to a Business Flag
- Impacted Business Process
- Dates
- Creating Business Flags
- Confidence

Standard Name

To ensure that Business Flags are universally understood across all edge applications and to simplify integration each Business Flag will have a Standard Name. This is a name that is used by all the products when sending information to each other. That way, if DataRaker product sends Oracle Utilities Meter Data Management a "TAMPER" business flag, it should result in

Copyright © 2026, Oracle and/or its affiliates.

the same functionality as when Utilities Customer Care and Billing sends a "TAMPER" business flag.

Standard Names are defined using an extendable lookup. In addition to common extendable lookup fields, the standard name also references a category and a class. The lookup also supports defining one or more external names, for cases where information is communicated from an external system that does not send the expected Standard Name.

The framework delivers the class values but does not deliver any standard names or category values. Refer to your specific edge application products to verify if any standard names or categories are delivered. Your implementation may configure appropriate standard names and categories based on your business rules.

Business Flag Type Defines Behavior for a Standard Name

Although the definition of the business flag standard names should be universally understood by the various integrated products that support them, each individual product defines what should occur when a business flag with a given standard name is created. This is configured using a business flag type. Only one active business flag type may exist for a given standard name. Business flags that are received from an external system will define the standard name, but will not have knowledge of the specific business flag types defined. The business flag type is determined based on the standard name.

The business flag type defines the business object to use when creating the resulting business flag record. The business object defines the lifecycle of a resulting business flag record.

Business Flag Type Algorithms

The business flag type includes support for algorithms. This allows for an implementation to define a common business object that may be used for different business flag types (if a common lifecycle is followed) but allow for different functionality to kick in depending on the business flag type.

The product supplies a plug-in spot for **Additional Processing** that may be invoked by a business flag that enters the Additional Processing state. Refer to your product's library of business objects to determine if there is an Additional Processing state that supports calling algorithms on the business flag type.

Objects Linked to a Business Flag

There are two types of links between an object in the system and a given business flag.

Affected Entity

Each business flag is associated with a single record in the system that is considered the "affected entity" or the entity that the business flag is associated with. The affected entity is defined by the specific business objects designed for the use cases supported by your edge product. For example, many utility base products may configure service point as the affected entity for its business flag use cases. Each business flag created is linked to a specific service point. Linking a specific entity to each business flag allows for algorithms to trigger functionality for that entity such as an investigation order. In addition, algorithms may be implemented in other business process areas that look for the existence of a business flag and act accordingly.

Related Objects

The business flag supports linking one or more related objects to a business flag to make it easier to trigger functionality or for impacted business processes to look for business flags. For

Copyright © 2026, Oracle and/or its affiliates.

example, when creating a business flag for a given service point, it may be useful to link all the accounts that are currently linked to the service point. Then, if an account oriented process should check for a business flag, it can look directly for a business flag linked to the account in its related object.

Impacted Business Process

The product supplies support for associating one or more impacted business processes to a business flag type. This configuration is used when functionality for that business process is impacted in some way based on the existence of a business flag of a given type. For example, maybe some process is put on hold when a certain type of business flag exists.

Note that configuring a business process on business flag type is not enough to trigger any impact on that business process when a business flag exists. There must also be some logic implemented in the business process functionality itself that knows to look for a business flag for a given record that is configured to impact the business process.

The definition of the business process is at the discretion of the edge application that supplies functionality to support this. For example, the business process could be defined as something broad such as "billing" or could be something more granular such as "billing estimation". The system supplies an extendable lookup to use for configuring the supported business processes. Refer to the values of the business process extendable lookup in your edge application or to the edge application specific Business Flag documentation for more information about supported business processes.

Dates

A business flag supports two dates: a Business Flag Date/Time and a Business Flag **End** Date/Time

- The business flag date / time is required for all business flags. For some types of business flags only one date is needed.
- For business flags that have a start and end period, the business flag date/time acts as the start date and the other field is the end date.

For a business flag that has a date range, it may be important for functionality implemented for impacted business processes. How the process treats the date will depend on its functionality.

- For some processes, the business flag is essentially expired after the end date has passed. This applies to impacted processes that are only looking at the current status of data in the system. For example, collection processing could be held if there is a business flag currently in effect (where the current date is within the date range). It would never look at historical business flags.
- For some processes where historical data may be relevant, a business flag effective during that same historical period may impact the process. For example if a business flag denotes an outage event for a given time period, perhaps estimated consumption should never be calculated for that time period.

Note that because business flags have a status, the design for the lifecycle of the business objects for the above effective dated use cases must carefully consider the states. For business flags that are considered expired after the end date passes, the BO lifecycle may be designed to transition to a final state after that date such that the record is no longer included in active processing. For business flags that continue to impact processing for a historic period, the BO lifecycle may be designed to remain in a non-final state such that it is clear that the record is still applicable.

Copyright © 2026, Oracle and/or its affiliates.

Creating Business Flags

Business flags may be created in a system for one of the following reasons:

- A message is received from an external system that initiates the creation of a business flag. In this case, logic in the external system has detected some situation that this product is being alerted about.
- Business logic in this product detects a situation that should be investigated or should act as a flag. In this scenario, there may not be any integration needed depending on the business rules.
- Business logic in this product detects a situation that another integrated product should be alerted about. In this scenario, the business flag record is used to send out information to the integrated product.
- A user manually creates a business flag based on knowledge of the affected entity. For example, a customer service representative may create a business flag as a result of contact with the customer.

Creating a Business Flag from a Web Service

The system supplies both a REST and SOAP inbound web service that may be used for an external system to initiate (or update) a business flag. The REST service is `F1-BusinessInsights`. The SOAP service is `F1-BusinessFlagSync`. In both cases, the web service invokes a "driver" service script whose ultimate responsibility is to determine the appropriate Business Flag Type based on the standard name or external standard name, and therefore the appropriate business object for the new business flag. Because different products may have different logic related to creating a business flag, the service script calls another service script linked to the maintenance object using the Business Flag Sync MO option.

The "driver" service script includes a "contextData" group, which in turn references the data area `F1-BusinessFlagContext`. This data area is to allow for an external system to provide additional information for a given business flag use case. For any integration that requires additional information to be passed in from the external system, a separate data area should be used to define the additional elements; and that data area should refer to `F1-BusinessFlagContext` as its extended data area.

The framework does not supply a Business Flag Sync service script, however individual edge applications supply a service script based on the use cases it supports out of the box.

> **Note**
>
> For products that are continuing to use XAI for external messages, the product also includes an XAI inbound service for the same Business Flag Sync Driver service script. Note that the product recommendation is to discontinue use of XAI and use inbound web services instead.

Error Handling

If there is a problem in trying to create a business flag based on incoming information, the Business Flag Sync Driver service script creates a special business flag record using the Business Flag Error Business Object. This is also configured on the maintenance object as an option. The framework product supplies the business object Business Flag Error (`F1-BusinessFlagError`) for this functionality. Refer to the business object description and configuration for more information.

Copyright © 2026, Oracle and/or its affiliates.

Confidence

There may be use cases where a condition is suspected, but not confirmed. The originating system should be able to assign a "confidence" level to the business flag.

For example, Oracle Utilities Analytics Insights or other machine learning platforms will aggregate and analyze large amounts of data to identify potential insights that need investigation. These platforms can initiate a business flag with a **Suspected** confidence.

An application receiving this business flag may adjust the confidence to either **Confirmed** or **Rejected**.

- – Setting the confidence to **Confirmed** is the equivalent of a true positive.
  - – Setting the confidence to **Rejected** is the equivalent of a false positive.

Note that the application that receives a business flag is responsible for acting on the value based on business rules. Any updates to the confidence can be communicated to the other products that received the same business flag, if desired.

Additionally, creating a business flag in the application with a confidence of **Missed** provides a way to denote a missed insight. This is the equivalent of a false negative.

Because a utility implementation may have multiple applications installed that support business flags, the following guidelines are suggested for designing where the confidence flag should be updated.

- If Oracle Utilities Service Order Management is implemented, it has the responsibility of updating the confidence flag and communicating the update to other products.
- Otherwise, the assumption is that Oracle Utilities Customer Care and Billing owns field work orchestration and that it will have the responsibility for updating the confidence flag and communicating the update to other products.

No product logic is provided to enforce the above suggestions, however, the business objects supplied by the different edge applications will support the recommended implementation.

Business Flag Updates from External System

When the product that is responsible for updating the Confidence flag makes a change, it should initiate an outbound message to alert other products. On the receiving side, the same inbound web service and Business Flag Sync service script is responsible for the update. Refer to Creating Business Flags for more information.

Setting Up Business Flag Configuration

The following topics highlight the general configuration steps required to use business flag functionality. Your specific product may supply additional functionality to support specific use cases for business flags. Refer to your specific product documentation and the library of business objects supplied for Business Flag in your product for more information.

- Standard Name Category Characteristic Type
- Standard Name Lookup
- Business Process Lookup
- Integration Configuration
- Defining Business Flag Types

Copyright © 2026, Oracle and/or its affiliates.

Standard Name Category Characteristic Type

Define one or more categories for grouping your standard names into logical business groupings.

Navigate to the Characteristic Type page. Search for and select the Business Flag Category characteristic type (`F1-BUSFC`).

Define desired category values and descriptions to be used for the standard names.

Standard Name Lookup

Navigate to the Extendable Lookup portal. Search for and select the **External Insights Standard Name** business object.

Define values that are recognized in the external systems that your implementation is receiving business flag details from.

Define a **Category** for the standard name that is appropriate for your product. Note that the category does not have to be in sync with standard name definitions in external products.

Select a **Class** for the standard name that is appropriate for its usage.

Refer to the embedded help for more information about configuring this object.

Business Process Lookup

If your specific product supports configuring business processes that may be impacted by the existence of a business flag, they are defined as an extendable lookup.

Navigate to the Extendable Lookup portal. Search for and select the **Business Process** business object.

Integration Configuration

The following points highlight configuration required to support receiving business flag information from an external source:

- Define a record for each External System that the product may be receiving business flag records from. This should be a value known by the external system and provided when new business flags are sent to this product.

When this product should initiate business flag information to be sent to an external system, configure one or more Outbound Message Type records. For each one, update the External System to configure how each outbound message type is communicated to the external system.

Defining Business Flag Types

Refer to About Business Flags for an overview of business flag functionality.

To maintain the business flag types applicable to your product, open **Admin**, and then **Integration**, and then **Business Flag Type.**

This is a standard All-in-One portal.

The information captured on the business flag type depends on the business objects supported by your product. Refer to the embedded help text for more information.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Business Flags

This section describes the functionality supported for viewing and maintaining business flags.

Refer to About Business Flags for an overview of business flag functionality.

Navigate using **Main**, and then **Integration**, and then **Business Flags**. You are brought to a query portal with options for searching for business flags.

Once a business flag record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Business Flag** zone provides basic information about a business flag. Refer to the embedded help for more information.

Market Transaction Management

This section describes various entities and aspects involved in the configuration and maintenance of market transactions. The framework product supplies basic functionality to support the objects related to market transaction management. However more specific functionality (business objects and algorithms) for specific use cases are supplied by the edge applications.

- Understanding Market Transaction Management
- Configuring Market Transaction Management
- Maintaining Market Transactions

Understanding Market Transaction Management

Several objects are provided to support market transaction management.

- Market Message Type. This object is used to define the different types of market messages, both inbound and outbound and to define the appropriate market message Business Object to use when creating objects of this type.
- Inbound Market Message. This object captures the instance of a market message received from an external source.
- Outbound Market Message. This object captures the instance of a market message that will initiate a message to an external source.
- Market Process Type. This object is used to define configuration related to processes that may be initiated to support steps that should be execute to support an inbound or an outbound market message. It defines the market process Business Object to use when creating market processes of this type. It may also define one or more market process event business objects to initiate events that are generated to support steps within the market process.
- Market Process. This object captures the instance of a market process that may be created to carry out business logic related to a specific inbound or an outbound market message.
- Market Process Event. This object is used to orchestrate individual events that may occur to support steps in a given market process.

Copyright © 2026, Oracle and/or its affiliates.

- Market Configuration. This object may be used to capture configuration needed to support market transactions. Each type of configuration should use a Business Object to define what information needs to be captured. Each type of configuration may define up to five key fields that may be used to uniquely identify an individual configuration instance.

Configuring Market Transaction Management

The following sections describe market transaction management configuration options.

- Defining Market Configurations
- Defining Market Message Types
- Defining Market Process Types

Defining Market Configurations

Each market configuration is used to define one or more options for a given type of configuration (based on a delivered business object).

The product defines a generic set of portals, described here, that may be used to define any type of market configuration. However, it may be that your implementation includes more specific configuration portals for certain types of market configurations.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Configuration.**

You are brought to the **Market Configuration Query** where you need to search for the market configuration (i.e., its business object).

Once you have found the appropriate market configuration, select the value and you are brought to a standard All-in-One portal that lists the existing values for the market configuration. The standard actions for an All-in-One portal are available here.

Defining Market Message Types

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Message Type.**

This is a standard All-in-One portal.

The information captured on the market message type depends on the business objects supported by your product. Refer to the embedded help text for more information.

Defining Market Process Types

This page is used to define market process types which control behavior of the resulting market processes and the valid market process events that could be triggered as a market process progresses through its lifecycle.

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Process Type.**

Copyright © 2026, Oracle and/or its affiliates.

This is a standard All-in-One portal.

The information captured on the market process type depends on the business objects supported by your product. Refer to the embedded help text for more information.

Maintaining Market Transactions

The following sections describe various pages available to view and work on market transactions.

- Maintaining Market Processes
- Maintaining Market Process Events
- Maintaining Market Messages

Maintaining Market Processes

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Process.**You are brought to a query portal with options for searching for market processes.

Once a market process record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Market Process** zone provides basic information about a market process. Refer to the embedded help for more information.

Related Objects

The related object tab includes several zones that display various objects that may be related to the market process.

Related Market Messages and Processes

This zone displays any entries in the related object collection that are outbound market messages, inbound market messages or market processes.

Related Processes

This zone displays market processes that are linked to this market process using the explicit related process table. The market processes linked here are marked either as a parent, a child or a sibling market process.

Other Related Objects

This zone displays market process events linked to this market process. In addition, the zone displays any miscellaneous objects that are linked to the market process in the related objects collection, besides outbound market messages, inbound market messages and market processes.

Maintaining Market Process Events

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

You can view this page when drilling into a market process event from the Market Process related objects tab.

Copyright © 2026, Oracle and/or its affiliates.

Once a market process record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Market Process Event** zone provides basic information about a market process. Refer to the embedded help for more information.

Related Objects

The related object tab includes a zone that displays all entries in the market process event's related object collection.

Maintaining Market Messages

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Message.**You are brought to a query portal with options for searching for market messages. Both inbound market messages and outbound market messages may be queried here.

Once a market message record has been selected, you are brought to the appropriate maintenance portal (inbound market message or outbound market message) to view and maintain the selected record. The inbound market message portal and the outbound market message portal have similar portal tabs and zones.

The **Market Message** zone provides basic information about the market message. Refer to the embedded help for more information.

Related Objects

The related object tab includes several zones that display various objects that may be related to the market process.

Related Market Messages and Processes

This zone displays any entries in the related object collection that are outbound market messages, inbound market messages or market processes.

Other Related Objects

This zone displays any miscellaneous objects that are linked to the inbound / outbound market message in the related objects collection, besides outbound market messages, inbound market messages and market processes.

Content Migration Assistant (CMA)

This chapter describes the Content Migration Assistant (CMA), a facility to enable the export of data from one environment to another.

> ⚠ **Caution**
>
> This chapter is intended for users responsible for creating environments for testing.

- Understanding CMA
- CMA Configuration

Copyright © 2026, Oracle and/or its affiliates.

- The CMA Execution Process
- Running Batch Jobs
- CMA Reference

Understanding CMA

The Content Migration Assistant (CMA) provides implementers with a flexible, extensible facility for migrating data from one environment to another (e.g., from a development environment to a production environment). Data is exported from the source system to a file. The file can then be checked in to a version control system for reuse, or can be immediately imported into the target system and applied.

> **Note**
>
> As used in this chapter, *source* systems are those on which export-related activities are conducted and *target* systems are those on which migration updates are to occur.

The following points highlight some common use cases for using this tool:

- Seeding a newly created environment with configuration data. For example, if you are creating test environments, you can import configuration data from a development environment. In this case, you will probably set up a "copy all configuration" type of migration. Another use case is if you want to troubleshoot an issue in production, you may want to copy all production configuration to a test environment.
- Updating an existing test environment with new configuration or updates to configuration from a development environment. In this case your set of objects will be more targeted and you are doing more of an incremental migration.
- Promoting tested configuration to your production environment.
- Migrating selected master and transaction entities from one environment to another for testing purposes.

Here are key aspects of the import process that apply to all cases:

- When records exist in both the target and the source, the CMA tool will fully **replace** the object with the version from the source. Special consideration should therefore be given to records whose primary key is system generated to ensure the correct entity is being replaced.
- While typically the source version of an object is applied directly onto the target environment, CMA does provide a way to Adjust Imported Data.
- The import process **does not handle deletion** of entire entities in the target environment. Refer to Configuration Data Deletion for more information.

Configuration steps are mainly used to define the data to migrate at the source system but also involve tasks to be performed at the target environment. Note that the products provide base delivered configuration that may be used as is or used as a template for building more specific configuration for a given implementation. At a high level, configuration tasks involve the following entities:

- System wide settings are defined in the **Migration Assistant Configuration** master configuration record.

Copyright © 2026, Oracle and/or its affiliates.

- Each type of record that may be copied requires a **Migration Plan**. The migration plan is used to identify the maintenance object (MO) for the record (using a business object) and allows for instructions to specify related records that may be included in the migration.
- A **Migration Request** is used to define the data to include in a given migration export.

Refer to CMA Configuration for more information.

Once you have a migration request defined to include desired data to export, the next steps are to export from the current environment and import to the target.

- The **Export** process includes all the steps needed to select records to be exported from the source environment and create the export file. For more information, refer to Exporting a Migration.
- The file created by the export, which is a BINARY file, needs to be transferred from the export directory to the import directory. The transfer needs to be done in such a way as to preserve the file structure. Refer to Additional Migration Considerations for more information.
- The **Import** processes include all the steps needed to read an imported file, compare the data in the file to the data in the target, review the proposed changes and apply the updates. For more information, refer to Importing and Applying a Migration.

The topics in this section provide a general understanding of the Content Migration Assistant (CMA) tool.

- Types of Migrations
- Migrating Configuration Data
- Migrating Business Data
- Data with System Generated Primary Keys
- Importing Data that References a User
- Importing Environment-Specific Data
- Configuration Data Deletion
- Additional Considerations
- Versions and CMA

Types of Migrations

The Content Migration Assistant is used for two general types of migrations: wholesale and targeted. The following topics provide some additional information about these concepts.

Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or administrative data from one environment to another. For example, a wholesale migration might be used when migrating administrative data from a development or test environment to a production environment.

A wholesale migration may be comprised of one or more migration requests that in total include all the administrative data to move. Migration plans used in wholesale migrations may be designed to omit subordinate instructions related to explicit foreign keys that are identified through constraints as they are not needed, assuming that the data they are referring to will also be included in the migration.

Copyright © 2026, Oracle and/or its affiliates.

With the ability to group migration requests, the expectation is that implementations follow these guidelines:

- Multiple migration requests using the Criteria-based or Entity List migration request classes are used to group information logically together to allow for more reuse.
- A Group migration request is used for the export. This allows for one data set to export and one data set on the import side, simplifying the process. Note that depending on the amount of data, this may be a large import set to process. An implementation may find it easier to create multiple migration requests that break down the process into several steps.

You should consider that the framework product provides base migration requests and your specific edge product may provide base migration requests as well that may or may not include framework migration plans. Using the product provided migration requests is beneficial with respect to maintenance. As features are added to the product (including new administrative maintenance objects), any impact on CMA should be included in the product owned migration requests. If your implementation introduces new custom administrative maintenance objects that should be included in CMA, then custom migration plans and a custom migration request should be added. Your implementation can build a Group migration request that includes the base migration request and your custom migration requests to have a consolidated export.

> **Note**
>
> Refer to Framework Provided Migration Objects for information about migration requests provided in the framework product. Refer to your specific product's documentation for information about addition base provided migration requests.

Targeted Migrations

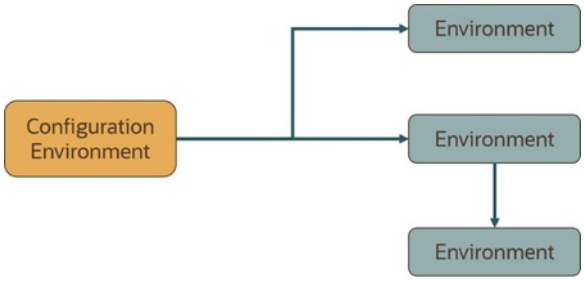
A targeted migration refers to migrating a specific subset of data from one environment to another. Migration plans used in targeted migrations are designed to be self-sufficient and include all necessary subordinate instructions needed to ensure that the exported entity can be imported without encountering referential integrity errors. With this type of migrations, any relationship explicitly excluded from the migration plan assumes the related entity already exists in the target environment. Examples of targeted migrations include:

- Migration of a new portal, its zones, and its application service.
- Migration of all outbound message types.
- Migration of selected accounts and their related master and transactional entities to a lower environment for testing purposes.

Migrating Configuration Data

A main use of CMA is to migrate configuration data, where the source environment is considered a kind of system of record and the target environment is getting updates based on changes to the source environment. Alternatively, you may be seeding a new environment. The expectation is that the target record is updated to look like the source record.

Copyright © 2026, Oracle and/or its affiliates.

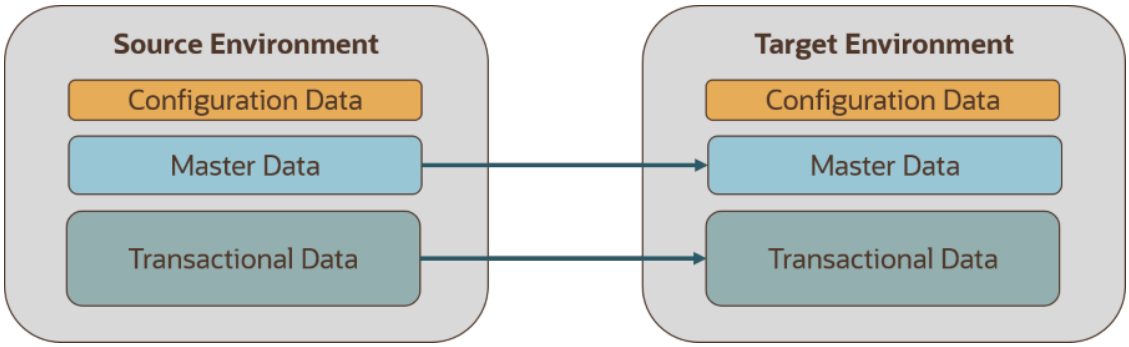


Configuration data may include a mixture of records that have an owner flag and ones that do not. The comparison step of the import process will generate appropriate insert or update SQL statements for the following data found in the export:

- Configuration data in a maintenance object with no owner flag. This is purely implementation data.
- Configuration data in a maintenance object with owner flag, where the owner is **Customer Modification**. For example, implementer-specific business objects.
- Configuration data in a maintenance object with owner flag, where the main record is owned by the product but where a child record exists with an owner of **Customer Modification**. For example, implementer-specific algorithms added to a base owned business object.
- Customizable fields in a record that is owned by the product. For example, the priority of a based owned To Do type.

Migrating Business Data

CMA may be used to perform a targeted migration of selected master entities and their related transactional data from one environment to another. For example, migrating a subset of accounts and their related data for testing purposes.



The following points highlight key differences between business data and configuration data that call for special considerations when designing your business data migrations:

- Migration of business data typically involves a much higher number of records in a data set compared with a configuration-only data set. While there is no explicit limit to the size of a data set, some reasonable limit has to be assumed.
- Having large sets of entities depend on relatively few entities in the same data set significantly slows the import process. This situation occurs when mixing configuration data, master data and high volume transaction data in a single migration. We therefore recommend designing separate migrations for these data classes. Refer to the **Reduce Dependency Between Migration Objects** section below for more information.

Copyright © 2026, Oracle and/or its affiliates.

- All business entities have a system generated key. Refer to Data with System Generated Primary Keys for more information.

The following sections expand on topics and concepts related to business data migration.

Reduce Dependency Between Migration Objects

CMA leverages the application's referential integrity and business rules to ensure entities remain compliant and valid when imported into another environment. These rules are there to prevent an entity from being imported when any of the entities it references do not exist in the target environment. When both the referencing entity and the entity it references are part of the same data set they establish a **dependency** that dictates the sequence in which these entities should be processed in order to be successfully imported. Note that the tool supports even the more complex and rare situations wherein the dependency between entities is cyclical, i.e. entity A (directly or indirectly) refers to entity B while entity B (directly or indirectly) refers back to entity A, for which sequential processing is not enough to guarantee a successful import.

The tool uses foreign key relationships between entities in the same data set to identify sets of dependent entities and group them into separate migration transaction records. Note that these foreign keys are defined in the application's data model repository and not at the database level for greater flexibility. The larger the migration transaction the more complex the process is to successfully import all its objects. So while the tool is designed to support all types of dependencies (sequential and cyclical), handling very large sets of dependent objects may incur a significant performance cost.

While configuration entities tend to be highly inter-dependent, the total number of entities that comprise the entire set of configuration objects is relatively small. Therefore, the impact on performance such dependencies have when importing a configuration-only data set is negligible. On the other hand, configuration entities tend to be referenced on many master and transaction entities as they control many aspects of the latter's business rules. As such, mixing configuration and business data in the same data set is likely to form extremely large sets of business entities that dependent on a small number of configuration entities. It is therefore strongly recommended to import or set up configuration data in the target environment prior to importing business data into it.

In a similar way, some transactional data tends to be of a much higher volume compared to the master data entities they reference. Here too, it is strongly recommend to migrate master data entities prior to their high volume related transactional data. For example, many interval measurement records (transaction data) reference the same measuring component (master data). Migrating a large volume of measurement data along with their measurement component would result in a very large migration transaction that may take long to import. Importing the measurement components prior to migrating their interval measurements will eliminate unnecessary dependencies within the measurements only data set and significantly improve the performance of the overall process.

Reasonable Data Volume

The overall volume of all business entities to migrate in a single data set should be reasonably sized. For example, importing several hundreds of accounts and their related master and transactional data is considered a reasonable size. Migrating too much data may reach physical and performance limitations of the tool.

You may use any supported migration request methods for describing the entities to export. If you opt for an Entity List migration request, consider using the Collect Entity zone to build the list of entities as you browse them on their respective portals.

Copyright © 2026, Oracle and/or its affiliates.

Single Source of Data

CMA uses an entity's prime key to determine if it is new to the target environment and therefore should be added or refers to an existing record and therefore needs to be replaced with the new version. All business entities have system generated keys, which are environment specific. It is therefore possible to have different entities in separate environments having the same prime key. When migrating entities whose keys are system generated, it is strongly recommended to migrate data to a target environment from a single source environment and avoid creating such entities in the target environment using the application. This practice ensures that prime keys for an imported entity are always in sync with their source environment. Refer to Data with System Generated Primary Keys for more information and considerations.

No Deletion

CMA does not handle deletion of entities of any kind, whether they are configuration or business entities. If a test needs to be repeated using an initial snapshot of the data, you should restore your target environment to a backup taken before test data was imported and import the latest version of the test data. If there is no need to delete previously imported entities, you may keep reloading test data from a single source environment as needed.

Bulk Import Mode

By default, the import process creates a migration object for each imported entity. This allows for granular reporting and error handling at the entity level. When importing a high volume data set of business entities, this granular management has a performance toll. In this situation, you may want to use the **Bulk Import** option on the migration data set import request. In this mode, a single migration object record is created for multiple entities of the same maintenance object. In the same way, a single migration transaction record is created for multiple logical transaction groups. Using this option reduces the migration object management effort throughout the process and thus results in better performance.

Note that each entity is still individually compared and validated, as in regular import processing, but if one entity is invalid the entire migration object is not applied hence impacting all entities grouped together with it. Bulk import mode is useful when importing a large set of data from a validated data source such that almost no errors are anticipated.

This option is only supported for master and transaction maintenance objects, i.e. it is not applicable to configuration migrations.

Insert Only Mode

By default, the comparison step of the import process needs to determine whether the imported entity is new and therefore should be added, represents a change to an existing entity and should be updated or the entity is unchanged. When importing a high volume data set of business entities that are all new to the current environment, this check is time consuming and can be avoided. In this situation, you may use the **Insert Only** option on the migration data set import request to indicate that all imported entities are assumed to be new additions to the current environment. By doing so, the import process avoids unnecessary steps to determine whether the entity should be added or updated and thus contributes to a faster import process.

This option is only supported for master and transaction maintenance objects, i.e. it is not applicable to configuration migrations.

Copyright © 2026, Oracle and/or its affiliates.

Separate Batch Processes for Managing Business Migration Objects

By default, the same import related batch processes manage configuration and business data migrations. Typically, business data migrations involve high volume of records compared to much lighter configuration data sets. Processing them together by the same batch process may slow down the performance of configuration migrations, preventing them from completing faster and more frequently. The issue mainly affects test-like environments where mixed data class migrations are more common: configuration data is imported from a lower environment and large test data is imported from a higher environment. You may adjust the base product's configuration in such environments to benefit from separating the import processes for configuration and business data.

Note that the ability to separate import processes applies to migration objects only because of their volume. Migration data set and transaction records are of low volume and as such remain managed by the same batch processes.

Designated batch processes are provided for importing migration objects containing business data but they are not used by default:

- F1-MGOPB - Migration Object Monitor (Business)
- F1-MGOAB - Migration Object Monitor (Business) - Apply

Follow these steps to use these separate designated batch processes for business data:

- Update the **Migration Object Business Data** (F1-MigrObjectBus) business object to reference the business data related batch controls on the following statuses:
  - **Pending, Error Applying, Needs Review** - Migration Object Monitor (Business) - F1-MGOPB
  - **Approved** - Migration Object Monitor (Business) - Apply - F1-MGOAB
- If your organization uses event driven job submissions for CMA batch processes then refer to Running Batch Jobs for additional configuration steps.

Data with System Generated Primary Keys

For records with system generated keys, there is an issue if the target environment already has a record with the same key but doesn't represent the same record. The tool shouldn't be just updating that target record with the view of the source in case this will break foreign key relationships that exist in the target.

The tool provides support for administrative data with system-generated primary keys. The logic relies on the maintenance object to use a method that looks at other attributes of the record (considered a "logical key") to detect whether the record being migrated already exists in the target region or not. The examples in this section are based on the Attachment maintenance object. Common attachments are considered administrative data. The attachment MO uses the file name and the creation date as the "logical key".

Imagine a common attachment for the "standard rate codes" file exists in a source region with the key 123456789. The table below highlights possible situations at the target region and actions supported in CMA.

Scenario	Target Situation	Action	Comments
1	No matching record	Record can be added with key 123456789.	

Copyright © 2026, Oracle and/or its affiliates.

Scenario	Target Situation	Action	Comments
2	Record exists with key 123456789 and logic confirms that it is also the "standard rate codes" attachment.	Record can be updated.	
3	Record exists with key 123456789, but logic detects that it is not the "standard rate codes" attachment.	Record is not updated. An error is issued.	The system cannot update this record because it's not the right attachment record.
4	The system detects that another attachment record exists for the "standard rate codes" attachment with a different ID.	Record is not updated. An error is issued.	Assumption is that the record was created directly in the target or was copied from a different source.

The use cases described in scenarios 3 and 4 above would require key mapping to keep track of the id from the source to the id in the target so that any other records from the source that reference this key as a foreign key would be updated as part of the migration. This functionality is not supported.

Scenarios 1 and 2 above are supported for maintenance objects that use the method to detect the logical key.

> **Note**
>
> If a maintenance object with a system generated key does not supply a method to detect the logical key, CMA will update an existing record with the same ID. For maintenance objects in the framework that provide this method, refer to Framework Provided Migration Configuration. For your specific edge application, refer to the CMA addendum for information about support for data with system generated keys.

The product recommends that an implementation establishes a migration strategy such that records with system generated keys are always created in the same region and always follow a standard migration path for promoting the data from this source region to other regions. Following this strategy, you would minimize or eliminate the possibility that a record for the same logical key is created in multiple places such that different IDs would be generated as described by scenario 4 above.

MOs with a Mixture of Administration and Non-Administration Data

There are some MOs that contain a mixture of master or transaction data and administrative data. The Attachment is an example of this. The product supports common attachments and owned attachments. Owned attachments are records that are specific to its owner. The owner could be master or transaction data and its attachments are therefore considered master or transaction data. Owned attachments are not candidates for migration using CMA. Common attachments on the other hand are considered administrative data and may be candidates for migration using CMA. For these use cases, an implementation may follow the suggested strategy of only creating the administrative data in one region so that IDs for common attachments are not reused. However, it is reasonable and expected that owned attachments are being created in the target region and may receive a system generated key that matches the key of a common attachment from the source region.

To try to minimize this issue, the system includes special logic to be used by any MO that may contain administrative data mixed in with master or transaction data. This special logic

Copyright © 2026, Oracle and/or its affiliates.

generates the key of an administrative record with a zero (0) in the middle of the key and ensures that the keys for master and transaction data do not include a zero in this spot. For maintenance objects in the framework that use this method, refer to Framework Provided Migration Configuration. For your specific edge application, refer to the CMA addendum for information about additional maintenance objects that may be in this category.

Importing Data that References a User

Although the tool supports copying users from one environment to another, the assumption is that it will be more common for implementations to define their valid users for each environment separately. In fact, the default CMA configuration for copying security configuration assumes that users are not being copied.

There may be times were a record is being copied from one environment to another where a user is referenced on the record but that user is not valid in the target environment. For example, if a batch control is configured as Timed with a user referenced, if this batch control is copied to another environment, what should happen if that user is not found? You may not want that user defined in the target environment. To handle this, the product provides support for a Default User that you can define in the migration assistant configuration. If this default user is configured then any time the CMA import process recognizes that a record references a User and the user does not exist in the target environment, it updates the record to instead refer to the default user.

> **Note**
>
> Please note that if you are trying to copy entities that includes valid users, you should ensure that all the users are copied to the target environment first. Otherwise, the entity will be updated to refer to the default user, which may not be your intention.

> **Note**
>
> If you copy records that reference a User within an XML field or CLOB, the CMA input tool is not able to detect that. A pre-compare algorithm is recommended if this use case exists in your implementation.

Importing Environment-Specific Data

Configuration entities that include references to environment-specific values should be imported with extreme care. Unintentionally overriding these values with data sourced from another environment may disrupt the proper function of the application. Note that these records may be initially imported from another environment but are typically adjusted manually to reflect current environment settings and should not be updated again by an import.

The sole purpose of some maintenance objects, like Message Sender, is to define environment-specific settings. As such, they are easily identifiable and included in a designated integration related migration request (F1-IntegrationConfig) that can be used carefully and as needed.

However, general purpose maintenance objects, like Extendable Lookup and Master Configuration, are designed to define various types of configurations where some may contain environment-specific information. For example, while most extendable lookup records do not contain environment-specific settings, the File Storage Configuration (F1-FileStorage) extendable lookup includes object storage definitions that are environment dependent.

Copyright © 2026, Oracle and/or its affiliates.

The following mechanism may be used to prevent an unintentional update of such records by a CMA import. It identifies the records containing environment-specific data and allows for them to be added but not updated as follows:

- Use the **Environment Specific** option to mark business objects that define environment specific information. For example, the business object for the File Storage Configuration extendable lookup is marked with this option.

> **Note**
>
> This option is available for Extendable Lookup and Master Configuration business objects but may be configured to be valid for other maintenance objects as needed.

- The product supports a generic migration plan **pre-compare algorithm** that prevents an update of an entity that its business object is marked as containing environment-specific data using the designated option.

> **Note**
>
> The algorithm is added to the Extendable Lookup and Master Configuration default migration plans and can be added to other migration plans as needed.

In addition, the following changes were made to keep such entities in environment-specific related migration requests:

- The wholesale **General System Options** migration request excludes extendable lookup and master configuration records that their business object indicates they contain environment-specific information.
- The wholesale **Framework Integration Configuration** migration request include only those extendable lookup and master configuration records that their business object indicates they contain environment-specific information.

Configuration Data Deletion

The import process does not handle deletion of entire entities in the target environment. When importing data from a source environment to a target environment, the import process for the migrated records is able to identify objects to add and objects to change. There is no mechanism for comparing the target environment to detect entities that were previously imported and no longer exist in the source system. In the same way, there is also no mechanism for indicating that records in the target environment should be deleted. The absence of those records in the import is not enough because the migration may be only importing a subset to add or update.

Note that CMA does orchestrate the deletion of child rows of an object as a result of a comparison. This is only applicable to child records that are owned by the implementation.

If data on the target system must be deleted, users must delete the records in the target accordingly.

This section describes available tools to assist with this task.

Copyright © 2026, Oracle and/or its affiliates.

Delete Entities Online

You may review and delete previously imported configuration entities using the Configuration Deletion portal.

Delete Entities In Batch

You may submit a request to delete configuration entities in batch. The process involves uploading a list of configuration entities from a CSV file onto a request record that is processed in batch by the request monitor process.

To do so create a request record using the **Delete Configuration Entities** (F1-DeleteConfigEntitiesType) request type. This option allows you to upload a list of entities to delete from a CSV file onto the request record and submit the request for batch processing when ready. When processed in batch, the request deletes all the entities in the list as a single transaction.

The file with the entities to delete should be prepared with the following header columns: maintenance object, key1, key2, key3, key4, key5. Add rows after the header row for each configuration entity to delete.

	A	B	C	D	E	F
1	Maintenar	key1	key2	key3	key4	key5
2						
3						
4						
5						

> **Note**
>
> The file should be of a reasonable size to allow the list to properly upload online. If the upload step times out use smaller files.

The deletion process abides to the application's standard referential integrity rules and therefore the order in the list should reflect data model relationship dependency.

The process stops at the first error, i.e. it does not accumulate exceptions. The user needs to review the exception, correct the issue, and resubmit the request as needed.

Additional Considerations

The following sections describe some additional considerations for CMA.

File Transfer Considerations

When moving the export file between systems, use the binary transfer option of whatever tool you use to move the file so that line-end characters are not converted from Linux-style to Windows-style or vice versa.

It is recommended to avoid using 'txt' for the export file's extension (defined in the master configuration). That file extension by default implies a non-binary file and tools that perform file transfer may treat this as a non-binary file unless explicitly stated. The recommendation is to define 'cma' as the extension. This is not a recognized file extension and most file transfer tools will transfer the file as is.

Copyright © 2026, Oracle and/or its affiliates.

Note that if the file gets converted, there are two likely outcomes - either a numeric conversion error, or a buffer under-run error may be received when attempting to import the file.

Multi-Language Environment Considerations

If your implementation uses a language other than English, it means that migrated administration objects may have multiple language rows (because English is always enabled). There are some important points with respect to multiple languages and CMA:

- As described in User Language, there are steps to follow when supporting an additional language. The steps outlined in that topic highlight that for system data, translation of the strings may be provided via a language pack provided by the product or may be the responsibility of your implementation. In either case, this effort is non-trivial and will have its own established plan. The expectation is that the translation of the system data is applied for each region at your implementation site. CMA should not be used to create a new language in a target region.
- For administrative / control data that your implementation develops as part of your project, the expectation is that descriptions for your supported language are entered in the region that is considered the source region used to promote changes to regions in the "chain". For example, control data is entered in a development region and promoted to a test with the supported language enabled in both regions.
- What if you export data from a region with more languages enabled than your target? This scenario is perhaps a case where the source region is a type of test or playpen region where the additional language is enabled for other purposes. In this case, if the language code doesn't exist at all in the target region, the import will produce an error given that the code is invalid. If the language code exists but is not enabled, this will cause the extra language rows to be inserted in the target region, but will not cause any issues. They are simply ignored.
- What if you export data from a region with fewer languages enabled than your target? In this situation, the import process will only create language rows for the languages that were copied from the source. It will not automatically create language rows in the target as part of the import. For this situation, the recommendation is to run the **New Language** batch program (`F1-LANG`) that creates any missing language entries.

Versions and CMA

This topic addresses two separate version related topics with respect to using CMA between two environments.

- Different versions of the application
- Different CMA file format versions

Different Versions of the Product

CMA was developed with the main requirement of copying data between two environments that are on the same version of the product. For example, copying configuration data from a development environment to a test environment on the same product version. It is highly recommended that you limit your use of CMA to this scenario. If it is impossible for you to ensure that both environments have the same product version, then this section highlights the possible challenges you may face when trying bring data in from a different product version.

In general, there isn't anything about how CMA functions that would limit your ability to copy data between two different versions of the product. However, there are several ways that your product may have enhanced or changed functionality for certain objects such that the data captured in a different version may not be compatible with the changes.

Copyright © 2026, Oracle and/or its affiliates.

Our products follow strict standards for upgrading clients to ensure backward compatibility. Upgrade scripts in the blueprint process know what was changed from one release to another and update the data accordingly, where applicable. When using CMA to move data from one version of the product to another, you are bypassing these upgrade scripts. It means your data is not being adjusted as expected. It's not recommended unless you are certain that the data of an object from one version is compatible in the other version.

The following points highlight this in more detail.

- For objects whose structure and rules have not changed between the two versions, there should be no issues.
- For objects that have changed between the two versions, there are scenarios that may cause additional steps.
  - If the target environment is a later version and includes a new optional column on an object that was not present in the source, the import should go through without any issues.
  - If the target environment is a later version and includes a new required column on an object that was not present in the source, a 'pre-compare' algorithm could be used to populate the new required value. Your implementation can provide this algorithm to populate the data according to your business rules.
  - If the target environment is an earlier version than the source, we cannot guarantee that there won't be specific issues encountered with specific objects.

> ⚠ **Caution**
>
> When importing data from a different environment and bypassing upgrade scripts, it may be that nothing about this product change causes an error in CMA. But because the data has not been updated as expected, it's possible that something will not work properly in the application that is not easy to detect.

Another issue you may encounter is that if system owned data is part of your import file, it's possible that the product has made changes to that data as part of an upgrade and your version of the data is different. Because CMA tries to make the target data look like the source data, when it sees a difference in product owned data, it won't proceed with importing that data. Some examples:

- Your import has included all portals, including system owned portals. In the target version, one of the zones on the portal is no longer supported by the product and is no longer linked. In your import, the zone is still linked and the system attempts to add that zone back to the portal. Because the data is system owned, it marks it as "Cannot Apply". In this case, you don't need to do anything. You can just skip this record.
- If in the previous use case you had added a custom zone to the portal and are trying to copy in that configuration, you will not be able to do it because of the system data differences. In this case you will need to manually add the custom zone to the portal in the target environment.

File Format Version

If your implementation has decided that you are going to use CMA across product versions, given all the cautions described in the previous section, this section provides information on the rare case where CMA has a change in how it writes the export file.

When creating the export file, the format of the file is specific and the CMA import step knows how to read the file. It is rare, but there are times when the product makes an enhancement to

Copyright © 2026, Oracle and/or its affiliates.

CMA that requires a change in the structure of the export file at which point the 'version' of the CMA file is incremented.

The CMA file format version has no impact to migrations that are exported and then imported into the same version of Oracle Utilities Application Framework. However, if your implementation has different environments on different versions of the product where the CMA file format is also different, then the system has functionality to cater for these differences.

- If you are exporting data from a lower file format version to a higher file format version, there is nothing special that you need to do. The export step captures the file format version and the import step reads this information and processes the file accordingly.
- If you are exporting data from a higher file format version and the product currently supports exporting to a lower file format version, at export time you are prompted for the earlier file format version. You may indicate the you want the earlier file format version when creating the migration data export.

> **Note**
>
> The product supports exporting to an earlier file format version for a reasonable amount of time, but not in perpetuity. If you don't see any option in the user interface to export to an earlier file format, it means that it is not supported in that release.

CMA Configuration

The following sections describe tasks required for CMA configuration.

- Master Configuration - Migration Assistant
- Migration Plans
- Defining a Migration Request
- Identifying Tables to Exclude From Migrations
- Configuring Custom Objects for Migration

Master Configuration - Migration Assistant

The **Migration Assistant Configuration** master configuration record defines system wide configuration used in both the source environment and the target environment.

As described in Importing Data that References a User, if you have records being imported that may reference a user does not exist (and shouldn't exist) in the target environment, define a Default User here.

In both the source environment and the target environment, the system needs to know the location of the export directory and the import directory along with the expected file suffix. For these fields, implementations may also rely on the system defaults. If no Migration Assistant Configuration record is found, the system assumes that there is an entry defined in the system's substitution variable list for `F1_CMA_FILES`. Further it defaults the values as follows:

- Export directory is the value for this variable plus `"\export"`.
- Import directory is the value of this variable plus `"\import"`.

Copyright © 2026, Oracle and/or its affiliates.

- File suffix is set to **cma**

Refer to Referencing URIs for more information about the substitution variable list.

For additional information about specific fields in the master configuration, refer to the embedded help.

> **Note**
>
> This record can be updated at any time to change details. The new configuration takes effect on all subsequent exports and imports.

Migration Plans

A migration plan defines one or more types of objects that are eligible for migration. It is essentially a set of instructions describing how the data to be exported is structured, allowing objects to be migrated together as a logical unit to ensure consistency and completeness.

The migration plan is always related to a given maintenance object, defined using its primary instruction. Multiple migration plans may exist for a given maintenance object if there are different requirements for migrating records in that MO under different circumstances. For each migration plan, you decide if additional data should be included as part of an object using subordinate instructions. For example, when copying a single algorithm type, should the system only copy that algorithm type record, or should it also copy all its algorithms and its related script (if applicable)? Refer to Types of Migrations for more information.

The following topics describe defining a migration plan as well as other topics for a migration plan.

- Defining a Migration Plan
- Understanding the BO Filtering Process
- Migration Plans for Objects with XML-Embedded Links

Defining a Migration Plan

To view or define a migration plan, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Plan.**

Use the **Migration Plan Query** portal to search for an existing migration plan. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Important! If you introduce a new migration plan, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following points provide information about defining **Instructions** for a migration plan.

The **Instruction Sequence** uniquely identifies the instruction. The recommendation is to use increments of 10 to allow insertion of other instructions in the future.

Copyright © 2026, Oracle and/or its affiliates.

Select **Primary** for the first **Instruction Type**. All migration plans must contain one and only one primary instruction. All subsequent instructions require a **Subordinate** instruction type. In this case, the **Parent Instruction Sequence** must be entered. This number, used to maintain the defined relationships in the exported data, must match an instruction sequence number at a higher level in the hierarchy.

The instruction **Description** provides a business description of the instruction.

Select a **Business Object** (BO) to define the type of object from which data will be derived.

> **Note**
>
> Though BOs are specified in each instruction, it's important to understand that each BO is used only for filtering purposes. The migrated data set comprises the complete contents of the *maintenance object* that the business object structure is defined against. For a more detailed explanation of this, see Understanding the BO Filtering Process.

> **Note**
>
> Refer to Identifying Tables to Exclude From Migrations for information about defining child tables to always exclude from a migration.

Traversal Criteria is used to define the relationship between each of the objects in a migration plan. The system provides three options to define how the child object is connected to the parent object so the system knows how to traverse from one object to another. **Traversal Criteria Type** options are **Constraint**, **SQL** and **XPath**. The following points explain each option:

- **Constraint** allows you to select a table constraint that represents a given record's relationship to another record in the system via a foreign key constraint defined in the meta-data. If **Constraint** is selected, the following additional fields are enabled:
  - **Constraint ID** is a unique identifier for the constraint. The search will show the valid table constraints for the MO of the instruction's BO and the MO of the parent instruction's BO.
  - **Constraint Owner** is used to define the owner of the constraint. This is populated automatically when selecting a constraint from the search.
- **SQL** lets you specify SQL join criteria between the parent instruction's object and the child object in the **SQL Traversal Criteria**. The syntax of the the traversal criteria is a WHERE clause (without including the word WHERE). When referring to a field on the parent instruction's object, use the syntax `#PARENT.TABLE_NAME.FIELD_NAME`. When referring to a field on the current instruction's object, use the syntax `#THIS.TABLE_NAME.FIELD_NAME`. For example, the following statement is used on a migration plan for Business Object, where the parent instruction is the BO and the subordinate instruction is used to reference the UI Map that is referred to as a BO option with the option type `"F1DU":#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_FLG = 'F1DU' AND @trim(#THIS.F1_MAP.MAP_CD) = @trim(#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_VAL).`
- The **XPath** option lets you apply syntax in an XPath expression referencing elements in the instructions' referenced business objects. This is entered in the **XPath Traversal Criteria**. For example, the display map collection statement in the SQL example noted above would be written as follows in XPath: `#this/mapCd = #parent/businessObjectOption/businessObjectOptionValue AND #parent/`

Copyright © 2026, Oracle and/or its affiliates.

`businessObjectOption/businessObjectOptionType = 'F1DU'`. This technique allows foreign key references that are mapped inside an XML column to be referenced.

> **Note**
>
> The #parent expressions may access elements that are stored in an XML column and described using mapXML and mdField. However, the #this expressions must refer to fields available in the business object using the mapField reference.

Defining **Next Migration Plan** provides the ability to indicate that in addition to copying the object defined in the instruction, any additional instructions included in that referenced migration plan will also be included in an export.

The **Algorithms** grid contains algorithms associated with each instruction. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the Sequence in which they should execute.

System Event	Optional / Required	Description
`Pre-Compare`	Optional	Algorithms of this type may be used to adjust the data after it is moved to the target system. These may only be defined on the primary instruction. Refer to Adjusting Imported Data for more information.
Import	Optional	Algorithms of this type are no longer supported.

Understanding the BO Filtering Process

Migration plan instructions require the definition of a business object to provide CMA with information about the record related to the instruction.

If the business object is the physical business object for the maintenance object, then CMA assumes that the instruction applies to all records that satisfy the traversal criteria. CMA recognizes the physical BO by comparing the BO to the value defined in the maintenance object option. If the business object defined is not the physical BO, then CMA will limit the records in the instruction to those that explicitly reference this BO or reference a child of this BO as its identifying BO value. (In other words, this BO must be in the parentage hierarchy of the records to be included in the instruction.)

> **Note**
>
> Unlike Bundling, CMA does not use the BO schema to drive what data is copied for a given record. The BO is only used as a filtering mechanism for selecting records. Refer to Identifying Tables to Exclude from Migration for information about how to ensure a child table is not included in a migration.

Copyright © 2026, Oracle and/or its affiliates.

For example, if you define a migration plan for Master Configuration and use the physical business object for the instruction (`F1-MstCfgPhysicalBO`) then all master configuration records are considered for the instruction. If instead the business object you define is Migration Assistant Configuration (`F1-MigrationAssistantConfig`) then only the record related to this business object is included in the instruction.

Migration Plans for Objects with XML-Embedded Links

When migrating objects where foreign key references are captured in the object's XML based field, subordinate instructions are needed to define the foreign key references in order for CMA to understand the relationships. This is in contrast to direct foreign keys where CMA can determine the relationships using constraints. The instructions provide two purposes. Defining instructions ensures that the related objects are included in the migration, if appropriate. If you know that the related objects are included because of other criteria, the instructions allow CMA to group related objects into transactions. This helps in the apply process at import time to ensure that related objects are grouped together. However, the apply process includes iterative steps to try to overcome dependencies like this so defining the instructions is not critical for this purpose.

The following are options for creating migration plans with XML-embedded links:

- One option is to use the specific logical (business) BO in the primary instruction to define the object you are copying. With this option, the subordinate instructions may use XPath criteria to define the related foreign key. When this approach is used, a separate Migration Plan must be created for each logical BO. (Refer to Understanding the BO Filtering Process for more information.) This option would only be used in isolated cases.
- Another option is to create a migration plan that uses the Physical BO as the primary instruction, and then include a subordinate instruction for the real logical BO, using SQL Traversal to join the object to itself by its primary key. Note that with this technique, the records that reference the logical BO will still only be included in the export file once. At this point further subordinate instructions may use XPath notation to define the foreign key data. Using the physical BO as the primary instruction ensures that all records in the MO are considered. The subordinate instructions with the logical BO and XPath notations will only apply to the records that are applicable to that BO. This option is useful for MOs that have a small number of logical business objects with disparate foreign keys.
- Another option is to use the physical BO in the primary instruction and use raw SQLs in the subordinate instruction's traversal criteria to identify the foreign keys using substring commands. A separate Subordinate Instruction is needed for each SQL corresponding to each element occurrence. Using this technique has the same advantages of the previous in that all records for the MO are included in the migration. However, this technique may be useful for maintenance objects with a larger number of business objects expected where each has one or more foreign keys. It's especially useful if many business objects reference the same foreign key. Then only one instruction is required for that foreign key. Note that a single migration plan may use this technique and the XPath technique for different elements.

A migration request may have multiple migration plans for the same maintenance object. That allows for some flexibility and long term maintainability in that the above techniques may be used in multiple migration plans. Consider the following example:

- A product provides base business objects with foreign keys defined in the XML field and provides the appropriate migration plan with instructions. An implementation extends this business object or perhaps creates their own business object for the same maintenance object and includes different additional foreign keys in the XML. Rather than duplicating the base migration plan and adding additional instructions for the additional foreign keys, the implementation can create a second migration plan for the MO with the additional foreign

Copyright © 2026, Oracle and/or its affiliates.

keys defined. A migration request should be defined to include both migration plans. In this case if the implementation has only one custom BO, they can choose to use the custom BO as the primary instruction as described above in the first option.

Defining a Migration Request

Migration Requests are used to define the data to be included in a migration. To view or define a migration request, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Request**.

Use the **Migration Request Query** portal to search for an existing migration request. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

There are three types or classes of migration request. The system provides a base business object for each along with a migration request class, which matches the business object. The subsequent sections provide more information about each class of migration request.

Note that all migration requests support defining a Category, which allows implementers to categorize the migration request.

In addition, all classes of migration request include the following zones:

- **Migration Request** This zone contains display-only information about the selected record. Please see the zone's help text for information about this zone's fields.
- **Referencing Migration Requests** This zone is only visible if the displayed migration request is included in a Group migration request. It lists each group migration request that includes it.

Other zones may appear for specific classes of migration requests. See the following sections for more information.

Criteria-based

This type of migration request defines a set of entities to be logically included together as part of a migration. For each migration instruction, selection criteria is defined to indicate which records for each MO should be included. Selection may be defined using SQL, an algorithm, an entity tag or explicit primary key values.

- For selection using SQL Statement, refer to the embedded help for examples.
- For selection using Algorithm, the algorithms that are eligible to be plugged in here are those valid for the **Migration Request - Select** system event.
- For selection using Entity Tag, reference a tag name. The entities linked to the tag at export time would be included in the export file.
- For selection using Specific Keys, the primary key (1 through 5) must be explicitly specified. Multiple rows are allowed.

Entity List

This type of migration request allows the user to choose explicit MO / prime keys. The MOs that are eligible are those that are configured with a **Default Migration Plan** option. Although the user is managing MO / PKs, the migration instructions are still defined with a migration plan. The system maps the migration instructions in a similar way to a `Criteria-based` migration requests that use a Specific Key selection type. Note however that it will create a separate migration instruction for each MO / PK combination. It does not try to group all PKs for the same MO / migration under one migration instruction.

Copyright © 2026, Oracle and/or its affiliates.

For this type of migration request, a user adds a migration request record with its description and other main information. Then, a special zone **Add Entities** is provided to find and select records based on a selected maintenance object and add to the migration request. The user is prompted to provide a reference and comments, if desired. If the category selected is one that requires a reference, then this will be validated. Alternatively, you may also use the Collect Entity zone to add specific entities as you review them on their respective maintenance portals.

When maintaining a migration request with existing entities, they are visible in the zone **Migration Request Entities** . This zone allows a user to remove the entity from the list.

Group

This type of migration request points to other migration requests. This allows you to define separate migration requests that represent logical groupings of migration plan instructions for ease of maintenance, but to combine all the separate migration requests into a single "grouped" migration request for streamlined export / import purposes.

The CMA export process will build an extract that includes the union of all the objects that qualify for the export and group them together based on their relationships.

Identifying Tables to Exclude From Migrations

Some maintenance objects that are eligible to be migrated may include child tables that should not be included in the migration. For example, if an object includes log tables, the entries in the log should reflect the actions on the object in that system, and will be different between the source system and the target system. If you have a custom Maintenance Object that includes tables you don't wish to migrate (such as a log table), use the **Non-Migrated Table** option on the MO to specify this table. All child records for this table will also be ignored during migration.

Another use case to consider is a child "many-to-many" table that connects two administrative objects and exists in the maintenance object of both tables. The child table may be in both MOs for convenience sake, but it may be that one MO is considered more of a "driver" object and the other more of a subordinate. If you are doing a migration where you want to copy a subset of objects, you may want to only copy the driver object and its children and their data but not their children. For example, a To Do Type includes a collection of valid To Do Roles and in turn the To Do Role refers to its To Do Types. If an implementation wants to copy a single To Do and include all its related information, including its To Do Roles, it does not want the migration of each To Do Role to in turn copy all its To Do Types (and their data).

> **Note**
>
> The MO option must be set in both the Source and Target systems for a given MO.

Configuring Custom Objects for Migration

During the implementation and extension of the product, new custom administrative maintenance objects may be introduced. If your implementation would like to migrate records in those maintenance objects using the Content Migration Assistant, additional steps must be performed, which are highlighted in the following sections.

Physical Business Object

As described in Understanding the BO Filtering Process, the migration plan requires a business object for its instruction. The business object is used to identify the records eligible for inclusion in the migration. Assuming your custom tables use one or more "logical" business

Copyright © 2026, Oracle and/or its affiliates.

objects for their processing, your implementation must decide if these business objects are appropriate for use by the migration plans, or if a physical BO is warranted. If so, create an appropriate physical BO.

Review MO Option Configuration

The following points highlight maintenance object (MO) configuration that should be reviewed or updated to support CMA:

- If a physical BO was created (above), link it to the MO as an option using the appropriate option type.
- Be sure that your MO defines an appropriate FK Reference and includes an Option on the MO that identifies the FK Reference. This is used by various portals and zones for CMA when showing detail about records being imported into the target region. Also be sure that this FK reference defines an Information program.
- As described in Identifying Tables to Exclude From Migrations, an MO option is used to identify child tables for an MO that should never be included in a migration. If your custom maintenance object includes a standard Log table, than the recommendation is to list that table as an excluded table. Depending on the specific design of the maintenance object, there may be other child tables to define.

Characteristic Type Configuration

The CMA import process will attempt to create a log record for any administrative object that includes a log table. If your implementation has introduced any custom administrative tables that you plan to include in a migration request and it includes a log table, you must, to ensure that the log creation is successful, add your log table as a valid characteristic entity to the characteristic type `F1-MGO` (Migration Object).

Navigate to Characteristic Type and select the characteristic type `F1-MGO`. Navigate to the **Characteristic Entity** tab and add a row to include the characteristic entity for your custom maintenance object's log table.

Standard CMA Configuration

Create one or more migration plans for the new object, depending on the type of data in the maintenance object and the types of migrations you envision:

- If you have implemented only one "logical" business object used to define the data in the MO, then a single migration plan that references the this BO (or the maybe the MO's physical BO) is appropriate.
- If you have implemented more than one "logical" business object, would the data for multiple business objects get copied together? Then perhaps a single migration plan that references the MO's physical BO is appropriate.
- Are there additional foreign keys defined using mapXML in the business object(s) for the MO? If so, then it is recommended to include sub-instructions to define the links. At this point, if multiple "logical" BOs exist, your implementation may choose to define all the additional elements in the same migration plan or choose to define separate migration plans for each logical BO.
- Your implementation may decide to define more than one migration plan for the same type of record based on the types of migrations you plan to include. For example, you may decide to include a migration plan that copies only the records in this maintenance object. You may decide to define another migration plan that copies the records in this MO along with related records in another MO (for a special type of migration). Having said that, be sure to design the migration plan with reuse in mind to minimize maintenance efforts.

Copyright © 2026, Oracle and/or its affiliates.

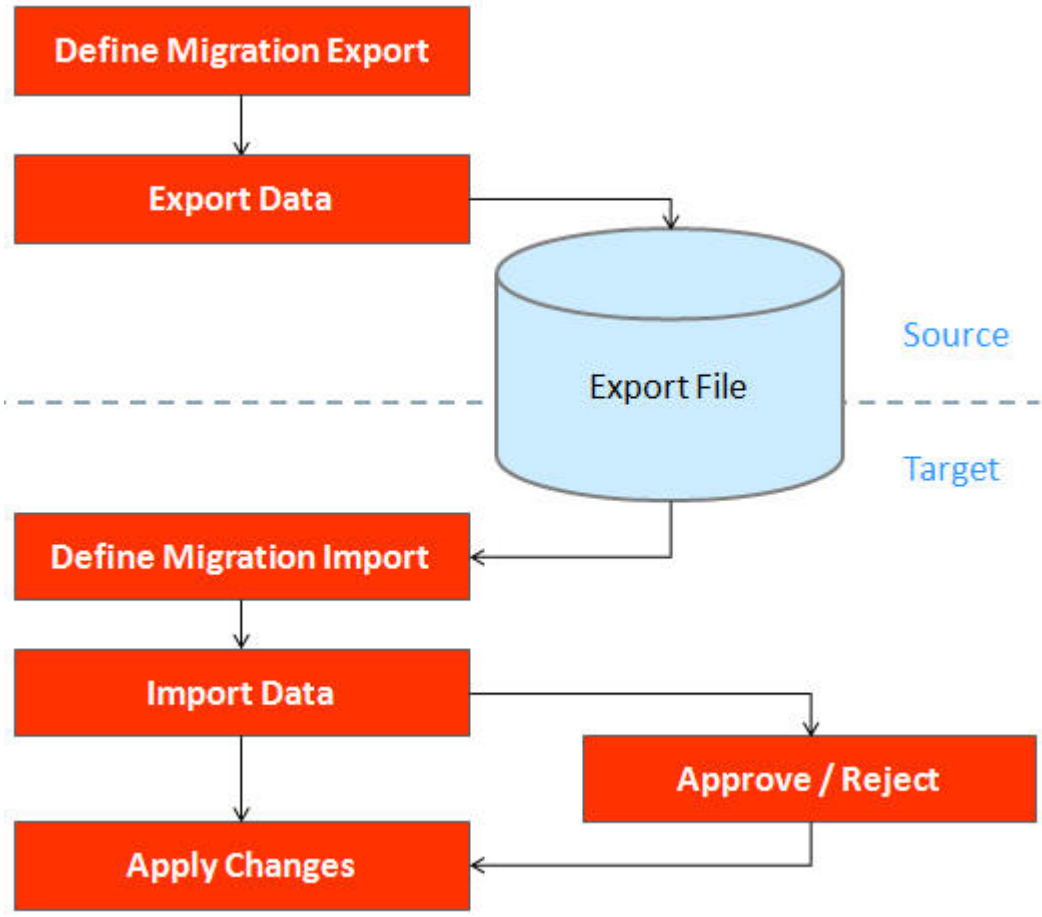
In order to support **Entity List** migration request, a default migration plan must be defined as an option on the maintenance object. This should be a single migration plan that supports all types of business objects for the MO.

If your implementation has a template migration request to use for migrations, include the new migration plan(s) as appropriate.

> ⚠ **Caution**
>
> Important! New migration plans and migration requests should follow naming conventions. Refer to System Data Naming Convention for more information.

The CMA Execution Process

The following diagram illustrates a high-level view of the Content Migration Assistant execution process. The subprocesses illustrated here are described in more detail in the following sections.

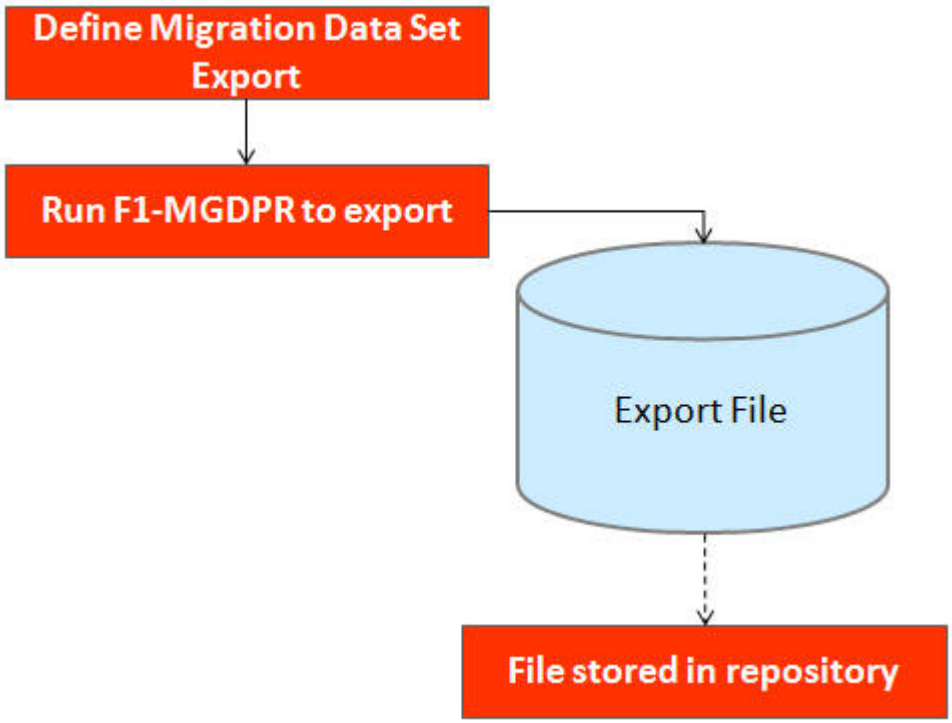


- Exporting a Migration
- Importing and Applying a Migration

Exporting a Migration

Copyright © 2026, Oracle and/or its affiliates.

The migration export process begins in the source environment by defining a **Migration Data Set Export**, which specifies a defined **Migration Request** and provides a file name and description for the export file. After the data set is defined and saved, the **Migration Data Set Export Monitor** batch job can be submitted generate the export file. The lifecycle of the Migration Data Set Export business object includes algorithms that select the appropriate records according to the migration request, determine dependencies between records to build groupings of related objects and create the export file.



The following topics provide more detail about this process.

- Migration Data Set Export
- Export Lifecycle

Migration Data Set Export

To migrate data from one region to another, define a **Migration Data Set Export**. This establishes the export file name and identifies the migration request.

To view an existing migration data set export, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Data Set Export**. Use the query criteria to locate the desired data set.

Note that you can also initiate the creation of an export data set from the Migration Request portal using the **Export** button.

The export requires the name of an existing **Migration Request**.

Enter a unique **File Name** for the export. Do not use spaces in the name, and do not enter the file extension or a path. The output location and file extension of the intended export file, which should appear in the **Export Directory** and **File Suffix** labels, are defined as described in the topic Migration Assistant Configuration.

Copyright © 2026, Oracle and/or its affiliates.

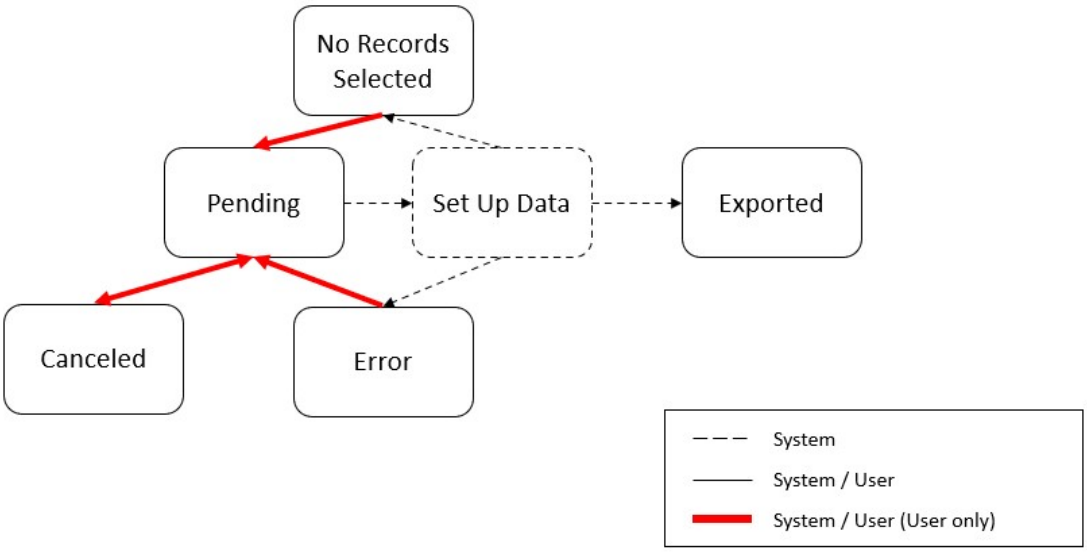
Enter an **Export Description** to provide information about the purpose of the export. Note that this field is not language enabled.

The **Source Environment Reference** is for information purposes. It should be populated with text that provides a meaningful description of the source environment. The default value is the URL of the source environment.

The **Migration File Version** is only visible if the product currently supports creating a file in an earlier file format version. If this element is visible and you are planning to import the file in a target environment that uses an older migration file format version, indicate the lower file format version that you wish the export to use.

Export Lifecycle

The following diagram describes the lifecycle of a Migration Data Set Export (data set).



The following points describe the lifecycle.

- The data set is created in **Pending** status.
- A user may choose to temporarily **Cancel** a pending data set to prevent it from being processed. The user can later return it to the Pending state when desired.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Export Monitor** (F1–MGDPR) selects pending records and transitions them to **Set up data**. Refer to Running Batch Jobs for more information.

> **Note**
>
> This batch control is delivered with the batch strategy configured as 'JOBS'. This ensures a more even distribution of work when there are multiple data sets to export.

- Set up data is a transitory state that includes the algorithm that does the work of determining the objects to include in the export and group related objects together into a transaction. If everything is successful, the export file is written to the appropriate file location and the record transitions to **Exported**. If the algorithm detects that no records are

Copyright © 2026, Oracle and/or its affiliates.

selected per the migration request instructions, the record transitions to **No Records Selected**. If an error is detected, the process stops and the record transitions to **Error**.

- If a record is in error and it is possible to correct the error, the record may be transitioned back to Pending to try again.
- If no records were selected and you want to adjust the migration request instructions to ensure that records are selected, the record may be transitioned back to Pending to try again.

When the process is marked as **Exported**, the export file can be imported into the target system.

> **Note**
>
> The export process creates a file, providing the benefits of having a standalone file. It can be stored in a version control system for audit purposes or provided to others for import purposes.

> **Caution**
>
> Under no circumstances should exported data files be edited manually. Doing so could cause data corruption when the file content is applied to the target environment.

> **Note**
>
> The export functionality is supported using the business object **Migration Data Set Export** (F1-MigrDataSetExport). The expectation is that implementations will use the delivered base business object and its logic and will have no reason to implement a custom business object for the CMA export process.

Importing and Applying a Migration

The import process is broken down into four general steps: Import, Compare, Approve, Apply. The following points provide an overview of the steps.

- **Import**. The first step covers importing the file and creating appropriate Migration Import records in the target environment to facilitate the subsequent steps.
- **Compare**. The compare step reviews each object that is in the import file and compares the object in the import against the equivalent record in the target environment. The comparison step results in noting which objects are unchanged, which are new (and the appropriate SQL to insert them) and which objects are changed (and the appropriate SQL to update them). Based on user configuration at import time, the objects that qualify for the import may be in a state that requires review or may be pre-approved.
- **Approve**. Once the comparison is complete, the user should review the results. There may be records marked for review. All of these records must be approved or rejected before the import can proceed. Users may choose to suppress individual SQL statements for a given object that is approved. When the user is satisfied with the results of the comparison and has completed the review, the import is marked to proceed to the Apply step. Optionally, a migration import may be configured to automatically apply.

Copyright © 2026, Oracle and/or its affiliates.

- **Apply**. This is the final step and is the step where the records in the target environment are added or updated. Because of potential high volumes of data and because of possible dependencies between records, this step supports two levels of attempting to apply the records. There is an apply step at the object level and an apply step at the transaction level. This will be described in more detail below.

- Import Step
- Compare Step
- Approval Step
- Apply Step
- Adjusting Data Prior to Comparing
- Import Process Summary
- Cancelling a Data Set
- Additional Note Regarding Imports
- Caching Considerations
- Maintaining Import Data

Import Step

The import process starts with verifying the import directory configured as described in the following topic Master Configuration - Migration Assistant and ensuring that the exported file is located in that directory. Then, in the target environment, a **Migration Data Set Import** record should be created. The user indicates the file name.

In addition, the user decides what the default status should be for resulting objects.

- The **Default Status for Add** sets the default status for objects that are determined to be ***new*** during the import comparison process. The default is to automatically set new objects to **Approved** status. Other options are to set any new objects to **Rejected** or **Needs Review** status.
- The **Default Status for Change** sets the default status for objects that are determined to be ***changed*** during the import comparison process. As with new objects, the default for changed objects is **Approved**, with **Rejected** or **Needs Review**options available.

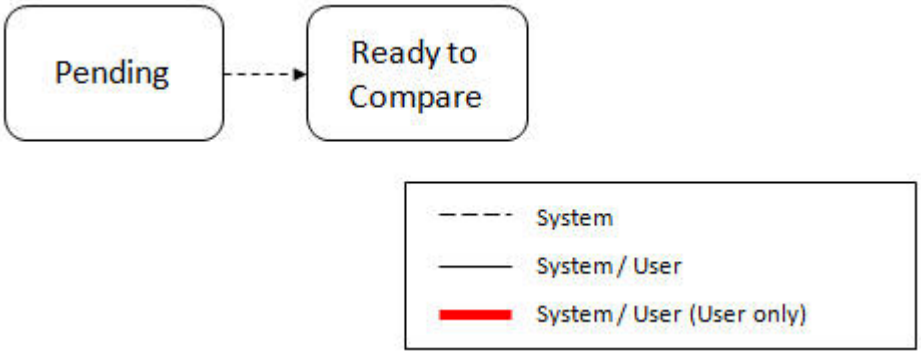
The user may also configure the **Automatically Apply** flag to **Yes**. This allows for use cases where the migration is repetitive and has been tested and the user feels that there is no need for manual approval. Note that when configuring this setting, neither of the Default Status values may be set to **Needs Review** and at least one must be set to **Approved**.

The file to import contains a list of all the objects included in the export. Any objects that the export step determined to be related have been grouped into “transactions”. Once the Migration Data Set Import is created, the next step is for the system to read in the file and create Migration Transactions and Migration Objects.

By default, the import step creates a Migration Object for each entity in the file. When importing a high volume data set of business entities, the user may request to import entities in bulk mode for performance reasons. In this mode, a single Migration Object record may contain multiple entities from the file. If at least one of the entities included in the migration object record is new and the remaining ones are either new or unchanged then the **Default Status for Add** setting applies to it. If at least one included entity is determined to be changed then the **Default Status for Change** applies to the record. Refer to bulk import mode for more information.

Copyright © 2026, Oracle and/or its affiliates.

The following is a portion of the Migration Data Set Import lifecycle as it pertains to the import step.



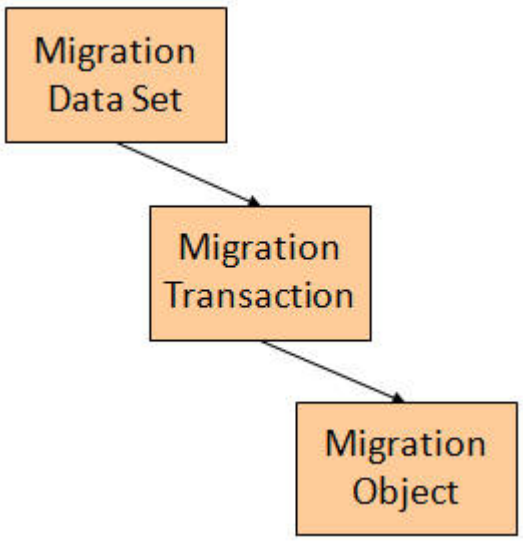
The following points describe the lifecycle.

- The data set is created in **Pending** status.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Import Monitor** (F1–MGDIM) selects pending records and transitions them to **Ready to Compare**. Refer to Import Process Summary and Running Batch Jobs for more information.

  The Ready to Compare state has an algorithm that is responsible for reading the related import file and creating the migration transactions and migration objects. The data set remains in this state until the comparison step is complete.

> **Note**
>
> A user may choose to **Cancel** a data set. Refer to Cancelling a Data Set for more information.

The following diagram highlights the relationships of the resulting migration import records.



The migration transaction and migration object each have their own lifecycle that will help manage the subsequent compare and apply steps. At the end of the import step, the status values of the three types of records are as follows:

- Migration Data Set Import is in the **Ready to Compare** state.

Copyright © 2026, Oracle and/or its affiliates.

- Migration Transaction is in the **Pending** state.
- Migration Object is in the **Pending** state.

> **Note**
>
> The import functionality is supported using business objects supplied by the base product. The expectation is that implementations will use the delivered base business objects and their logic and will have no reason to implement a custom business objects for the CMA import process. The base business objects are **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).

Compare Step

The import step results in the creation of one or more migration objects. Typically, one migration record is created for each record selected in the export based on the export's migration request and its configuration. Related objects are grouped together in migration transactions.

When importing a high volume data set of business entities, the user may request to import entities in bulk mode for performance reasons. In this mode, a single Migration Object record may contain multiple entities from the file. Refer to bulk import mode for more information.

The next step in the import process is the Comparison step. In this step, the data captured by the import file for each object is compared to the view of that object in the target environment.

To cater for a possible large volume of objects, the comparison is done via a batch monitor. To aide in performance of the process, the monitor is performed on the migration objects so that it can be run multi-threaded. Once the objects are finished with the comparison, the migration transactions and the migration data set should be updated with an appropriate overall status before continuing to the next step. As a result, the comparison actually requires three steps: Migration Object Comparison, Migration Transaction Status Update and Migration Data Set Export Status Update. The steps are explained in detail in the following sections.

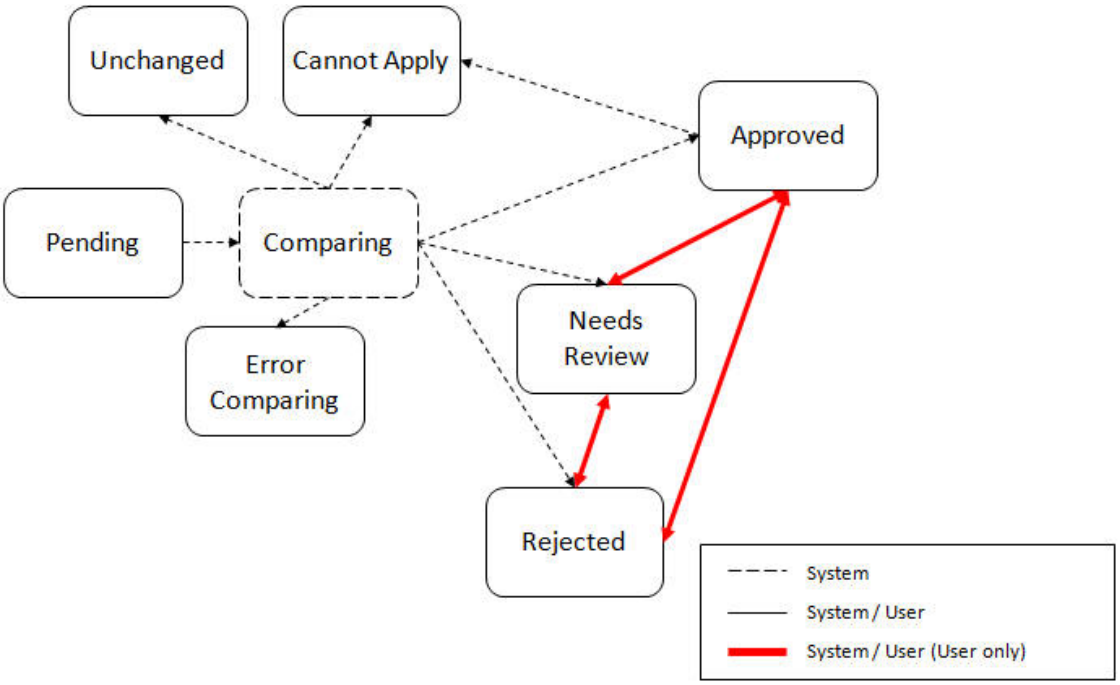
> **Note**
>
> Refer to Running Batch Jobs for more information about streamlining the various steps in the process.

Migration Object Compare

This is the main step of the comparison. The **Migration Object Monitor** (F1–MGOPR) selects pending migration object records and transitions them to **Comparing**. This is a transitory state that includes the algorithm that does the work of comparing. There are various possible outcomes that could occur based on the logic in the algorithm. The following diagram illustrates a portion of the migration object lifecycle that pertains to comparison.

> **Note**
>
> When using separate batch processes for business data the **Migration Object Monitor (Business)** process (F1-MGOPB) works in the same way to compare business migration objects.

Copyright © 2026, Oracle and/or its affiliates.



The following points describe the lifecycle.

- When **Pending** records are selected by the monitor batch job, it transitions to **Comparing**. If the migration object refers to one or more pre-compare algorithms, they are executed to adjust the data prior to comparison. Then algorithm will determine the appropriate next state by comparing the source data to the target data.
- Entity related notes:
  - If the record in the migration object is found in the target environment and the data is exactly the same, the record transitions to **Unchanged** (with the object action value also set to **Unchanged**).
  - If the record in the migration object is found in the target environment and the data is different, the algorithm sets the object action value to **Change** and generates the appropriate SQL to be used later in the Apply step to update the record. It then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Change setting captured on the Data Set.
  - If the record in the migration object is not found in the target environment, the algorithm sets the object action value to **Add** and generates the appropriate SQL to be used later in the Apply step to insert the record. It then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Add setting captured on the Data Set.
  - When the import is performed in bulk mode each included entity is compared separately but the overall status of the record is set as follows:
    - If all included entities are unchanged then the migration object record transitions to **Unchanged** (with the object action value also set to **Unchanged**).
    - If all included entities are either unchanged or changed then the object action value is set to **Change**), corresponding SQLs are generated for the changed entities only and the migration object transitions based on the Default Status For Change setting captured on the Data Set.
    - If all included entities are either unchanged or new then the object action value is set to **Add**), corresponding SQLs are generated for the new entities only and the

Copyright © 2026, Oracle and/or its affiliates.

migration object transitions based on the Default Status For Add setting captured on the Data Set.

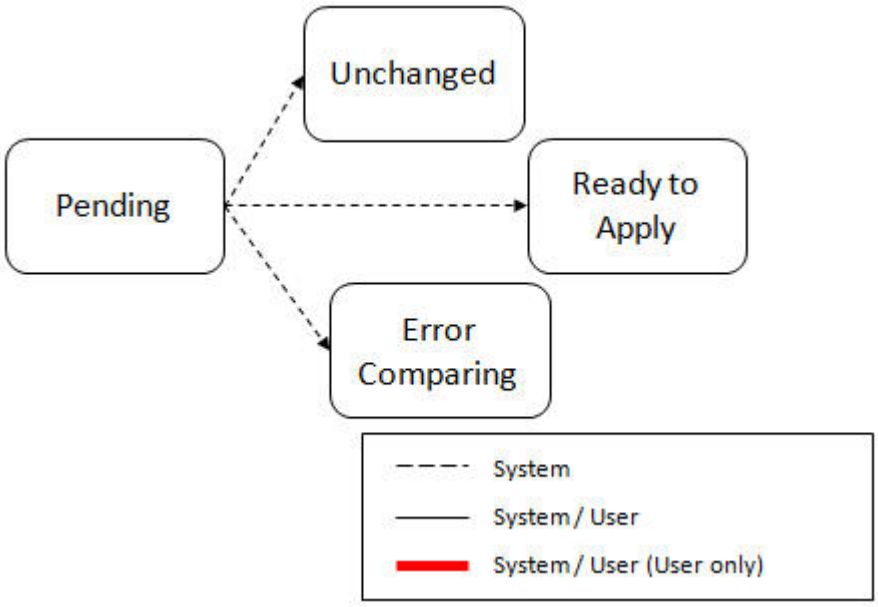
    * If some of the included entities are new and some changed then the object action value is set to **Mixed Actions**), corresponding SQLs are generated for the each entity and the migration object transitions based on the Default Status For Change setting captured on the Data Set.

- If there is any issue with attempting to parse the object data from the import, the record transitions to **Error Comparing**.
- If there is any reason that the imported object is not valid for import, the record transitions to **Cannot Apply**. The log will be updated with the error that caused the record to transition to this state. An example is that perhaps the record was exported in a different version of the product and has additional elements that are not recognized in this version.

> **Note**
>
> Refer to Cancelling a Data Set for information about cancelling a data set and its impact on its related objects.

Migration Transaction Status Update

After the import step, the migration transaction remains in the Pending state until all its objects have completed the comparison step. At that point, the status of the transactions should be updated based on the results of their objects. The **Migration Transaction Monitor** (F1–MGTPR) selects pending migration transaction records and runs its monitor algorithms. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates a portion of the migration transaction lifecycle that pertains to comparison.



The following points describe the lifecycle possible next states after Pending.

- If any related migration object is in the Error Comparing state, the transaction transitions to **Error Comparing**.
- If all related migration objects are in the Unchanged state, the transaction transitions to **Unchanged**.

Copyright © 2026, Oracle and/or its affiliates.

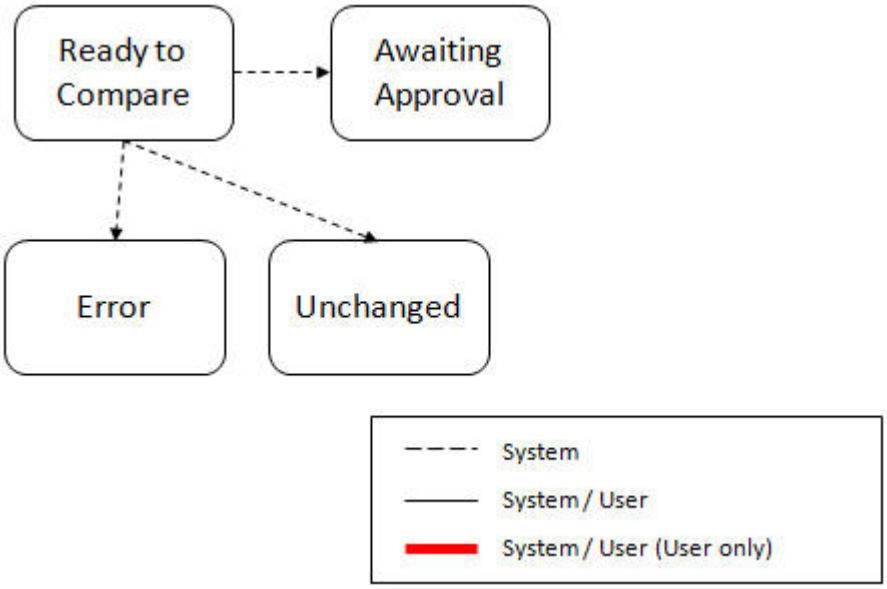
- Otherwise, the transaction transitions to **Ready to Apply**. This means that at least one object is in an “apply-able” state.

The transaction remains in the **Ready to Apply** state until a user has approved the data set to move to the Apply step and the transaction’s related objects have attempted to apply themselves. This is described in more detail below.

> **Note**
>
> Refer to Cancelling a Data Set for information about cancelling a data set and its impact on its related objects.

Migration Data Set Import Status Update

Once all the objects and all transactions have been updated via the previous two steps, the migration data set export must be updated based on the results of their transactions. The **Migration Data Set Import Monitor** (F1-MGDIM) selects Ready to Compare data sets and runs its monitor algorithms. Note that this is the same monitor process that is used to select Pending data sets. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates the portion of the migration transaction lifecycle that pertains to comparison.



The following points describe the lifecycle possible next states after Ready to Compare.

- If any related migration transactions is in the Error Comparing state, the data set transitions to **Error**.
- If all related migration transactions are in the Unchanged state, the data set transitions to **Unchanged**.
- Otherwise, the transaction transitions to **Awaiting Approval**. This means that there are no errors and at least one object is in an “apply-able” state.

The data set remains in the **Awaiting Approval** state until a user decides that the data set and all its records are ready to progress to the Apply step.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> A user can choose to cancel a data set at any time while it is in progress. Refer to Cancelling a Data Set for more information.

Approval Step

Once the comparison is complete and the data set transitions to the Awaiting Approval state, a user needs to progress the data set to **Apply Objects** to trigger the Apply step. The following points describe steps a user may take during the approval step.

- If the data set configuration for the Default State for Add and Change was set to **Approved**, then any migration object that is determined to be eligible for the Apply step will be in the Approved state. In this situation, a user may want to review the data set and its transactions and objects to see verify that the results make sense. At that time, the user is able to move an object to Needs Review or Rejected as appropriate.
- If the data set configuration for the Default State for Add and Change was set to **Needs Review** for either option, then each migration object in the Needs Review state must be reviewed and the user must either Reject or Approve each object before moving to the Apply step.
- If the data set configuration for the Default State for Add and Change was set to **Rejected** for either option, the assumption is that the rejected records don't need to be reviewed. But if a user finds a rejected record that shouldn't be rejected, it may be transitioned to Approved (or Needs Review) as appropriate.

A user has the option to mark one or more SQLs within a migration object to be suppressed. In this case, if the migration object is approved, then only the SQLs that are not marked as Suppressed will be applied. This option is useful when importing a record that has many child records. It may be that the user wants to only include a subset of the child records. This may be done by suppressing the child records that should not be included.

Once the user is comfortable with the data set's results and no more objects are in the Needs Review state, the user should transition the record to **Apply Objects**. This will initiate the Apply step.

Alternatively, if the Automatically Apply flag was set to **Yes** when creating the import record, the import data set will progress from **Awaiting Approval** to **Apply Objects**. Refer to Import Process Summary for more information.

> **Note**
>
> Refer to Maintaining Import Data for details about the pages provided to help the user review a data set and its transactions and objects to help in the approval step.

> **Note**
>
> A user can choose to cancel a data set at any time while it is in progress.

Apply Step

Copyright © 2026, Oracle and/or its affiliates.

The apply step is the step where records in the target environment are added or updated. Like the comparison step, the apply step is actually multiple steps to optimally handle high volume and dependencies between records as smoothly as possible.

> **Note**
>
> Refer to Running Batch Jobs for more information about streamlining the various steps in the process.

Before explaining the apply steps in detail, the following points highlight the type of data that may be included in a given data set.

1. Records that have no foreign keys and therefore no dependencies on other records. Examples: Message, Display Profile.
2. Records that have foreign keys that may already be in the target. Examples: Algorithms for existing algorithm types, To Do Roles for existing To Do Types.
3. Records that have foreign keys that are new records but also part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: Script-based Algorithm Type where the script is also in the migration.
4. Records that have foreign keys that are new records but also part of the migration. CMA did not detect the relationship. This may occur if the referenced foreign key is in a XML or parameter column and the migration plan did not include an instruction to explicitly define the relationship. Example, a Zone that references a visibility script.
5. Records that have circular references where both records are new and are part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: plug-in Script for a BO plug-in spot. The script references the BO and the BO references an algorithm for the script's algorithm type. Another example is when the same record is maintained by multiple maintenance objects and therefore exists in multiple migration objects.

To handle high volume data, the first step in the apply process is to perform the apply logic at the migration object level via a multi-threaded batch job. This should result in all records in categories 1 and 2 above being applied successfully.

For records in categories 3 and 4 above, if a record with a foreign key is added or updated before its related record, the validation will fail. However, if the related record is added first and then the record referring to it is added, validation will pass. The tool handles these dependencies as follows:

- Dependency between master and transaction entities is typically hierarchical and in most cases straight-forward. The tool leverages that knowledge to orchestrate the processing of objects in an optimal way that follows their dependency order as much as possible. Note that relationship between entities could be complex and this approach does not eliminate all processing order related errors but rather significantly reduce them.
- Dependency between configuration entities is more complex and inter-twined and therefore migration objects are not ordered, i.e. the multi-threaded batch process may not process records in the desired order.
- To overcome the potential issue of processing order related errors, the Apply step has special functionality, described in detail below.

For records in category 5 above, the circular reference will mean that the apply process at the object level will not successfully add or update these records. The apply process at the transaction level will cover these records. This is described in detail below.

Copyright © 2026, Oracle and/or its affiliates.

Apply Objects

Once the Data Set is in the state of **Apply Objects**, the **Migration Object Monitor - Apply** process (F1-MGOAP) runs to attempt to apply the objects.

> **Note**
>
> When using separate batch processes for business data the **Migration Object Monitor (Business)- Apply** process (F1-MGOAB) works in the same way to apply business migration objects.

The background process in conjunction with the Apply algorithm have special functionality to ensure records in categories 3 and 4 (above) successfully apply during this step:

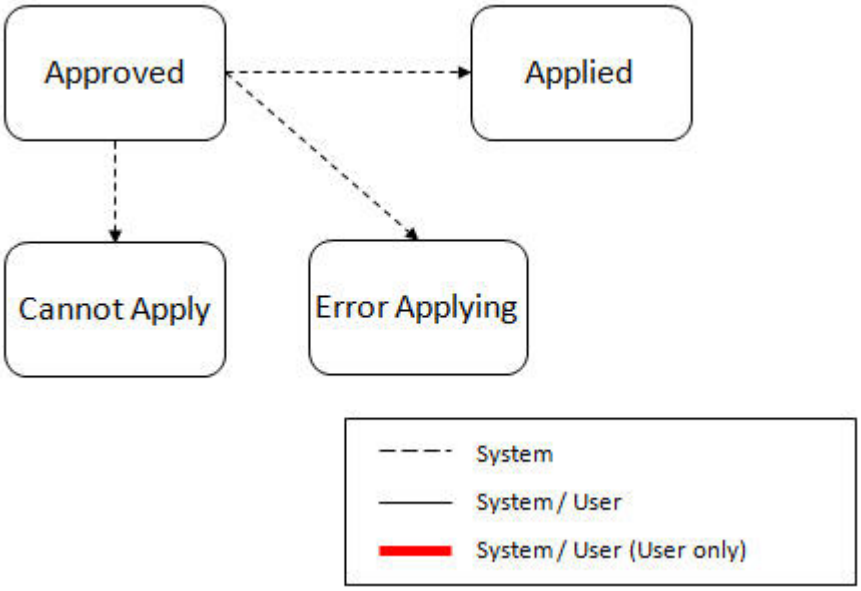
- The **Migration Object Monitor - Apply** process is a special one that continually re-selects records in the **Approved** state until there are no more eligible records to process.
- When an error is received in the Apply Object algorithm, the algorithm increments an "iteration count" on the migration object record. If the iteration count does not exceed a maximum count (noted in the algorithm), the object remains in the **Approved** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum defined in the algorithm, the record transitions to the **Error Applying** state.

> **Note**
>
> When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the ‘excess’ threads to wait for the supported number of threads to finish.

> **Note**
>
> Each retry iteration to apply an object after an error slows down the Apply batch process. When the quality of the imported data is suspected to result in many errors, you may use the **Maximum Retries** batch parameter. Refer to the batch control's parameter description for more information.

The following diagram is the portion of the migration object lifecycle that pertains to the Apply step.

Copyright © 2026, Oracle and/or its affiliates.



At the completion of the Apply monitor process, typically the objects will be in the **Applied** state or the **Error Applying** state. The records in the Error Applying state are in that state for one of two reasons.

- They are in category 5 described above where the records have a circular reference with another record. For this scenario, the Apply Transactions step described below should successfully apply the records.
- There is some other error that is unrelated to the records in the current migration. In this case, manual intervention may be required. Refer to the Resolving Errors section below for more information.

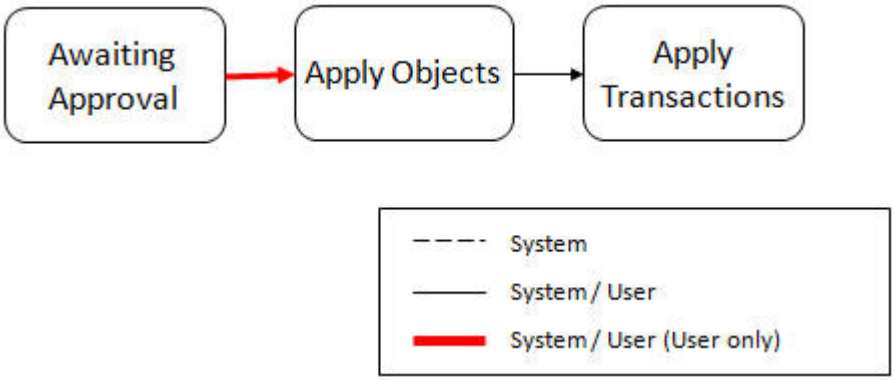
As shown in the diagram, the Apply Objects algorithm may also detect a reason that the object cannot be applied. This may occur if the object in the target environment has been updated since the comparison step, making the SQL captured at that point no longer applicable. If this occurs, after the current migration is fully applied, the original file may imported again, and new comparisons can be generated and applied.

Apply Transactions

Ideally, after the Apply Objects step, all the objects are **Applied** or are in **Error Applying** due to the "circular reference" situation. The typical next step is to turn over responsibility to the transactions. The migration transactions can then attempt to apply their objects in bulk.

In order to ensure that multiple background processes are not trying to select migration objects to run the Apply step, the Transactions are only eligible to attempt to "apply my objects" if the Data Set is in the **Apply Transactions** state.

Copyright © 2026, Oracle and/or its affiliates.



A monitor algorithm (executed by the data set monitor batch process) on the Apply Objects state checks to see if all migration objects are no longer **Approved** or the count of records in **Error Applying** does not exceed a configured limit. If so, it automatically transitions the record to the **Apply Transactions** state.

If the number of objects in **Error Applying** exceeds a configured limit, the monitor algorithm does not automatically transition the record. In that case, a user must determine if the large number of errors can be resolved or manually transition to **Apply Transactions** (despite the large number of errors). The Resolving Errors section below describes alternative steps that the user may take if there are errors.

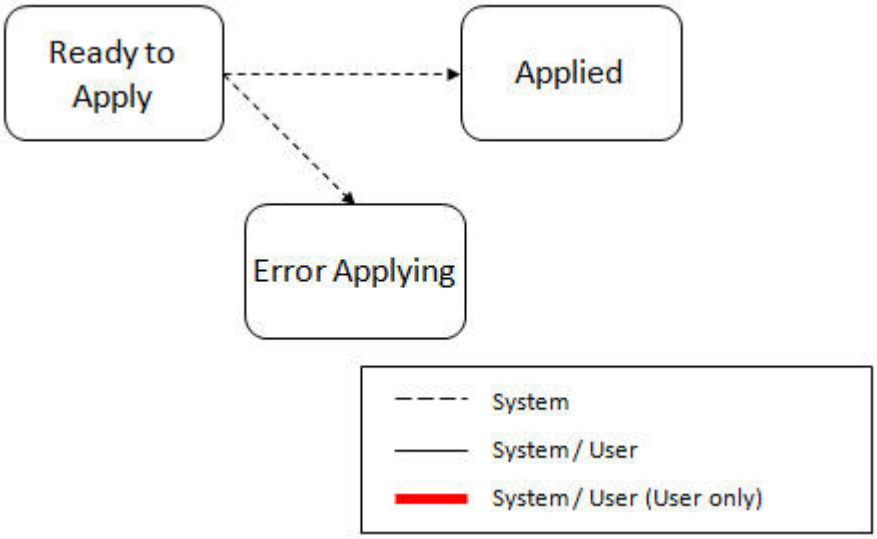
Once the Data Set is in the state of **Apply Transactions**, the **Migration Transaction Monitor - Apply** process (F1-MGTAP) runs. It attempts to apply the transaction's objects. If no migration objects are in error, the migration transaction simply transitions to **Applied**. If any of the migration objects are in **Error Applying**, the background process and the Apply algorithm have special functionality to try to overcome dependencies in migrated objects:

- The Apply algorithm selects all migration objects in error and performs all their SQL, then validates all the records. If there are objects in the transaction with circular references, they should pass validation at this point.
- Because there may still be some dependencies across transactions, similar error handling described in the Apply Objects step occurs here. When an error is received in the Apply Transaction's Object algorithm for any of the objects in the transaction, the algorithm increments an "iteration count" on the migration transaction record. If the iteration count does not exceed a maximum count (noted in the algorithm), the transaction remains in the **Ready to Apply** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum, the record transitions to the **Error Applying** state. Note that if any objects in the transaction are in error, none of the objects are applied. They all remain in error.
- The **Migration Transaction Monitor - Apply** process is a special one that continually re-selects records in the **Ready to Apply** state until there are no more eligible records to process.

> **Note**
>
> When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the 'excess' threads to wait for the supported number of threads to finish, erasing the benefit of the iteration processing.

The following diagram is the portion of the migration transaction lifecycle that pertains to the Apply step illustrating the points above.

Copyright © 2026, Oracle and/or its affiliates.



If at the end of the transaction level Apply process there are transactions in error (and therefore there are still objects in error), a user must review the errors and determine how to fix them. Refer to the Resolving Errors section below for more information.

Resolving Errors

As mentioned in the previous sections, errors may be received after the Apply Objects process runs. If the number of records in error is below a certain limit (and the data set monitor batch job is submitted to execute the monitor algorithms) the system will automatically transition the data set to the **Apply Transactions**. If the monitor batch job is not run or if the number of objects in error exceeds a certain limit, a user must make the decision after viewing the errors in the Objects in Error zone on the Migration Data Set Import portal.

- If the errors appear to be dependency related, the user can decide to let the "transactions apply their objects" and transition the data set to **Apply Transactions**, described above.
- If the errors appear to be related to an outside issue that can be manually resolved, the user may choose to fix the issue and redo the Apply Objects step.
- The user may also decide to reject one or more objects to remove them from the migration.

After the Apply Transactions step, if there are still errors, a user must review the records and determine how to proceed. Errors are visible in the **Transactions in Error** zone on the Migration Data Set Import portal.

- The user may decide to reject one or more objects to remove them from the migration.
- The user may manually resolve an issue external to the migration and then decide to do one of the following:
  - Redo the **Apply Objects** step. This is recommended if there are a large number of Objects still in error and not a large number of dependencies expected. The benefits of running the Apply Objects multi-threaded will ensure that the process runs efficiently.
  - Redo the **Apply Transactions** step.

Because the objects and transactions are in Error Applying, in order to "retry" the Apply step after manually fixing an error, the system needs to move the records back to the state that allows them to be picked up by the appropriate Apply process. For migration objects, records need to be moved back to **Approved**. For migration transactions, records need to be moved back to **Ready to Apply**. The following points describe the Retry logic for migration objects.

Copyright © 2026, Oracle and/or its affiliates.

- If a user decides to **Retry Objects** (using an action button on the Migration Data Set Import page), the data set transitions to the **Retry Objects** state. At this point the Migration Object monitor must be run.
- The monitor on the **Error Applying** state for the objects detects that the data set is in the state of **Retry Objects** and that triggers the transition back to **Approved**.
- The next step is to transition the data set from **Retry Objects** to **Apply Objects**. This may be done manually or by running the Migration Data Set Import monitor process.
- Now the objects are eligible to be picked up by the object level Apply process.

Analogous logic exists for the migration transactions.

- If a user decides to **Retry Transactions** (using an action button on the Migration Data Set Import page), the data set transitions to the **Retry Transactions** state. At this point the Migration Transaction monitor must be run.
- The monitor on the **Error Applying** state for the transactions detects that the data set is in the state of **Retry Transactions** and that triggers the transition back to **Ready to Apply**.
- The next step is to transition the data set from **Retry Transactions** to **Apply Transactions**. This may be done manually or by running the Migration Data Set Import monitor process.
- Now the transactions are eligible to be picked up by the transaction level Apply process.

The retry logic may also occur when transitioning between the Apply Objects and Apply Transactions depending on whether or not there are errors. The following scenario highlights this point.

- After the **Apply Objects**step there are objects in **Error Applying**. The data set transitions to **Apply Transactions** and the Apply step is done at the transaction level.
- After the **Apply Transactions** step there are transactions in **Error Applying**.
- User chooses to try to apply objects again (by clicking **Retry Objects**). The steps outlined above for retrying objects are followed at this point.
- After the apply objects, user may choose to retry objects again (after fixing errors if applicable).
- At some point the user will transition to**Apply Transactions** again. If there are transactions in **Error Applying**, the system will automatically transition the data set to **Retry Transactions** and the steps outlined above for retry transactions are followed.

Finalize Apply Step

Once all the migration objects for a migration transaction are in a final state (**Applied**, **Rejected** or **Cannot Apply**), the migration transaction transitions to the **Applied** state. Once all the migration transactions are in the **Applied** state, the Migration Data Set record transitions to the **Completed** and the import is complete.

> **Note**
>
> To review the full lifecycle for each record, refer to the Business Object - Summary tab in the application metadata for the base business objects **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).

Copyright © 2026, Oracle and/or its affiliates.

Adjusting Data Prior to Comparing

There are reasons that the data you are importing requires some adjustment before applying it in the target environment. In such cases, an algorithm plugged into the migration plan primary instruction may be used to adjust the data when importing. This algorithm is executed by the comparison algorithm before any comparison is performed. Algorithms of this system event receive the view of the source record (being imported) and the view of the existing record in the target region, if it exists. The data is provided using the physical BO of the migration plan's maintenance object. The algorithm may make changes and pass a new view of the record that should be used for the comparison. This system event supports multiple algorithms that are executed in sequence. Each algorithm receives the original record's data, the target record's data (if applicable) and the 'new' view of the data (as populated by previous algorithms, if any). The final 'new' view of the data is used for the object comparison. The system includes the algorithms linked to the primary instruction of the migration plan from the source environment. If the system finds any additional pre-compare algorithms linked to the migration plan's primary instruction in the target system, those algorithms are also executed.

> **NOT_SUPPORTED**
>
> Refer to Base Business Objects for more information about physical BOs.

Some examples of records that may require import algorithms.

- Batch Control references its next batch sequence number along with snapshot information like the last run date / time. This information is only relevant with respect to its environment. The instruction for a batch control can include an algorithm to not overwrite the batch sequence number when copying a batch control.
- Some products include administrative objects that reference a master data object. Master data objects are not copied as part of CMA. An import algorithm may be used to adjust the referenced master data foreign key when importing, for example to reset it (or not overwrite when updating). If the algorithm knows how to find the appropriate master data record to link, that may also be included.
- If you are importing data that from a source environment on a lower version, if the object you are importing has been enhanced in the latest release with a new required field, you may need to provide an algorithm to populate data in that field.

Note that it is possible to use the algorithm to "reset" the source data as a way of indicating that the record should not be imported. For these situations, the migration object comparison step will transition the record to **Unchanged** and will use an object action value of **Canceled**. (Note that object action is a simple lookup value. The record is not transitioned to the **Canceled** BO state as to reserve that status for user initiated cancellations of the object or one of its parent records). This technique not expected to be used often because ideally using appropriate selection criteria at export time should ensure that the only records exported are those that should be imported.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Legacy 'Import' system event.** The system originally provided an Import system event / plug-in spot. The purpose of algorithms for this plug-in spot were similar in that they were meant to adjust imported data prior to adding or updating. The algorithms were executed in the Apply step. The logic does not allow for easily interacting with the record using a BO. This makes it difficult to use a plug-in script as the plug-in type. In addition, it is difficult to update elements in an XML column. The support for the plug-in spot will be removed in a future release. Algorithms to adjust the data should be using the pre-compare system event.

Import Process Summary

The following table summarizes the steps required to complete the import process from start to finish. Note that this section **only a summary** and assumes that you are familiar with the details described in the previous sections. It highlights what steps are manual and what steps are performed by a batch monitor process. For each step, the table highlights the Next Action sequence that would occur. For the Apply steps, there are two parts where multiple next actions are possible based on whether there are errors and the user's decision on how to resolve the error. Refer to Resolving Errors for more information. The possible next actions have the same sequence with a letter following the sequence highlighting the action to take based on the results of the previous step.

> **Note**
>
> When running the Apply batch jobs, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.

Also note that a sequence and action marked in bold is considered the "normal path".

When using separate batch processes for business data any reference to a migration object monitor batch process applies also to the business data related batch controls.

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Add-Amount Tracking Detailed - Resulting Status	Next Action Sequence
Import	**10**	**Create Import record**	Manual	Mig Iggraat ii oonn DDaat aa SSeet IImpoort - APdedn ding	11

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	11	Import file	Batch	FM 1-Mag Dim-Mag Stat Dim Pro Stat Sen Ind Pro	20

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Load-Amortization	Next Action Sequence
				Completed - Resulting Status	
				Completed - Migration Transaction - Pe	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Pending - Resulting Status	Next Action Sequence
				nding Migration Object - Pending	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Contractor Cd-Amount Contracted - Resulting Status	Next Action Sequence
Compare	**20**	**Migration Object Compare**	Batch	FM1 1-Mig Ga OP Ro-n Mib g reactti-o Ap Op br jo ev ce td Mo N re ie td s	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Procedure Control - Amount Tracked - Resulting Status	Next Action Sequence
				Rev. Wewn Rejected Unchanged Error	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Amount Cost-Adjusted - Resulting Status	Next Action Sequence
				to a Process Cost-Adjusted - Resulting Status	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracked	Next Action Sequence
				Migration Object Monitor (Business	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Manual / Batch Action	Next Action Sequence
			[Perform Control Amount Tracing Ordered - Resulting Status s))]

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	21	Migration Transaction status update	Batch	FM 21-Mar Tr Pro-Mirgam astaic ori Toma-Rse acdty it oon Ap	22

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked	Next Action Sequence
				Resulting Status	
			te	Priority, Unchanged or Error Comparing	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	22	Migration Data Set Import status up	Batch	FM 1-g MGA DI M-n Mia g ra a Ste it d m Dp a tr a S- e A tw Ia i m p t d m	30

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amortization	Next Action Sequence
		date		Mgmt to overall, Apply Objects (if config	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
				ured for Automatic Apply), Unchang	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracking Stated - Resulting Status	Next Action Sequence
				ed or Error	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
Approval	**30**	**Review comparison results, approv**	Manual	Mig rat ion Dat ae Set - Ap pro ro t v, e dd ro ir l R le ij ne	40

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Accounted - Resulting Status	Next Action Sequence
		e/rejectasneeded (This step is skip		to the Tranes action/Obj beeitsNaesendesc	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Perform Action Sequence	Next Action Sequence
			ppedifthedatasetisconfigured for	eesviaerwy) · Migration Data Set Import - A	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			Automatic Apply)	pply Objects	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Manual / Batch Action	Next Action Sequence
Apply	40	Batch Apply Objects	41 Appropriate next action is based on error review, if applicable.

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
				Apply [When using separate business	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
				essdataprocessessubmitalsoF1-M	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amortization	Next Action Sequence
				GOAB (Migration Object Monitor (Bu	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Actioned - Resulting Status	Next Action Sequence
				siness) - Apply)]	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	41a	Migration Data Set Import status up	Batch	F-42 1-Mig Data Import Mgr Start Data Set Import Process Action Sequence	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Contract Code	Next Action Sequence
		date-auto transition to Apply Tran		Manual Actions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			sactions. Only applicable if the nu		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Error Action	Error Action Sequence	Error Action Sequence-dependent Amount	Error Action Sequence-dependent Resulting Status

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			pplying is below a threshold		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Actioned - Resulting Status	Next Action Sequence
	41b	**Migration Data Set Import status up**	Manual	Migration Data Set Import - click Ap	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Accounted - Resulting Status	Next Action Sequence
			date-manual transition to Apply Tr	ply Transactions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		ansactions. Occurs if user reviews			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		errors and determines that they may			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Percentage Cd-Amount Cd-Resulting Status	Next Action Sequence
			be resolved in the Apply Transactio		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			n s t e p .		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Manual / Batch Action	Next Action Sequence
		System detects that all the transac	42 Migration Data Set Import - Apply Tr

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Load-Amount Tracking Stated - Resulting Status	Next Action Sequence
		tions are in the Ready to Apply state		ansactions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		and proceeds to the Apply Transacti			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Procedure Control - Amount Tracked - Resulting Status	Next Action Sequence
		onsstate.			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Manual / Batch Action	Next Action Sequence
		System detects that there are trans	45 Migration Data Set Import - Retry Tr

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		actions in the Error Applying state		ansactions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracking Stated - Resulting Status	Next Action Sequence
		and transitions instead to the Retr			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Parent to a Ind-Amount Transaction	Next Action Sequence
		y Transactions state.			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	41c	Migration Data Set Import status up	Manual	Migration Data Set Import - Clerk Web	44

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracking Stated - Resulting Status	Next Action Sequence
			date-manual transition to Retry Ob	tryc Objects	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		jects. Occurs if user review error			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracking Stated - Resulting Status	Next Action Sequence
		sand decides to fix an external error			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		and wants to try the batch level Appl			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		yagain at the object level.			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Next Action Sequence
	42	Apply Transactions	Appropriate next action is based on error review, if applicable.

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Load-Amortization	Next Action Sequence
				Error to or Applying Migration Object - A	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
				Applied or Error Applying	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Load-Amount Transaction Detailed-Resulting Status	Next Action Sequence
	43a	Migration Data Set Import status up	Batch	FN/A 1-g MGa DI M-n Mia g ra a Ste it d m Dp a tr at S-e C to Im p l oe rt	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Perform Action Code	Next Action Sequence
		date-autotransitionto Completed		Monitor	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracked	Next Action Sequence
		Applicable if all transactions are			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		Applied			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracking Stated - Resulting Status	Next Action Sequence
	43b	Migration Data Set Import status up	Manual	Migration Data Set Import - Clerk Web	44

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
		date-manual transition to Retry Ob		Retry Objects	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		jects. Occurs if user review error			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to an IC-Amount Trace Ordered - Resulting Status	Next Action Sequence
		sand decides to fix an external error			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracked	Next Action Sequence
		and wants to try the batch level Appl			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		yagain at the Object level.			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	43c	Migration Data Set Import status up	Manual	Migration Data Set Import - Clerk Reactor Amount Transaction Detailed - Resulting Status	45

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			date-manual transition to Retry Tr	tary Transactions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		ansactions. Occurs if user reviews			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to a Local Amount Adjustment Needed - Resulting Status	Next Action Sequence
		errors and decides to fix an external			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracking Stated - Resulting Status	Next Action Sequence
		error and wants to try the batch leve			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracking Stated - Resulting Status	Next Action Sequence
		l Apply again at the Transaction lev			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			el.		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	44	Migration Objects status update fr	Batch	Fi-Mig Ca Oil Pro-Mig reactio Ap Oprjo evoted Monit	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Reached - Resulting Status	Next Action Sequence
			om Error Applying back to Approved.	or [When using separate business da	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Accounted - Resulting Status	Next Action Sequence
		Occurs if user chose to Retry Object		tap rocesses submitals o F1-MGOPB(	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracked	Next Action Sequence
			s.	Migration Object Monitor (Business	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Next Action Sequence
				[Perform Control Amortization - Resulting Status s))]

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Code	Next Action Sequence
			Batch	F-401-Migration Data Set Imports status up	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		date from Retry Objects to Apply Obj		tj Mcrtistor	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			ects		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
	45	Migration Transaction status upd	Batch	FM-42 1-Mig Ga Tr Pro-Mirgam astaic on Toma-Rsea cdty it oon Ap	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Error Action Sequence	Next Action Sequence	Error Action Sequence - Resulting Status

Step	Seq	Action	Manual / Batch	Process to Apply-Amount Adjusted-Resulting Status	Next Action Sequence
		y to Apply. Occurs if user chose to Re			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control-Amount Tracked	Next Action Sequence
		try Transaction so rif the user tran			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Apply Transactions and Reconcile Amount - Resulting Status	Next Action Sequence

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
		hesystemdetectsthattherearerTra			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Accounted - Resulting Status	Next Action Sequence
		nsactionsintheErrorApplyingsta			

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			te.		

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process Action Sequence	Next Action Sequence
		Migration Data Set Import status up	Batch	FM-42 1-Migration Data Set Import Mapping Status Setup Trr	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Load-Amount Transaction Detailed - Resulting Status	Next Action Sequence
		date from Retry Objects to Apply Obj		Mosaic Actions	

Copyright © 2026, Oracle and/or its affiliates.

Step	Seq	Action	Manual / Batch	Process to Control Amount Tracked - Resulting Status	Next Action Sequence
			ects		

The following table summarizes the batch monitor jobs that are used in the import process. You can see that there are special monitor processes for the Apply step for both the Object and Transaction. However, for all other states that have monitor logic, the standard monitor process for that MO is used.

Batch Control	Description	Comments
F1-MGDIM	Migration Data Set Import Monitor	Processes data set records in the following states: • Pending • Ready to Compare • Apply Objects • Retry Objects • Apply Transactions • Retry Transactions
F1-MGTPR	Migration Transaction Monitor (Deferred)	Processes transaction records in the following states: • Pending • Error Applying

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Description	Comments
F1-MGTAP	Migration Transaction Monitor - Apply	Processes transaction records in the Ready to Apply state where the data set is in the Apply Transactions or Canceled state. **Note** Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.
F1-MGOPR	Migration Object Monitor	Processes object records in the following states: • Pending • Needs Review (check for Data Set cancellation) • Error Applying
F1-MGOPB	Migration Object Monitor (Business) Applicable only when separate business data processing is enabled.	Similar to F1-MGOPR (Migration Object Monitor) but restricted to business data migration objects.

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Description	Comments
F1-MGOAP	Migration Object Monitor - Apply	Processes object records in the Approved state where the data set is in the Apply Objects or Canceled state. **Note** Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.
F1-MGOAB	Migration Object Monitor (Business) - Apply Applicable only when separate business data processing is enabled.	Similar to F1-MGOAP (Migration Object Monitor - Apply) but restricted to business data migration objects.

Refer to Running Batch Jobs for more information about managing the batch jobs, including ways to automate the above steps.

Cancelling a Data Set

A user may choose to **Cancel** a data set to prevent it from being processed at any point during the process.

The request sets the data set in **Pending Cancel** status. If related migration objects have not yet been created, the data set transitions immediately to the **Cancelled** status and the cancellation request is complete.

If related migration transactions or migration objects have already been created, they will not be canceled as part of the data set getting canceled (due to possible high volumes of related records). They will be canceled the next time an appropriate monitor batch process runs. The child records checks to see if the data set has been canceled prior to any state transition.

Once all the migration transactions and objects are in a final state, the next batch monitor processing of the data set transitions it to the **Cancelled** status to complete the cancellation request.

Copyright © 2026, Oracle and/or its affiliates.

Additional Note Regarding Imports

The following points describe miscellaneous comments related to Migration Import.

- CMA relies on the fact that database referential integrity constraints are not in place, and that the SQL statements can be run in any order within the transaction. Any archiving solution that requires referential integrity constraints (such as Information Lifecycle Management) would not be possible on this data. Given that CMA migrations comprise administrative data and not transactional data, this should be a reasonable exception.
- The validation that is performed is only via the **Page Validate** service. BO validation algorithms are not executed. Page validation does not include validation of the business object against the schema (for example, for required fields, field sizes, etc.).
- If multiple migration requests are exported at the same time, on the import side, you should consider importing, reviewing, and applying an entire file/data set before moving on to the next one. The reason is that if objects are included in more than one file, two sets of "inserts" will be generated, but only the first will succeed. The second will cause the object to transition to "Cannot Apply". If instead you wait until the first file is completed before importing the second, the second data set will not generate any SQL for the object, since it has already been inserted. It's a matter of efficiency: If you first import all files and then try to apply all, you'll have to identify the duplicated object as an error and then mark the object as rejected before applying the transaction. This may also be avoided by using a Group migration request to include all objects in one file rather than multiple files.
- The system provides an algorithm to purge "unchanged" migration objects for a given migration data set. This may be plugged in as a BO exit algorithm on the **Ready to Compare** state for the Migration Data Set Import business object (`F1-MigrDataSetImport`).

The following points describe miscellaneous comments related to importing a business data set in bulk mode.

- The number of entities included in each migration object is captured on the migration object record.
- The comparison step only generates SQLs for new or changed entities. SQLs are not generated for unchanged entities.
- Searching for migration objects based on their included entities is only supported for entities that were determined as new or changed. In other words, this type of search by included entities is only possible after the comparison step is performed.

Caching Considerations

There are two important caching considerations when using CMA to migrate administrative data.

- The "apply" batch processes update data. As described in Batch Cache, if any batch job is updating data marked as "cached for batch" it must be run in a special thread pool. Contact your system administrator and ask for the name of a thread pool that has turned off caching. This thread pool name must be entered in the batch controls used for CMA processing in the **Thread Pool Name** parameter.
- After a successful migration, the target region now has new administrative data which needs to be part of various caches. It is recommended to flush the server cache (which will trigger a 'global' flush of the cache). If the thread pool workers in the target region are configured to refresh their caches when a global flush is requested, then this is the only

Copyright © 2026, Oracle and/or its affiliates.

step required. If not, then the F1-FLUSH batch job should also be submitted to refresh the caches used in batch processing.

NOT_SUPPORTED

Refer to Caching Overview for more information.

NOT_SUPPORTED

Refer to Running Batch Jobs for more information about running CMA batch jobs.

Maintaining Import Data

This section describes the portals provided to add, view and maintain migration import data.

- Migration Data Set Import
- Migration Transaction Portal
- Migration Object Portal
- Configuration Deletion

Migration Data Set Import

Use the Migration Data Set Import portal to view and maintain migration data set import records. Refer to Importing and Applying a Migration for an overview of the import process.

Navigate using **Admin**, and then **Implementation Tools**, and then **Migration Data Set Import.** You are brought to a query portal with options for searching for import data sets. In addition, the query provides an option to specifically search for data sets that have either objects in error or transactions in error. Once a data set has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are visible on the **Main** tab:

- **Migration Data Set Import.** This zone contains display-only information about the selected record. Please see the zone's help text for information about this zone's fields.
- **Entity Summary.** This zone provides a summary count of all migration objects included in the current data set across the various statuses.

The following zones are visible on the **Transactions** tab:

- **Migration Data Set Transactions in Error.** This zone is only visible if there are transactions for this data set in a non-final state that have errors. It indicates the error for each transaction. A user may use this zone to review errors after the monitor batch job to apply transactions completes. The errors received when attempting to apply objects at the transaction level may differ from those received when attempting to apply objects at the object level. A transaction log is created for each object error received and these exceptions are shown in this zone.

Note

Refer to Apply Step for more information about resolving errors.

Copyright © 2026, Oracle and/or its affiliates.

- **Migration Data Set Transactions.** This zone is visible once the Import Step has occurred and lists all the transactions that are related to the data set. To see more information about a specific migration transaction, click the hypertext for its ID. This brings you to the Migration Transaction portal.

The following zones are visible on the **Objects** tab:

- **Migration Data Set Objects in Error.** This zone is only visible if there are objects for this data set in a non-final status that have errors. It indicates the error for each object. A user may use this zone to review errors after the monitor batch job to apply objects completes. Using the error information shown, the user can choose to drill into the record to transition it to **Error Applying** or choose to manually fix the cause of the errors and click **Retry Objects**. The user may also choose to select one or more records to **Reject**.

> **Note**
>
> Refer to Apply Step for more information about resolving errors.

- **Migration Data Set Impacted Object Summary.** This zone is visible once the Import Step has occurred and lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the Migration Object portal. A user may choose to update the status of one or more records by checking the records and clicking **Approve**, **Reject** or **Needs Review** accordingly.

The following zones are visible on the **Errors** tab:

- **Errors Summary.** This zone summarizes migration objects in error by maintenance object, error message category and number. Broadcasting a row provides a more detailed list of distinct error messages.
- **Errors.** This zone lists distinct errors for a selected maintenance object along with a count of migration objects in error with the same exact error message. This information may be used to highlight common configuration issues across objects. A sample migration object is also provided for review. The zone is visible when a selection is made on the summary zone.

The following zones are visible on the **Log** tab, in addition to the standard log entries:

- **Object Retries Summary.** This zone provides a summary of retry processing performed while applying objects. Retrying objects too many times slows down the apply step. Incomplete or invalid export data may result in many objects being retried too many times until they are eventually set to 'Error Applying' status.

Migration Transaction Portal

This page appears after drilling into a specific migration transaction from the migration data set portal or from the migration object portal.

Refer to Importing and Applying a Migration for an overview of the import process.

The following zones are visible on the main tab:

- **Migration Transaction.** This zone contains display-only information about the selected record. Please see the zone help text for information about this zone's fields.
- **Migration Transaction Objects.** This zone lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the Migration Object portal. A user may choose to update the status of

Copyright © 2026, Oracle and/or its affiliates.

one or more records by checking the records and clicking **Approve**, **Reject** or **Needs Review** accordingly.

- **Entity Summary.** This zone provides a summary count of all migration objects included in the current transaction across the various statuses.

Migration Object Portal

This page appears after drilling into a specific migration object from the migration data set portal or from the migration transaction portal.

Refer to Importing and Applying a Migration for an overview of the import process.

The **Migration Object** zone contains display-only information about the selected record. Please see the zone help text for information about this zone's fields.

The **Migration Object - List of SQL Statements** zone lists the SQL statements that have been generated as a result of the comparison step. A check mark in the Suppressed checkbox indicates that the SQL for that row will not be applied during the apply step.

The **Migration Object - Algorithms** zone displays any pre-compare algorithms that are associated with the migration plan for this migration object.

Configuration Deletion

Use the Configuration Deletion portal to review and delete configuration entities that were previously imported and are no longer needed. Refer to Importing and Applying a Migration for an overview of the import process.

Navigate using **Admin**, and then **Implementation Tools**, and then **Configuration Deletion.** You are brought to a portal where you can define a set of import data sets as representing a complete imported configuration and delete entities that were previously imported but not included in the reference set.

The following zones are visible on the **Main** tab:

- **Configuration Summary.** This zone summarizes the number of imported entities by maintenance object that are included in a selected set of import data asets that represent a complete imported configuration. Selecting a maintenance object lists potential entities for deletion.
- **Entities to Delete.** This zone lists potential entities for deletion, i.e. entities of the selected maintenance object that were previously imported but not included in the reference data set. You may select and delete entities from this list.

Running Batch Jobs

There are several batch jobs that are part of the CMA process, especially the import step (which are highlighted in Import Process Summary). And in some cases, a single batch jobs may process multiple states in the same business object lifecycle. Implementations must decide the best way to manage the batch job submission depending on how they plan to work.

- **Batch scheduler**. If an implementation wishes to put these batch jobs in the batch scheduler, a given job may need to be included several times to manage progressing the records to completion.
- **Timed Batches**. The batch controls can be configured as timed batches so that they run every N minutes based on the setting. This allows for the batch jobs to run periodically and

Copyright © 2026, Oracle and/or its affiliates.

process whatever is ready. A user doesn't have to manually submit a batch request. Navigate to the Batch Control page and select the appropriate batch controls. For each one, change the Batch Control Type to **Timed**. Fill in the additional information that appears for timed batches.

- **Event Driven**. The system provides BO enter plug-in algorithms and batch control post processing plug-in algorithms that automatically submit the appropriate next batch job for that step in the process. This allows for as much automation as possible for the steps that don't require user input. Note that configuration is required because the BOs / batch controls are not configured for this scenario by default. The following table highlights the BO and status where an algorithm may be plugged in and the name of the algorithm to use.

Business Object	Status	Algorithm
Migration Data Set Export	**Pending**	`F1-MGDPR-SJ` (Submit Migration Data Set Export Monitor).
Migration Data Set Import	**Pending**	`F1-MGDIM-SJ` (Submit Migration Data Set Import Monitor).
	Ready To Compare, **Retry Objects**	`F1-MGOPR-SJ` (Submit Migration Object Monitor). When using separate batch processes for business data add also this algorithm: `F1-MGOPRB-SJ` (Submit Migration Object Monitor for Business Data).
	Apply Objects	`F1-MGOAP-SJ` (Submit Migration Object Apply Monitor). When using separate batch processes for business data add also this algorithm: `F1-MGOAPB-SJ` (Submit Migration Object Apply Monitor for Business Data).
	Apply Transactions	`F1-MGTAP-SJ` (Submit Migration Transaction Apply Monitor).
	Retry Transactions	`F1-MGTPR-SJ` (Submit Migration Transaction Monitor).

The following table highlights the batch controls where an algorithm may be plugged in and the name of the algorithm to use.

Batch Control	Algorithm
F1-MGTPR (Migration Transaction Monitor)	`F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor).
F1-MGTAP (Migration Transaction Monitor - Apply)	`F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor).
F1-MGOAP (Migration Object Monitor - Apply) When using separate batch processes for business data update also this batch control:	`F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor).
F1-MGOAB (Migration Object Monitor (Business) - Apply)	

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Algorithm
F1-MGOPR (Migration Object Monitor) When using separate batch processes for business data update also this batch control: F1-MGOPB (Migration Object Monitor (Business))	`F1-MGTPR-NJ` (Submit Migration Transaction Monitor).

- **Manual submission**. The user managing the CMA import process submits the appropriate batch jobs on demand when a particular step is ready. Navigate to Batch Job Submission, select the appropriate batch control and fill in the parameters as needed.

Note that for the CMA batch processes used to migrate administrative data, there are multiple important caching considerations. Refer to Caching Considerations for more information.

> ⚠ **Caution**
>
> Be sure that the Thread Count set when submitting the batch job does not exceed the number supported by the thread pool. Otherwise the extra threads will wait until the supported number of threads are finished, possibly resulting in a large number of errors in the Apply steps.

Refer to the parameter descriptions in the batch control metadata for more information about filling in the parameters.

For additional details on submission controls, refer to the topic Batch Job Submission Portal in the Batch Jobs section.

CMA Reference

This section provides additional reference information.

- Framework-Provided Migration Configuration

Framework-Provided Migration Configuration

This topic describes special information relating to migration objects provided for use by CMA in the product. Additional objects may be provided by your specific product. Any special information for objects is provided separately in each product's documentation.

The following points highlight some information about Framework-provided migration requests. Navigate to the migration request page in the application to view the details of all provided objects.

- Several base migration requests are supplied to logically group system and administrative tables. For example, there is a migration request for Framework System Configuration **F1-SystemConfig** where most system configuration objects are included. There is another one provided for CMA related configuration objects.
- The system supplies a group migration request `F1-FrameworkConfig` (Framework Configuration), which includes several other migration requests. The expectation is that this migration request includes all the typical objects that are included in a wholesale migration. Your specific product may include this migration request into its own group migration request to support a wholesale migration of all the framework and product administrative tables. An implementation may choose to build a custom group migration

Copyright © 2026, Oracle and/or its affiliates.

request. In this case, review the various migration requests provided by base to see if any may be included as components for the custom migration request. Then any new migration plans added to the base migration request in future releases are automatically included in future migrations.

- There are several different security related migration requests that include different combinations of migration plans to support multiple possible business requirements related to security migration. Note that the security migration request included in the above mentioned group migration request is the one that does not include users. If your implementation wants to copy users to a target environment, refer to Importing Data that References a User for some considerations.

> **Note**
>
> Refer to your specific product's CMA documentation for its recommendation on which migration requests to use for a full migration of framework and product administrative tables.

The following points highlight some information about the Framework-provided migration plans. Navigate to the migration plan page in the application to view the details of all provided objects.

- Fields and characteristic types are not migrated with an object (like a business object or a data area) unless specifically indicated.
- The **Application Service** used by an object is migrated only if it is CM-owned.
- The **Batch Control** object optionally references a User (for 'timed' batches). Refer to Importing Data that References a User for some considerations about copying the user. Also note that when running a batch job, snapshot information is captured on the batch control. Updates like this increment the version number. If a batch control record is part of the migration and the comparison step has detected a change to the batch control, the Apply step will error out for this batch control if a batch job is submitted between the compare and apply step.

> **Note**
>
> CMA batch controls that are part of the import step are executing and as such, the system does not include these records in a migration. If your implementation changes default parameters for any of the batch controls, the recommendation is to manually make those changes to the target region.

- The base migration plans for MO and BO include instructions to copy option types that use foreign key references to refer to other objects. Note that the data stored in the options are not validated, so defining these instructions is not required when doing wholesale migrations. However, including subordinate instructions for foreign key references is useful for targeted migrations to ensure that the related data is included in the migration. If you add additional MO or BO option types that use foreign keys and you want to support targeted migrations, you must create custom migration plans and requests for MO and BO, respectively to include these referenced objects in the migration plan. Note that you do not need to duplicate the instructions in the base migration plans. You may define the additional migration plans to only have the additional custom option types. When submitting a migration request for MO or BO you must include both the base migration plans and the custom migration plans in the request.
- For scripts, schema-based objects and zones, the migration plans provided by the product migrate, through constraints, some of the typical associated data with them. However, data

Copyright © 2026, Oracle and/or its affiliates.

specified through alternate formats (such as through **Edit Data** steps in scripts, referenced in schemas for schema-based objects, or data from mnemonics in zone parameters, etc.) are not identified and combined in the same transaction. The iterative processing functionality of the import step should resolve any timing issues that may result in validation errors for these types of objects.

- There are two migration plans for **Scripts**. The migration plan **F1-ScriptOnly** migrates just the script and its **Application Service** (provided the Application Service is CM-owned). The migration plan **F1-Script** includes most related objects, but does not migrate any objects referenced in the edit data area steps. It does not move the **Function** maintenance object. It may be included in any appropriate custom targeted migration request where scripts and related data should be migrated.
- If your implementation includes a **Feature Configuration** setting for the `F1_DBCONINFO` entry that will be included in a migration request, be sure that the import user on the target region has the appropriate security rights to this entry (**Administrator** access mode for the Feature Configuration application service (**CILTWSDP**).
- The common attachments in the Attachment maintenance object may be considered administrative data to include in a migration. Because this MO has a system generated key, as described in Administrative Data with System Generated Primary Keys, it uses a logical key of the file name and the creation date to determine if the record exists in the target environment. In addition, this MO contains administrative data (common attachments) and non-administrative data (owned attachments). To try to minimize the possibility of key "collision", new common attachments receive a generated key that includes a zero in the middle whereas owned attachments receive a generated key that does not have a zero in the middle.
- The Menu maintenance object has a user defined key, however, its menu lines and menu items have system generated keys. To avoid the possibility of overriding a menu line or menu item incorrectly, the menu MO will check the menu line's menu name in the source and target to be sure they match and will check the menu item's menu line in the source and target to be sure they match otherwise an error will be issued in the comparison step.
- For the system messages, the product provides three different migration plans.
  - Message Category and its Messages (F1-MessageCategory). This migration plan is included in the **F1-SystemConfig** migration request.
  - Message Category (F1-MessageCategoryOnly). This migration plan is provided to support a targeted migration where an implementation has created a custom message category and wants to move it but doesn’t want to move all its messages.
  - Message (F1-Message). This migration plan is provided to support a targeted migration where only specific messages within a message category should be migrated.
- For lookup values, the product provides two different migration plans.
  - Lookup Field and its Values (F1-Lookup). This migration plan is included in the **F1-SystemConfig** migration request.
  - Lookup Value (F1-LookupValue). This migration plan is provided to support a targeted migration where only specific lookup values within a lookup field should be migrated.
- There are some system data objects where no information in a base delivered record may be modified by an implementation. For these records, the base delivered migration requests include selection criteria to only select CM-owned records (because the base records will always exist in the target region assuming both regions have the same release). An example is Algorithm Type. The **F1-SystemConfig** migration request only includes CM-owned algorithm types. However, many system data objects support custom changes to one or more fields, for example the Zone object allows an implementation to

Copyright © 2026, Oracle and/or its affiliates.

override the zone text or certain parameters. Other system data objects support custom additions to a collection. For example, the Maintenance Object allows an implementation to add algorithms or options. For the migration plans related to these system data objects, all records are included in the base delivered migration requests to allow for any customized configuration to be migrated. It means that during the Import / Compare step many base delivered objects that are not customized will be marked **Unchanged**.

- Many of the integration related maintenance objects that include references to environment-specific data, such as Message Senders. This data should be migrated with extreme care. When appropriate, consider taking advantage of URI substitution. Refer to Referencing URIs for more information.

Facts

Fact is an optional configuration tool for simple workflow-type business messages or tasks. The base package does not provide a dedicated Fact user interface because fact is generic by design. Implementations configure their own user interface to visualize the desired custom business process. The topics in this section describe the generic Fact entity and how it can be customized.

- Fact Is A Generic Entity
- Fact's Business Object Controls Everything
- Fact Supports A Log

Fact Is A Generic Entity

The Fact maintenance object is a generic entity that can be configured to represent custom entities and support automated workflows for a variety of applications. Each fact references a business object to describe the type of entity it is. A status column on the fact may be used to capture its current state in the processing lifecycle controlled by its business object.

The maintenance object also supports a standard characteristic collection as well as a CLOB element to capture additional information.

Fact's Business Object Controls Everything

A fact's business object controls its contents, lifecycle and various other business rules:

- Its schema defines where each piece of information resides on the physical Fact maintenance object.
- If may define a lifecycle for all fact instances of this type to follow. Each fact must exist in a valid state as per its business object's lifecycle definition.
- It may define validation and other business rules to control the behavior of facts of this type.

NOT_SUPPORTED

For more information about business objects, refer to The Big Picture of Business Objects.

Copyright © 2026, Oracle and/or its affiliates.

Fact Supports A Log

The Fact maintenance object supports a log. Any significant event related to a Fact may be recorded on its log. The system automatically records a log record when the fact is created and when it transitions into a new state. In addition, any custom process or manual user activity can add log entries.

NOT_SUPPORTED

Refer to State Transitions Are Audited for more information on logging.

Conversion

This section describes the overall conversion process and the tools provided to support it.

- Understanding The Conversion Process
- Conversion Entity Dashboard

Understanding The Conversion Process

When you're ready to convert data from your legacy system, you will have analyzed your processing requirements according to your business and organizational needs and set up the control tables accordingly. After the control tables are set up, you are ready to load data into the system from your legacy system.

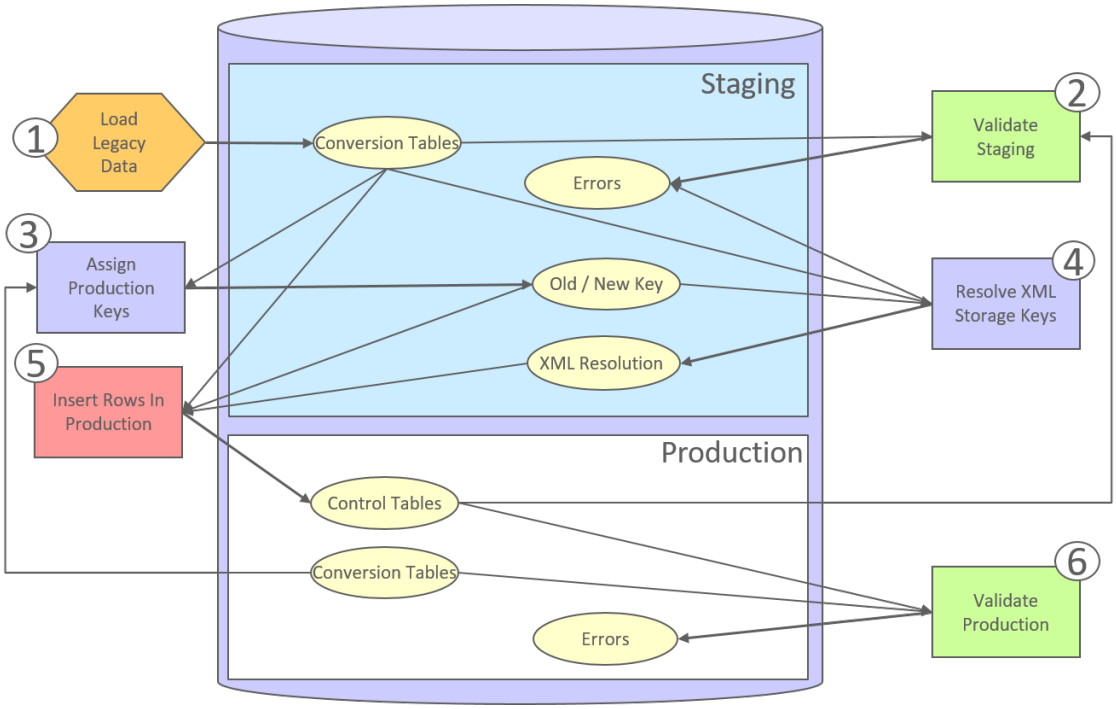
There are two table owners in the system database. We refer to the first owner as **staging** and the second owner as **production**. The staging owner is linked to the tables into which you insert your pre-validated data. The production owner is linked to the tables used by your production system.

Note

It's important to notice that **control tables exist only in production**. In staging, they exist as views to the production schema. Refer to Multiple Owners In A Single Database for more information about table ownership.

The conversion effort involves several steps as illustrated in the following diagram:

Copyright © 2026, Oracle and/or its affiliates.



The following points briefly outline each of the above tasks:

- **Load Legacy Data.** During this step, your legacy data is loaded into the system. Notice that you are not migrating this data directly into production. Rather, your rows are loaded into tables that are identical to the production tables; they just have a different owner. Mapping legacy data into the system is probably the most challenging part of the conversion process because the system is a normalized database (and most legacy applications are not).
- **Validate Staging.** During this step, the system validates the data you loaded into the staging tables. Note that validation rules validate your staging data using the control tables that have been set up in production.
- **Assign Production Keys.** During this step, the system allocates random, clustered keys to the rows in the staging database.
- **XML Resolution.** During this step, the system resolves legacy keys that may be mapped to an XML storage field with new values that were assigned in the key assignment step.
- **Insert Rows Into Production.** During this step, the system populates your production tables with rows from the staging. When the rows are inserted, their prime keys and foreign keys are reassigned using the data populated in the key assignment and XML resolution steps.
- **Validate Production.** During this step, you rerun the object validation processes, but this time against production. We recommend performing this step to confirm that the insertion programs have executed successfully.

> **Note**
>
> Additional steps may be needed to support specific data conversion requirements. Refer to your product documentation for additional conversion related information.

This section further discusses concepts and guidelines related to the conversion process.

Copyright © 2026, Oracle and/or its affiliates.

- Conversion Entities
- Conversion Steps
- A Note About Keys
- Multiple Owners In A Single Database

Conversion Entities

Conversion is enabled for selected master and transaction maintenance objects. In the same way, only a subset of a maintenance object's tables are enabled for conversion.

The following is provided for each maintenance object that supports conversion:

- An entity validation batch control. This batch process validates data in staging and can also be run to validate data in production.
- A key assignment batch control for each **table** that is eligible for conversion and has a system-generated key.
- An XML resolution batch control if the maintenance object has at least one table that includes an XML storage field.
- An insertion to production batch control for every **table** of the maintenance object that is eligible for conversion.

You may review the conversion configuration provided for an entity and more using the Conversion Entity Dashboard portal.

Conversion Steps

The following sections provide more details about the steps in the conversion process.

> **Note**
>
> Additional steps may be needed to support specific data conversion requirements. Refer to your product's documentation for additional conversion related information.

- Load Legacy Data Into Staging Tables
- Validate Information In The Staging Tables
- Allocate Production Keys
- XML Resolution
- Insert Production Data
- Validate Production

Load Legacy Data Into Staging Tables

This section provides some high level discussion about mapping and loading legacy data to the system's staging tables.

Copyright © 2026, Oracle and/or its affiliates.

Mass Load Utility

Unless in a Cloud installation, you can use any method you prefer to load data from your legacy application. However, we recommend that you investigate your database's mass load utility (as opposed to using insert statements) as the mechanism to load the staging tables. In addition, we strongly recommend that you disable the indexes on these tables before populating these tables and then enable the indexes after populating these tables.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

Populating Keys

Some tables in the staging database have keys that are system-assigned random numbers. Special consideration should be given when populating these keys. Refer to A Note About Keys for more information on how to properly populate these tables. Those tables that don't have system-assigned random numbers have keys that are a concatenation of the parent's prime-key plus one or more additional fields.

Populating Characteristic Tables

There are many maintenance objects that include a characteristic table used to capture miscellaneous information about the object. Most of these tables include an indexed column used when searching by characteristic value called Search Characteristic Value. During conversion and depending on the type of characteristic, this column must be populated as follows:

- **Predefined.** Populate search characteristic value with the contents of the characteristic value column converted to upper case.
- **Ad hoc.** Populate search characteristic value with the first 50 bytes of the ad hoc characteristic value column converted to upper case.
- **Foreign key.** Populate search characteristic value by concatenating the values of each foreign key characteristic value column to a maximum of 50 bytes.

Validate Information In The Staging Tables

During the first validation step, the system validates the data you loaded into the staging tables. This section provides some high level discussion about the validation process.

A Batch Process Per Maintenance Object

Each of the maintenance objects that are eligible for conversion must be validated using its respective entity validation conversion batch process. These are classic background processes that can also be run against production data.

These processes are multi-threaded and have no dependency on other processes.

Validation Mode

By default, each entity is fully validated for schema as well as entity validation rules, at the maintenance object and business object level. The batch process performs the same business

Copyright © 2026, Oracle and/or its affiliates.

rules that are used to validate data added by users in your production system. You may restrict the batch process to perform schema validation only or entity validation rules only using a batch parameter. This may be useful in identifying common data issues in preliminary runs.

The batch process sets an indication available to validation rules, that they are being called as part of a conversion process. A custom validation algorithm may use this indication to skip unnecessary rules that are not applicable as part of conversion.

> **Note**
>
> The indication is accessible via the scripting $$isConvGenericValidation context variable.

Random Sample Mode

By default all records are selected for validation. The "Process Every Nth" batch parameter allows you execute this process in a random-sample mode to highlight pervasive errors. In this mode, you are actually telling the batch process to validate every N records.

We strongly recommend validating each entity in the following steps:

- Execute the validation batch process in random-sample mode.
- Review and correct the errors. Note, you can use the base package's transactions to correct an error if the error isn't so egregious that it prevents the object from being displayed on the browser.
- After all pervasive errors have been corrected; re-execute each object's validation batch process in all-instances mode to highlight elusive, one-off errors.

Reported Errors

Errors encountered during validation are logged onto the conversion Validation Error (CI_VAL_ERR) table. Note that at the start of this job, all rows in the conversion error table for the process maintenance object are deleted.

You can view errors highlighted by the validation process using the Conversion Entity Dashboard portal.

Uneven Thread Distribution Using Legacy Keys

By default, the 'Object Validation' batch process distributes thread processing load using the **Thread Level SQL Select** (THDS) batch strategy. This method relies on the entity's prime key to be evenly distributed.

In the staging schema, entities still reference their legacy prime keys which may not be evenly distributed. When the distribution skew is significant, the performance of this batch process may be impacted. In this situation, you may want to use the **Key Ranges** (KEYS) batch strategy instead. Refer to Parallel Background Processesfor more information.

To do so, add the **Batch Strategy** common batch parameter to the corresponding 'Object Validation' batch control record and set it to the 'KEYS' value.

Recommendations To Speed Up Validation

The following points describe ways to accelerate the execution of the validation process:

- Ensure that statistics are recalculated after data has been inserted into the staging tables.

Copyright © 2026, Oracle and/or its affiliates.

- Execute the process multi threaded.
- Execute shorter running validation processes (e.g., less records) first so that the error data can be analyzed while other processes are busy running.
- Remember that validation can be run in random-sample mode. We recommend running these batch processes using a large sample value for this parameter until the pervasive problems have been rectified.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

Another use for these programs

In addition to validating your objects after conversion or an upgrade, the validation programs have another use. Say for example, you want to experiment with changing the validation of a business entity and you want to determine the impact of this new validation on your existing records. You could change the validation and then run that entity's validation batch process- it will produce errors for each record that fails the new validation.

Allocate Production Keys

During the key assignment step, the system assigns a new key for each legacy key for a table that its prime-key requires a system-generated random key. The conversion process allocates new prime keys to take advantage of the system's parallel processing and data-clustering techniques in the production system (these techniques are dependent on randomly assigned, clustered keys).

The topics in this section provides some high level discussion about the key assignment process and describe the background processes used to assign production keys to the staging data.

The Old Key / New Key Table

It's important to understand that the system does not overwrite the prime-keys on the rows in the staging database, as this is a very expensive IO transaction. Rather, a series of tables exist that hold each row's old key and the new key that will be assigned to it when the row is transferred into the production database. We refer to these tables as the "old key / new key" tables.

The convention "<1st letter of owner flag>K_<table_name>" is used to denote the old key / new key table name. For example, the old key / new key table for CI_ACCT is called CK_ACCT.

The insertion batch process that transfers the rows into the production database use the new key for the main record of the key along with any other record where this key is a foreign key. In the same way, the XML resolution process resolves conversion foreign keys residing in XML storage fields and replaces them with their corresponding new keys from these table.

A Batch Process Per Table

A key assignment batch process is provided for each table that has a system-generated key and belongs to a maintenance objects that is eligible for conversion. The batch process is responsible for populating the corresponding old key / new key table (in other words, you don't

Copyright © 2026, Oracle and/or its affiliates.

have to populate these tables). These processes are single threaded. Refer to a table's corresponding key assignment batch process for more information.

Key Assignment Dependencies

Most tables with system-generated keys do not inherit part of their key from another table's key. Their corresponding key assignment batch process have no dependencies and can therefore be executed in any order you please.

Some tables inherit part of their key from another table's key. Key assignment batch processes for such tables must be executed in key inheritance order. In other words, the key assignment process for a table should be run after the process that generates keys for the entity it depend on. The Conversion Entity Dashboard portal displays key inheritance dependency information if any exist for a conversion entity.

> **Note**
>
> You may run multiple key assignment batch process in parallel as long as they are independent with respect to key inheritance.

Iterative Conversions

Rather than perform a "big bang" conversion (one where all records of an entity are populated at once), some implementations have the opportunity to go live on subsets of their entity base. If this describes your implementation, please be aware that the system takes into account the existing prime keys in the production database before it allocates a new key value. This means when you convert the next subset of customers, you can be assured of getting clean and unique keys.

Key assignment logic creates the initial values of keys by a manipulation of the sequential row number, starting from 1. After any conversion run, a subsequent conversion run will start with that row number again at 1, and the possibility of duplicate keys being assigned will be higher. The key assignment batch process allows you to specify a starting value for that row number. The purpose of this parameter is to increase the value of row number by the given value, and minimize the chance of duplicate key assignment. This parameter is only used if you are performing conversions where data already exists in the tables in the production database.

Run Type

Key assignment is performed in two steps:

- **Initial Key Generation.** The system allocates new keys to the rows in the staging tables (i.e., it populates the respective old key / new key table).
- **Duplicate Key Resolution.** The system reassigns keys that are duplicates compared to production.

By default, both steps are performed at the same run but you have the option to run them separately by indicating which step to run via a batch parameter. The proper use of this parameter will greatly speed up the key assignment step as described under **Recommendations To Speed Up Key Generation** section.

Recommendations To Speed Up Key Generation

The following points describe ways to accelerate the execution of the key generation programs.

Copyright © 2026, Oracle and/or its affiliates.

- For a non-Cloud installation:
  - Make the size of your rollback segments large. The exact size is dependent on the number of rows involved in your conversion. Our research has shown that processing 7 million rows generates roughly 3GB of rollback information.
    * Setup the rollback segment(s) to about 10GB with auto extend to a maximum size of 20GB to determine the high water mark
    * A next extent value on the order of 100M should be used.
  - Make sure to turn off all small rollback segments (otherwise Oracle will use them rather than the large rollback segments described above).
  - After the key assignment programs execute, you can reclaim the space by:
    * Keep a low value for the "minimum extent" parameter for the rollback.
    * Shrink the rollback segments and the underlying data files at the end of the large batch jobs.
- Compute statistics on the old key / new key tables after every 50% increase in table size. Key generation is performed in tiers or steps because of the inheritance dependency between some tables and their keys. Although key generation for the inheritance dependency tier currently being processed is performed by means of set-based SQL, computation of statistics between tiers will allow the database to compute the optimum access path to the keys being inherited from the previous tier's generation run.
- Optimal use of the **Run Type** batch parameter.
  - Before any key assignments, alter both the "old key" F1_CX_ID index and the "new key" CI_ID index on the old key / new key tables to be unusable.
  - Run all key assignment batch processes in tiers, submitting each job to only perform the **Initial Key Generation** step.
  - Rebuild the indexes on the old key / new key tables. Rebuilding the indexes using both the PARALLEL and NOLOGGING parameters will speed the index creation process. Statistics should be computed for these indexes.
  - Run all key assignment batch processes in the current tier that were previously run in initial key generation mode, to perform the **Duplicate Key Resolution** step. This will reassign all duplicate keys.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

XML Resolution

Most system-generated foreign keys are stored in physical fields and characteristic tables and as such their legacy value is replaced with its corresponding new key as part of the insertion to production process. Rarely, maintenance objects may store system-generated foreign keys in an XML storage field, in other words, a field defined with the data type of CLOB or XML. This step is only applicable to such maintenance objects.

Copyright © 2026, Oracle and/or its affiliates.

During this step, the system resolves convertible system-generated foreign keys that may reside in XML storage fields you may have loaded into the staging tables. This section provides some high level discussion about the XML resolution process.

The XML Resolution Table

It's important to understand that the system does not overwrite the prime-keys on the XML storage fields in the staging database, as this is a very expensive IO transaction. Rather, a corresponding XML resolution table exists for each table that defines an XML resolution field to capture each row's resolved XML storage content, in other words, the content with all the old keys replaced with the new assigned keys.

The convention "<1st letter of owner flag>R_<table_name>" is used to denote the XML resolution table name.

The insertion batch process that transfers the rows into the production database replaces each XML storage field with its resolved value from the corresponding XML resolution table.

A Batch Process Per Maintenance Object

Each of the maintenance objects that are eligible for conversion and support XML storage fields is provided with an XML resolution batch process. These batch processes must be run to resolves foreign keys that may reside in these XML storage fields.

These processes are multi-threaded and must be executed after the key assignment step has completed and before inserting data to production.

XML Resolution Eligibility

Not all maintenance objects that support XML storage fields actually store convertible system-generated foreign keys in their XML storage field. If none of the business objects associated with the maintenance object involve mapping of such foreign keys to an XML storage field then XML resolution is not needed for any row in the maintenance object. The XML resolution batch process detects such situation and completes right away without storing any rows in any of the maintenance object's XML resolution tables.

> **Note**
>
> If the XML resolution batch process detects that the maintenance object is eligible, the maintenance object must reference a physical BO. If a product delivered business object supports system-generated foreign keys in an XML storage field, the product should be supplying a physical BO. If your implementation has system-generated foreign keys in an XML storage field in a custom business object for a maintenance object where the product has not supplied a physical BO, you must create a physical BO and link it to the maintenance object.

Only Resolved Values Are Captured

XML storage fields typically store large amounts of data. To avoid capturing the same XML content redundantly, the system only stores values in the resolved XML storage fields if the resolved value is different than the original value, in other words, if at least one key was resolved.

If for a given record the resolved XML content is the same as the original content then the following rules apply:

- If this is the primary table of the maintenance object a record is inserted into its corresponding XML resolution table for that record with no value in the XML storage field.

Copyright © 2026, Oracle and/or its affiliates.

- If this is a child table of the maintenance object then no record is inserted into the corresponding XML resolution for that record.

Reported Errors

Errors encountered during XML resolution are logged onto the conversion Validation Error (CI_VAL_ERR) table. Note that at the start of this job, all rows in the conversion error table for the process maintenance object are deleted.

You can view errors highlighted by the XML resolution process using the **Validation Error Summary** page.

Uneven Thread Distribution Using Legacy Keys

By default, the 'XML resolution' batch process distributes thread processing load using the **Thread Level SQL Select** (THDS) batch strategy. This method relies on the entity's prime key to be evenly distributed.

In the staging schema, entities still reference their legacy prime keys which may not be evenly distributed. When the distribution skew is significant, the performance of this batch process may be impacted. In this situation, you may want to use the **Key Ranges** (KEYS) batch strategy instead. Refer to Parallel Background Processesfor more information.

To do so, add the **Batch Strategy** common batch parameter to the corresponding ' XML resolution' batch control record and set it to the 'KEYS' value.

Insert Production Data

The topics in this section describe the background processes used to populate the production database with the information in the staging database.

A Batch Process Per Table

An insert to production batch process is provided for each table of a maintenance objects that is eligible for conversion.

The batch process is responsible for transferring to production all the rows from the staging table while replacing all references to old keys with their corresponding new keys as follows:

- Keys residing in prime keys, foreign key fields and characteristics are resolved using the corresponding old key / new key tables.

  > **Important**
  >
  > If metadata constraints are missing, then FK references for physical fields may be missed. It is important to review each conversion table that it has all the necessary constraints defined.

- Keys residing in generic foreign key reference tables where each row represents a reference to an entity's maintenance object and prime key values are also resolved using the corresponding old key / new key tables. Note that all entities referenced this way must either be convertible or view of production. A reference to an entity that is explicitly marked as not convertible is not supported as it cannot be resolved.
- Keys residing in XML storage fields are resolved using the corresponding XML resolution tables.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Foreign keys residing in any other form of storage are not resolved. For example, any references stored in raw elements, log records, or any other field that is not used as an XML storage field.

All insertion batch processes are independent and may run concurrently. Also note, all insertion batch processes can be run in many parallel threads as described in the next section (in order to speed execution).

Recommendations To Speed Up Insertion

The following points describe ways to accelerate the execution of the insertion batch processes:

- Before running the first insertion batch process:
  - Rebuild the index on the prime key on the old key / new key table.
  - Re-analyze the statistics on the old key / new key table.
  - Alter all indexes on the production tables being inserted into to be unusable.
- After the insertion programs have populated production data, rebuild the indexes and compute statistics for these tables.

> **Note**
>
> In a Cloud installation, refer to the **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

Validate Production

During this step, you rerun the object validation batch processes, but this time in production. We recommend rerunning these batch processes to confirm that the insertion batch processes have executed successfully. We recommend running these batches in random sample mode (e.g., validate every 1000th object) rather than conducting a full validation in order to save time. However, if you have time, you should run these processes in full validation mode (to validate every object).

A Note About Keys

The prime keys of the tables in the staging database are either system-assigned random numbers or they aren't. Those tables that don't have system-assigned random numbers have keys that are a concatenation of the parent's prime-key plus one or more additional fields. Every table whose prime key is a system-assigned random number has a related table that manages its keys; we refer to these secondary tables as "key tables".

The following points provide more information about the key tables:

- Key tables are used by programs that allocate new keys. For example, before a new key is allocated, the key assignment program checks the corresponding key table to see if it exists.

Copyright © 2026, Oracle and/or its affiliates.

- Key tables only have two columns:
  - The key of the object.
  - An environment ID. The environment ID identifies the database in which the object resides.
- Key tables are named the same as their primary table with a suffix of "_K". For example: The key table for CI_ACCT is CI_ACCT_K.
- The key table for a table whose prime key is a system-assigned is defined on its table definition record.
- When you populate rows in tables with system-assigned keys, you must also populate a row in the related key table. For example, if you insert a row into CI_ACCT, you must also insert a row into CI_ACCT_K. The environment ID of these rows must be the same as the environment ID on this database's installation record.
- When you populate rows in tables that reference this record as a foreign key, you must use the appropriate key to ensure the proper data relationships. For example, if you insert a row in CI_SA for the above account, the ACCT_ID column must contain the temporary account key.
- When you insert rows into your staging database, the keys do not have to be random, system-assigned numbers. They just have to be unique. A later process, Allocate Production Keys, will allocate random, system-assigned keys prior to production being populated.

Multiple Owners In A Single Database

The conversion process relies on the following table owner configuration in the system database:

- The production owner is linked to the tables used by your production system. These tables have an owner ID of CISADM.
- The staging owner is linked to the tables into which you insert your pre-validated data. These tables have an owner ID of CISSTG.

The staging owner schema is almost identical to the production schema, with the following exceptions:

- Control tables are not actual tables in staging but rather views to the corresponding tables in production.
- Conversion specific tables designed solely to support the conversion process exist only in the staging schema. For example, tables that manage key assignment and XML resolution.

This section provides some high level concepts related to these table owners.

Validation Always Uses Production Control Data

When the validation batch processes run against your staging data, they validate the staging data against the production control tables (and insert errors into the staging error table). This means that the SQL statements that access / update entities needs to use the staging owner whereas the SQL statements that access control tables need to use the production owner. But notice that when these same validation batch processes run against production, the SQL statements will never access the staging owner. Rather, they all point at the production owner.

This is accomplished as follows:

Copyright © 2026, Oracle and/or its affiliates.

- A separate application server must exist for each owner. Each application server points at a specific database user ID.

> **Note**
>
> In a Cloud installation, the application server may only point to a specific database user ID at any point in time. Refer to **Data Conversion Support for Cloud Implementations** for more information about switching owners.

- The database user ID associated with the staging owner uses CISSTG as the owner for the master and transaction tables, but it uses CISADM as the owner of the production control tables.
- The database user ID associated with the production owner uses CISADM as the owner for all tables.

You may wonder why we went to this trouble. There are several reasons:

- We wanted to reuse the validation logic that exists in the programs that validate your production data. In order to achieve this, these programs must sometimes point at the staging owner, and other times they must point at the production owner (and this must be transparent to the programs otherwise two sets of SQL would be necessary).
- We wanted to let you use the application to look at and correct staging data. This can be accomplished by creating an application server that points at your staging database with the ownership characteristics described above.
- We wanted the validation programs to be able to validate your production data (in addition to your staging data). Why would you want to validate production data if only clean data can be added to production? Consider the following scenarios:
  - After an upgrade, you might want to validate your production data to ensure your pre-existing user-exit logic still works.
  - You may want to conduct an experiment of the ramifications of changing your validation logic. To do this, you could make a temporary change to user exit logic (in production) and then run the validation jobs on a random sample basis.
  - You forget to run a validation program before populating production and you want to see the damage. If you follow the instructions in this document, this should never happen. However, accidents happen. And if they do, at least there's a way to determine the ramifications.

Only Validation Can Work In Both Owners

While the redirection of owner ID's is a useful technique for the validation batch processes, it cannot be used by the key assignment and production insert batch processes? Why, because these processes have to access the same tables but with different owners at the same time. They also need to reference conversion specific tables that do not exist in production. For example, the batch process that inserts rows into a table in production must select rows from the staging version of that table, resolve keys from the conversion old key / new key mapping tables and insert the resolved records into the production version of that same table

This is accomplished as follows:

- In staging, a view to production exists for each eligible to conversion table. These views have hard-coded the database owner to point to production. For example, there is a view called CX_PER that points at person table in production.

Copyright © 2026, Oracle and/or its affiliates.

- The key assignment and insertion programs use these views whenever then need to access production data.

Conversion Entity Dashboard

This portal provides an overview of a maintenance object's conversion configuration along with information related to its current status and progress within the overall conversion process.

Refer to Understanding Conversion for more information.

You can access the portal from the **Admin Menu**, and then **Implementation Tools**, and then **Conversion Entity Query.** You are brought to a query portal with options for searching for a specific maintenance object that is eligible for conversion. Once a maintenance object has been selected you are brought to the conversion entity dashboard portal to review the information for the selected record.

The following zones may appear as part of the portal's **Main** tab page.

- **Tables.** This zone provides information about tables that belong to the maintenance object and are marked as eligible for conversion.
- **Batch Controls.** This zone lists the conversion batch controls defined for the maintenance object in the order they should be executed within the conversion process along with current execution details if any.
- **Error Summary.** This zone provides a summary of all conversion errors related to the maintenance object.
- **Duplicate Keys.** This zone lists duplicate legacy keys that need to be resolved by the key assignment conversion process.

The **Errors** tab page provides a query zone that allows you to review conversion errors related to the maintenance object by various criteria.

Data Export

This section describes concepts and tools related to the various types of data export methods supported by the product.

- Understanding Data Export
- Defining Data Export Control
- Generalized Export Dashboard

Understanding Data Export

The application supports the following classes of export methods for a maintenance object:

- **Generalized.** This method may be used for maintenance objects that are low to medium in volume and update activity. Most maintenance objects are eligible for this method.
- **Specialized.** This method should be used for maintenance objects that are high in volume or very frequently updated.

Copyright © 2026, Oracle and/or its affiliates.

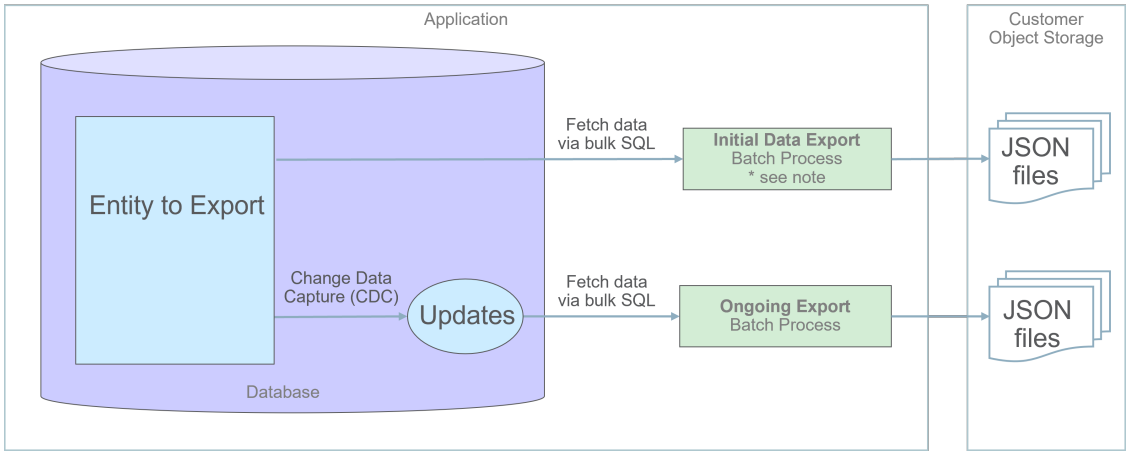
> **Note**
>
> It is your organization's responsibility for designing the process by which these files are imported to one or more target environments.

This section discusses concepts and guidelines related to data export features.

- Generalized Data Export
- Specialized Data Export

Generalized Data Export

The generalized data export method provides a file-based export of data in JSON format. Exporting the data of a maintenance object using this method involves an initial export of the entity's entire data followed by an ongoing export of changes as they are made over time. These processes are illustrated in the following diagram:



The overall process involves the following steps:

- Enable entities for export in the Generalized Export Dashboard portal. You may enable many entities at once. This step also allows you to enable ongoing export of these entities.
- Submit the Generalized Initial Export Initiator (`F1-GEXPI`) batch process to export the current content of all enabled entities. This step submits a separate batch process for each entity you have enabled for export.

  > **Note**
  >
  > A full database export can be used instead. The export file can be imported in the target environment to establish the initial baseline. Cloud customers may contact cloud operations to request such export for on-premise use. This schema export is limited for reporting purposes and is intended for initial extract only.

- Schedule the Generalized Ongoing Export (`F1-GEEXO`) batch process to run periodically.

The following sections further discusses concepts and guidelines related to the generalized data export method.

Copyright © 2026, Oracle and/or its affiliates.

Eligibility

While most entities are eligible for generalized data export, not all of them are. Here are a few of the common reasons for that:

- The entity manages extremely high volume of data. Tracking changes to such entity would double the volume of managed data and impact overall performance.
- The entity is updated very frequently. Tracking these events may impact overall performance.
- The entity is used by the product's infrastructure and operational processes. These entities are maintained differently and cannot be subject to the generalized data export methods.

By default a maintenance object is eligible for the generalized data export method unless explicitly marked otherwise by the **Data Export Class** option. The option may be used to mark an entity as either not allowed for any type of export or allowed for a specialized export only. The lack of such explicit option means the maintenance object is eligible for the generalized export.

> ⚠ **Caution**
>
> If an entity is explicitly marked as not eligible for generalized data export there is a good reason for that and its configuration should not be changed.

One Method Fits All

The generalized data export method is designed to accommodate many eligible maintenance objects in a generic way that must also takes performance considerations into account. For this reason this method does not support entity specific filtering options nor custom rules of any kind.

The following rules apply to all entities that may be exported using this method:

- Entire data is exported.
  - All fields on all tables that belong to the maintenance object are exported except for "key" and "log" tables if any.
  - Maintenance object log tables typically do not contain useful business information and as such are omitted from export for performance and data volume reasons. However, there are entities for which log records provide analytical value. When applicable, use the **Export Log Tables** option to explicitly include log tables for a specific maintenance object.
- Same data structure and format is used for all entities. Refer to the **Record Format** section for more information.
- Data is exported to files only. Refer to the **Export To Files Only** section for more information.
- All changes are captured. It is not possible to exclude some changes from being tracked for ongoing export based on custom business rules.
- When data changes, a full snapshot of the data related to that entity is exported to avoid heavy data merging on the target side.
- When the entity is deleted, only the primary key of the entity is exported along with an indication that it is deleted.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Export is not supported for entities that store fields in a time zone other than the base time zone. This type of definition is not commonly used and includes maintenance objects that at least one of their fields is defined as stored in **Reference Time Zone**.

Export To Files Only

The immediate destination of both initial and ongoing export processes are files in a customer specified location. In a Cloud installation files are created in a customer owned Object Storage.

Based on your business requirements you may further consume these files by downstream applications such as data lakes etc.

Enabling Data Export

To enable data export for an eligible maintenance object a **Data Export Control** record needs to be created. The record keeps an indication of whether the initial export for the maintenance object has been completed or not and whether changes made to the entity should be tracked and exported on an ongoing basis. A designated portal allows you to enable the generalized data export for many maintenance objects at once as well as monitor their current export status at a glance. Refer to Generalized Export Dashboard portal for more information.

> **Caution**
>
> Enabling and disabling entities for export should only be made via the Generalized Export Dashboard or the Data Export Control portals to ensure the configuration is valid and complete.

> **Caution**
>
> When applicable, ongoing data export should be enabled before submitting the initial export batch process for the entity. This ensures that changes made during the initial export are properly tracked.

Ongoing export options are cached in various online and batch caches. In order for these options to take effect and track entity changes made by any online or batch process, the various caches must be properly flushed when these options are changed. Using the designated portals to enable entities for export triggers a 'global' flush of the caches. If the batch thread pool workers are configured to refresh their caches when a global flush is requested, then this is the only step required. If not, then the `F1-FLUSH` batch job should also be submitted to refresh the caches used in batch processing. Refer to Caching Overview for more information.

Initial Data Export

Each maintenance object that is eligible for the generalized data export method is associated with an initial data export batch control that is referenced on the maintenance object using the **Export Batch Control** option. The same common batch program is used by all these batch controls to export the entire data of an entity to files. The process is multi-threaded and produces one file per thread by default. An optional batch parameter allows for multiple smaller files to be generated per thread by setting a limit on the number of records written to each file. Refer to the **File Size** section for more information.

Copyright © 2026, Oracle and/or its affiliates.

Instead of manually submitting the generalized initial export batch process for each maintenance object you have enabled for export, submit the F1-GEXPI Generalized Initial Export Initiator batch process to automatically submit them all at once.

The initial export process for each maintenance object updates the corresponding data export control record for the maintenance object to indicate that the process has started and updates it again upon completion. This allows you to view the overall status of the initial export across all enabled entities on the Generalized Export Dashboard portal.

Note that the initiator batch process only submits batch processes for enabled maintenance objects that according to their data export control record their initial export has not started yet. If you need to export one or more entities again, reset the initial export status indication on their data export control records and submit the initiator batch process again.

The initial export file name is constructed as follows:

```
INIT_EXPORT[_file prefix(optional)]_[maintenance object]_[batch
run]_[thread]_[thread count]_[timestamp][_file sequence].json.gz
```

The "file sequence" part of the file name is used only when more than one file is created per thread. The sequence suffix is populated starting from the second file as follows _2.

Refer to any of the individual generalized initial export batch controls and the initiator batch control for more information.

Restricting Initial Export By Date

By default, the Initial Data Export batch process exports the entire data for an entity. In some situations, typically around high volume historical data, there may be a need to restrict the export to more recent data, for example the last few months of data.

The initial export batch process supports an optional batch parameter **Restrict By Date** that can be used to limit the scope of data to export. The parameter references a date field and a requested time frame in terms of number of days prior to business date. Refer to the F1-GEIXP Generalized Initial Export Template batch control for more information.

> ⚠ **Caution**
>
> Excluding records from export should be carefully considered as this may result in data discrepancies when the excluded data is referenced on other exported entities.

> ⓘ **Note**
>
> The option is only applicable to specific entities that have a qualifying reference date and the time frame for export may vary for each customer. Therefore, the new parameter is not added to any base product Initial Export batch control aside of the template batch control for documentation purposes. You may clone the base product batch control for a specific entity and add the parameter as per your business needs.

Ongoing Data Export

Data changes are tracked by the Capture Change for Ongoing Export (F1-MO-REGCHN) Maintenance Object Audit algorithm. The algorithm records the prime key of a changed entity

Copyright © 2026, Oracle and/or its affiliates.

in a designated **Data Export Update** table that serves as a queue of changed entities to export. The queue is later on consumed by the ongoing data export batch process.

> **Note**
>
> The Maintenance Object Audit plug-in spot is capable of tracking changes made to the entity by any layer of the application. Making changes using direct SQL would bypass this mechanism and therefore should be avoided.

The audit rule is added to a maintenance object when the ongoing data export option on its corresponding data export control record is enabled. In the same way the audit rule is removed from the maintenance object when the ongoing export option is disabled.

> **Note**
>
> The ongoing export option has to be enabled on the data export control record for the entity before it is initially exported. This ensures changes made while the initial export process is in progress are not missed.

Unlike the initial export batch process that requires a batch control to be defined for each eligible maintenance object, a single batch process is used to export ongoing changes made to all entities.

The following describes the ongoing export batch process at a high level:

- All changes queued in the Data Export Update table are exported to a designated ongoing export file. Refer to the **Managing a Large Backlog** section for more information.
- A full snapshot of the data related to the changed entity is exported to avoid heavy data merging on the target side.
- Once exported, records are deleted from the Data Export Update table.

The process is multi-threaded and a separate file is produced for each thread by default. An optional batch parameter allows for multiple smaller files to be generated per thread by setting a limit on the number of records written to each file. Refer to the **File Size** section for more information.

> **Note**
>
> It is important to schedule the ongoing export batch process to run frequently enough to avoid accumulating a high volume of records in the Data Export Update table. The smaller the table is kept the faster it is for the change data capture mechanism to add records to it.

The ongoing export file name is constructed as follows:

```
INC_EXPORT_[batch run]_[thread]_[thread count]_[timestamp][_file
sequence].json.gz
```

The "file sequence" part of the file name is used only when more than one file is created per thread. The sequence suffix is populated starting from the second file as follows _2.

Copyright © 2026, Oracle and/or its affiliates.

Refer to Generalized Ongoing Export (F1-GEEX0) batch control for more information.

File Size

The initial export batch process produces a single file per thread by default. For some high volume entities this may result in files that are too large to manage. To produce multiple and smaller files for a specific entity, limit the number of records written to each file by setting the corresponding parameter on the initial export batch control defined for the entity. The same parameter may be provided when submitting the Initial Export Initiator batch process, in which case the initiator value is used only when a corresponding value has not been defined on the entity's specific batch control record.

In the same way, the ongoing export batch process produces a single file per thread by default. To produce multiple and smaller files, limit the number of records written to each file by setting the corresponding parameter on the batch control.

To avoid the creation of too many files by a single batch run, the system sets a maximum limit of around 500 files per run. When the requested number of records per file is determined to cause the number of generated files to exceed the system limit, the actual value used is adjusted to meet the system limit.

Managing a Large Backlog

When a large backlog of changes accumulates in the ongoing export queue for some reason, the next batch takes longer to export which delays the downstream import step. A better option is for the backlog to be cleared in smaller "chunks", allowing the downstream process to import them as they are made available.

You may limit the number of records the batch process exports using the **Thread Processing Limit** batch parameter. When specified, the number of records exported by each thread is capped by the specified limit, leaving the remaining records to be consumed by subsequent batch runs. Instead of waiting for the next scheduled batch run, a new batch process may be automatically submitted, if explicitly requested to do so using the **Automatic Backlog Clearance** batch parameter.

It is important to ensure that the backlog clears gradually and that the queue resumes its normal size in a timely manner. The processing limit must therefore be set with an appropriate value that would promote proper clearance of the queue. To do so, at least 20% of the queue, and no less than 100,000 records, must be consumed by each batch run across all threads. If the specified limit does not meet the minimum clearance requirement then the processing limit is adjusted accordingly.

Record Format

Data is fetched and exported directly from the database in bulk read operations for performance reasons. For that reason, the structure used does not reflect the logical data model of the entity but rather its physical list of tables. Data for a given instance is organized by tables followed by rows in each table and for each row its list of fields in JSON format.

The following describes the format and structure used:

```
{
"OBJ": "<mo name>",
"TIMESTAMP": "<export time in ISO format for example
2019-07-25T11:06:04.740615Z>",
"PK1": "<mo pk1 value>",
"PK2": "<mo pk2 value if any>",  ← PK2-5 should only be included when
applicable
"PK3": "<mo pk3 value if any>",
```


Copyright © 2026, Oracle and/or its affiliates.

```
"PK4": "<mo pk4 value if any>",
"PK5": "<mo pk5 value if any>",
"DELETED": true,  ← should only be included when the entity is deleted
"DATA":
{
"<MO table name 1>":
[
{<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>},...
],
"<MO table name 2>":
[
{<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>}
],...
 "<MO table name n>":
[
 {<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>}
]
}
```

Notes about field values and formats:

- All fields for a row are included even when the field is empty or null.
- All string values are trimmed of excess trailing spaces but if a field is empty it exported as a single space and not null.
- An empty array is used for a table node when no records exists for the entity in the table.
- Date/time information is converted to UTC time zone and exported in ISO format for example 2019-07-25T11:06:04.740615Z.
- Deletes have an explicit indicator in the header.

Order of Consuming Files

It is important to apply all the initial export files for a maintenance object before applying ongoing change files to ensure data consistency.

The following is a recommended procedure for consuming generalized data export files:

- Apply all initial data export files before ongoing export files.
- Apply initial files for a maintenance object in run number order.
- Apply ongoing export files in run number order.
- If a new initial export is available the file consumption procedure should be changed temporarily as follows:
  - Stop consumption of ongoing export files.
  - Apply the new initial export files.
  - Resume consumption of ongoing export files.

Copyright © 2026, Oracle and/or its affiliates.

Recover from a Lost Ongoing Export File

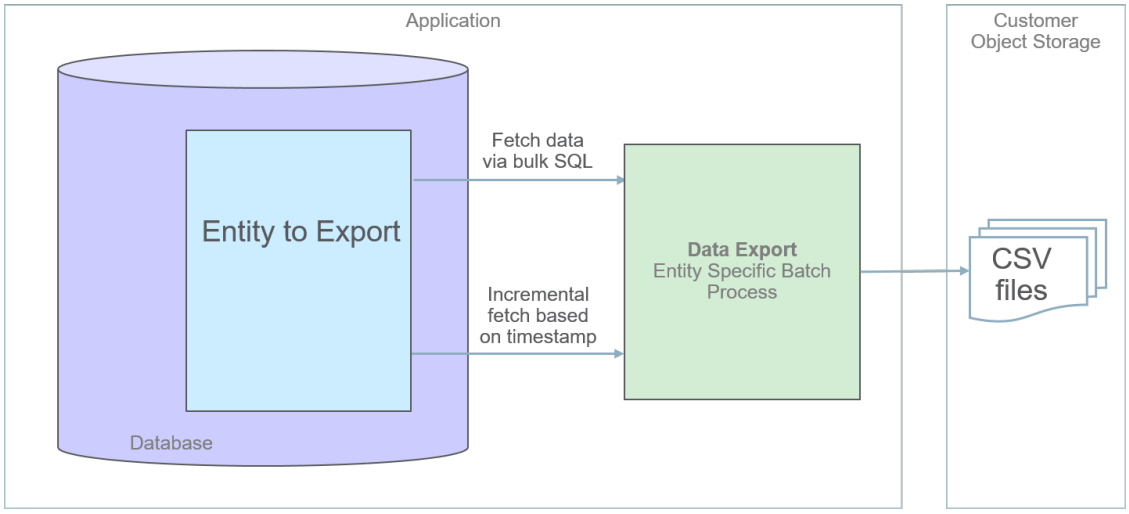
The ongoing export batch process keeps a backup of all entity keys, that were exported on a specific batch run, in a designated backup table. In rare situations where an ongoing export file is deleted or becomes corrupted before it gets processed, the backup table may be used to identify the missing changes and export them again.

Use the Restore Generalized Export Keys (`F1-GERST`) batch process to recover from this situation. The process identifies the entities that were exported during a specified batch run and adds them back to the ongoing changes queue from the backup table. The entities would then be exported again wen the next ongoing export runs. This method does not restore the exact content of the original file that was lost as it cannot be reproduced. Instead, the process ensures that the entities that were included in the lost file would be exported again.

> **Note**
>
> Backup records follow a weekly retention policy for performance reasons. They are meant to support near future situations of file recovery and are not an historical change data capture log.

Specialized Data Export

Several performance considerations are called for when exporting the data of a high volume entity or an entity that is updated in extremely high frequency. The generalized method does not fit such entities and a specific batch process is designed for each, aiming to achieve an optimal export method for the entity. The initial run of the process exports the entity's existing data while subsequent runs of the same process exports data incrementally, i.e. it exports records that were added or updated since the previous run based a timestamp field. The overall process is illustrated in the following diagram:



The following sections further discusses concepts and guidelines related to the specialized data export method.

Copyright © 2026, Oracle and/or its affiliates.

A Specific Process For Each Entity

A maintenance object that is designed to be exported using the specialized method is explicitly marked as such by the **Data Export Class** option and its specific batch process is defined using the **Export Batch Control** option.

All specialized batch processes are designed to export the data in a CSV format as it produces smaller files in size compared to more verbose formats. Refer to each batch process for more information about specific export considerations and parameters.

Enabling Data Export

To enable data export for a maintenance object a specialized type of Data Export Control record needs to be created. The record serves as an indication that such entity should be exported and keeps track of the date and time through which data has been exported so far, allowing subsequent export run to only export incremental changes.

Incremental Data Export

The specialized export method tracks changes incrementally from prior export run and does not use a change data capture mechanism. It therefore relies on a timestamp field on the primary table of the maintenance object to indicate when data was added or changed. Only entities that were added or changed since the previous export run are exported with each subsequent run. The date and time through which data has been exported is recorded for each maintenance object on its corresponding Data Export Control record.

Order of Consuming Files

Specialized data export files should be consumed in run number order to ensure a consistent and chronological uptake of data.

Defining Data Export Control

This portal is used to maintain Data Export Control records. These records capture your organization's export settings and current export state for a selected maintenance object. For a generalized export it keeps an indication as to whether the initial export for the maintenance object has been completed or not and whether changes made to the maintenance object should be tracked and exported on an ongoing basis. For a specialized data export it keeps the date/time through which data was exported.

Refer to Understanding Data Export for more information.

You can access the portal from the**Admin Menu**, and then **Implementation Tools**, and then **Data Export Control.** You are brought to a query portal with options for searching for a specific record. Once a data export control record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Data Export Control** zone provides information about the selected record.

Generalized Export Dashboard

The number of maintenance objects that are eligible for generalized export is large. This portal allows you to easily select and enable export for many eligible entities at once as well as review the current export state of those already enabled at a glance.

Copyright © 2026, Oracle and/or its affiliates.

Refer to Generalized Data Export for more information about the generalized data export method.

You can access the portal from the **Admin Menu**, and then **Implementation Tools**, and then **Generalized Export Dashboard.**

The following zones may appear as part of the portal's **Main** tab page.

- **Export Entities.** This zone lists entities for which generalized data export control settings have already been set.
- **Eligible Entities.** This zone lists entities that are eligible for the generalized data export method and for which such settings have not yet been set.

Copyright © 2026, Oracle and/or its affiliates.

Part IV
Customer to Meter Administrative User Guide

Welcome to the Oracle Utilities Customer To Meter Administrative User Guide. This document is used with the following products:

- Oracle Utilities Customer to Meter
- Oracle Utilities Customer Cloud Service
- Oracle Utilities Billing Cloud Service

> **Note**
>
> Refer to Defining Oracle Utilities Billing Cloud Service Options for specific information related to Oracle Utilities Billing Cloud Service

- Oracle Utilities Rate Cloud Service

> **Note**
>
> Refer to Defining Oracle Utilities Rate Cloud Service Options for specific information related to Oracle Utilities Rate Cloud Service.

This guide describes administration and configuration features and functionality, including:

- Customer to Meter Overview
- Customer To Meter Configuration
- Defining General Options
- Defining System-Wide Options
- Defining Financial Transaction Options
- Defining Customer Options
- Defining Service Order Options
- Defining Credit & Collections Options
- Defining Asset Options
- Defining Location Options
- Defining Device Options
- Defining Device Installation Options
- Defining Device Measurement Options
- Defining VEE Options
- Defining Usage Options
- Defining Time of Use Options
- Defining Premise and Service Point Options

Copyright © 2026, Oracle and/or its affiliates.

- Defining Bill and Measurement Cycles
- Defining Service Agreement Type (SA Types)
- Defining SA Relationship Options
- Defining Service Credit Options
- Defining Loan Options
- Defining Non-Billed Budget Options
- Defining Quotation Options
- Defining Case Management Options
- Defining Umbrella Agreement Options
- Reports Addendum
- Defining Overdue Processing Options
- Defining Prepaid Metering Options
- Defining Conservation Program Options
- Defining Open Market Options
- Defining Consumer Contract Options
- Configuration
- Integration
- Defining Communications Options
- Data Extracts
- Aggregation
- Consumption Synchronization
- Defining Dashboard Options
- Measurement Reprocessing
- Information Lifecycle Management (ILM)
- Outage Storm Mode
- Running Initial Periodic Estimation
- Defining Customer Program Management Options
- Defining Oracle Utilities Rate Cloud Service Options
- Defining Oracle Utilities Billing Cloud Service Options
- Smart Grid Gateway Adapters
- Defining Settlement Options

Customer to Meter Overview

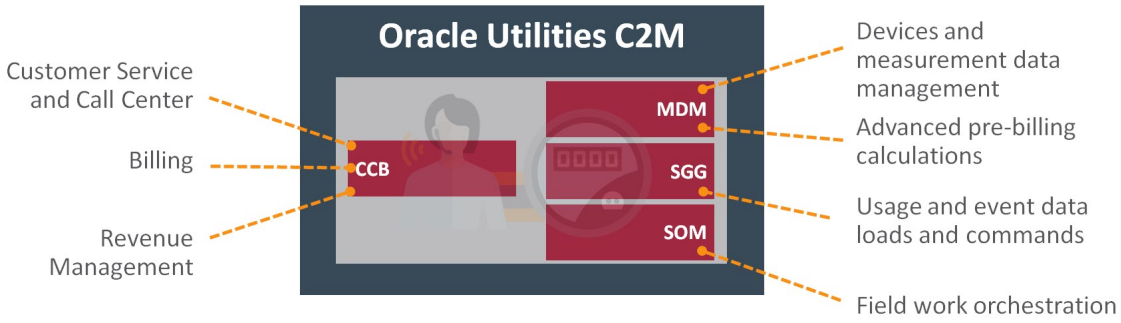
- Customer to Meter Overview
- Best Practices
- Additional Resources

Copyright © 2026, Oracle and/or its affiliates.

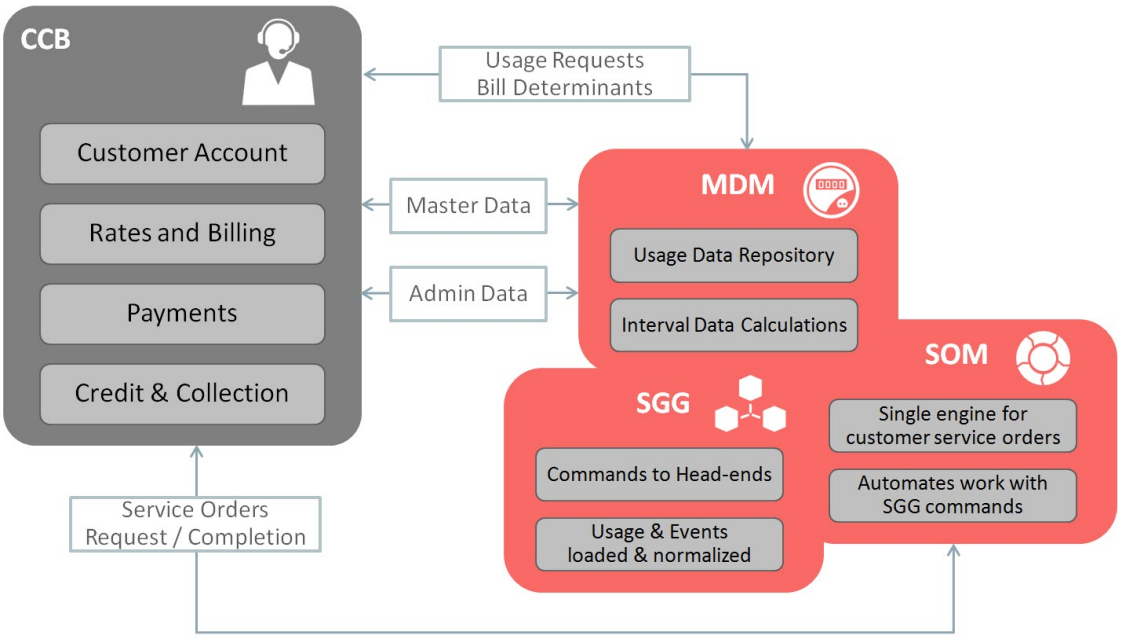
Customer to Meter Overview

Oracle Utilities Customer to Meter is a next generation customer service and billing application that incorporates a modern meter data management system. This system is designed to scale with mass smart meter volumes while handling the complexity associated with a utility's processes across multiple types of service, operating divisions, and jurisdictions. The new application components allow for traditional scalar meter functionality that is equivalent to prior versions of Oracle Utilities Customer Care and Billing; so, it can accommodate the many types of utility environments, and evolve with the business to include more advanced digital requirements.

The roles for each application area in Oracle Utilities Customer to Meter are defined as follows:



The application components have internal integration for master, admin, and transactional data. The primary transactions between them are for bill determinant processing with MDM and service order processing with SOM. These transactional processes can be initiated by the CCB solution, or from other external systems.



In addition, Oracle Utilities Customer To Meter can also include asset management functionality via Oracle Utilities Operational Device Management which provides the ability to handle large volumes of assets and to manage the receipt, installation, maintenance, tracking and removal of those assets.

Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Customer Care and Billing Overview
- Oracle Utilities Meter Data Management Overview
- Oracle Utilities Smart Grid Gateway Overview
- Oracle Utilities Service Order Management Overview
- Oracle Utilities Operational Device Management Overview
- Oracle Utilities Market Settlements Management Overview

Best Practices

- Performance Recommendations
- Referencing Master Data by Identifiers
- Smart Grid Gateway Native Implementations
- Recommendations for Creating an On-Premises Production Environment

Performance Recommendations

This section includes best practices and recommendations regarding performance.

- Partitioning and Data Removal
- Device Channels to Process
- Data Conversion
- Initial Measurement Loading Recommendations
- VEE Recommendations
- Measurement Processing Recommendations
- Aggregation Recommendations
- Usage Transaction Recommendations
- Usage Calculation Recommendations
- User Interface Recommendations
- Exception Handling Recommendations
- SQL Recommendations
- Custom Rules and Processes
- Testing Recommendations
- Java Recommendations — On-Premises Implementation Only

Partitioning and Data Removal

- **Implement Partitioning**: Partitioning is essential to manage the large amounts of data coming into Meter Data Management. Customers will have significant performance problems without partitioning. Partitioning is required for good performance and to organize the data so old data can easily be dropped from the database. Customers need historic partitions (for conversion and replacement reads) and future partitions.

Copyright © 2026, Oracle and/or its affiliates.

- Cloud Implementations: Customers are no longer required to run the "Add Partition" (K1-ILMAD) batch process every month. Oracle automatically adds partitions for entities, every month, on desired domains and schemas. However customer are required to run the "ILM Automation - New Subretention Value" (F1-ILMSV) batch process (for sub-partitions, if needed) on a regular basis to ensure that new partitions are created as needed.
- On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* for partitioning guidelines for on-premises implementations.

- **Implement Compression**: Compression is essential to reduce the size of the database. For standard hardware we recommend Advanced Compression. For engineered systems that use Oracle Exadata, we highly recommend HCC (Hybrid Columnar Compression) this reduces the overall database by approximately 50%.
  - Cloud Implementations: Customers should use the “ILM Automation – Non-Measurement Compression” (K1-ILMNC) and “ILM Automation – Measurement Compression” (K1-ILMMC) batch processes to change the compression level from Advanced to Hybrid Columnar for non-Measurement data at least two months old, which can result in significant disk space savings. Use the following parameters when running this batch process: Owner, MO (optional), DOP (Degree of Parallelism), Estimated Time Limit (How long job can run until it will not compress another partition). In larger databases this can run for several days a month.
  - On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* for compression guidelines for on-premises implementations.
- **Setup Information Lifecycle Management (ILM)**: – The Information Lifecycle Management processes mark items eligible to be removed and update partitions to segregate the data that can be removed. Customers need to define the “ILM Configuration” default retention periods (via the ILM Master Configuration). The retention period should be defined for all ILM objects. ILM Crawlers are provided for low volume transactions. High Volume transactions are marked as eligible for archiving automatically.

  > **Note**
  >
  > Completed IMDs (Initial Measurement Data records) can be removed shortly after being imported. Removing IMDs after 3 months will reduce the compressed database by approximately 50%. There will be database storages savings by archiving other ILM objects as well, such as Activities and To Do Entries.

  Different types of implementations call for different approaches to using ILM.
  - Cloud Implementations: Run the ILM Crawlers Batch Job (batch controls by object) and once the data is aged, run the "ILM Automation - Drop Partition" (K1-ILMDR)_batch process to remove the old partitions from the cloud service database. Please refer to the detailed description of this batch process for additional details.
  - On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* and Information Lifecycle Management for ILM guidelines for on-premises implementations.

Device Channels to Process

Copyright © 2026, Oracle and/or its affiliates.

Oracle does not recommend customer process all channels of data that Meter Data Management can derive.

- Oracle does not recommend utilities hard-code devices with time of use periods (e.g.: on-peak, off-peak). Hard-coded time of use periods will likely require reprogramming in the future when those periods change. Meter Data Management can easily calculate time of use values from interval data.
- Oracle does not recommend customers hard-code demand channels (such as channel used to measure kW), since these must be reset every bill cycle. Demand can be derived from kWh interval data.
- Likewise, Oracle does not recommend customers hard-code channels for Power Factor since that can be derived from kWh and kAVRh interval data.

Data Conversion

- Only convert essential data. Only convert objects that are essential to running the business.
- Pay careful attention to measurement data. Measurement data is very, very large and may take weeks to convert. Consider this: if it takes one hour to load measurements every day, how long will it take to load 5 years of measurements? Although conversion jobs are faster than regular data loading, measurement conversion still takes time. Here are a few measurement best practices:
  - Projects are encouraged to only convert the minimum amount of measurement data required.
  - If historical measurements are required, try to limit the amount of data converted. The best practice is to convert 13 months of consumption to support VEE.
  - Only convert channels absolutely required, usually billing channels, and don't convert other interval data if not needed.
  - Measurement data values should be converted directly into the Measurement table and not use the IMD and VEE processes.
  - **Do not convert Initial Measurement (IMDs)**. Completed IMDs are not needed for any Meter Data Management processes.
  - Remember that billing quantities will likely also be stored in the customer information system, so confirm that the measurement data is required before conversion.

Initial Measurement Loading Recommendations

General Recommendations

- We recommend using Direct Measurement Processing for initial load IMDs, this has a performance boost over legacy IMD initial load processing while also saving considerable storage.
- We recommend using the SGG native implementation payload upload.
  - SGG native usage upload is most efficient when it is processing files of roughly equal size where the number of files is evenly distributed across the number of threads used for processing. For example, if you have 10 threads processing you would want at least 10 roughly equal sized files.

Copyright © 2026, Oracle and/or its affiliates.

        * If your head end system does not produce enough files or the files vary greatly in size then we recommend implementing a process external to MDM to split the files so they are (1) roughly equal in size and (2) evenly distribute across the number of threads used for processing

            * If you are unable to implement an external process to split the files then certain adapters support a chunkSize parameter that can be used to ensure the files are evenly distributed across the threads. See Common Parameters for more information about the `chunkSize` parameter.

- Load initial measurements and perform VEE processing every 4 to 6 hours to optimize performance.
- Initial measurement data (IMD) processing is impacted by a number of factors, including the size of the IMDs being processed, and other concurrent processing (such as other batch processes).
  The size of an IMD is determined by two primary factors, the interval size and the duration of the IMD.
  – The interval size determines how many measurement values are contained in the IMD. For example, an IMD containing 1 days' worth of measurement of hourly intervals contains 24 interval values, while an a IMD containing 1 days' worth of measurement of 15-minute intervals contains 96 interval values, and an IMD containing 1 days' worth of measurement of 5-minute intervals contains 288 interval values. All three contain 1 days' worth of measurements but will require different amounts of compute processing.
  – The second factor impacting the size of an IMD is its duration. However, as noted above the interval size of the IMD plays a significant role in determining the overall size of the IMD. As a general rule, the smaller the interval size, the shorter the duration should be.

  We recommend that you limit the number of measurements in a single IMD. In general we recommend a maximum of 1 days' worth of measurements in a single IMD, but this should be adjusted based on the interval size (again, the smaller the interval size, the shorter the duration should be). We recommend trying different combinations of durations and the number of IMDs (such as 3 IMDs per day each containing 8 hours of intervals, 4 IMDs per day each containing 6 hours of intervals, and so so) to find a volume that works best in your environment.

  One other factor to consider if using legacy measurement processing is the impact on database storage. In general, the number of IMDs has a greater impact on database storage than the number of intervals within each IMD – sending more IMDs per day results in a larger impact on database storage. For example, sending the same number of intervals in a day in a single IMD has a smaller database storage impact than sending the same number of intervals spread across multiple IMDs.

  Mitigating concurrent processing involves optimizing the scheduling of IMD and other processing to best suit the needs of your implementation. The initial measurement data and event loading processes should be scheduled to avoid overlap with other intense processes. For instance, do not produce bill determinants when loading meter readings.
- Initial measurement payloads should have very selective criterion to get to the exact measuring component (MC). In an ideal case, it should be the MC identifier along with device serial number. If this is not provided, then the UOM/TOU/SQI configured on the MC type is used to retrieve the exact business object (BO) from the Service Provider which might have the same values for multiple channels.
  Typically, projects use tools such as SQL Loader to migrate large volumes of historical data directly into the Measurement table rather than use IMDs.
- Avoid loading large numbers of initial measurements for a single device configuration in a single payload.

Copyright © 2026, Oracle and/or its affiliates.

  - D1-IMDV2 helps maintain a constant throughput of IMDs, however, a large number of IMDs for a single device configuration will still cause the thread that picks up that work complete materially later than other threads thus delaying your batch schedule

- Populating the raw data section of the IMD record will reduce overall throughput. Avoid retaining “raw” data in Smart Grid Gateway payload processing whenever possible.
- Payload processing (in which usage and event data are loaded into the system) should be run separately from measurement processing (in which VEE is performed and final measurements are created). Do not run these processes simultaneously.
- The duration of individual initial measurements (IMDs) within a payload should be no shorter than the time elapsed between when payloads are collected and delivered (with a minimum of 4 to 6 hours per initial measurement). For example if your system is collecting and delivering measurements three times a day, the duration of individual initial measurements should be approximately eight hours.
- If you are processing “fabricated” IMDs to migrate data, keep in mind that the system has not been tested or benchmarked for this purpose.
  For example, including 30 days’ worth of measurements in a single IMD is not typical, and has not been performance tested.

Direct Measurement Processing Recommendations

- If you have not enabled Initial Load IMD Ordering on your Measuring Component Types and you are leveraging the chunkSize parameter we recommend a chunkSize that will result in smaller chunks so a smaller number of IMDs are processed per chunk. This is important because the IMDs will be processed through to completion and therefore perform many more changes on the database per IMD than legacy IMD processing.

Legacy Initial Load Measurement Processing Recommendations

- The system includes a number of Initial Measurement Data (IMD) monitor batch processes that can be used to minimize performance issues when adding large numbers of historical or backlog readings. These use the IMD as the processing work unit instead of the Device Configuration. This prevents situations in which an error on an IMD rolls back processing for all of the IMDS for a given device configuration. These include:
  - D1-IMDV2 (IMD Monitor - Physical Devices V2): used to process initial measurements related to physical devices
  - D1-IMDD2 (IMD Monitor - Deferred IMDs V2): used to process previously held or deferred IMDs
  - D1-IMDM2 (IMD Monitor - Unattached MCs V2): used to process IMDs for standalone measuring components without a Device Configuration such as weather or price data

  Refer to the Detailed Descriptions of these batch processes for additional information about each.
- We recommend configuring a monitor process on the Pending status of the Initial Load initial measurement business objects so that the VEE and measurement creation processes can be deferred to batch processing.
  - Deferring processing has the following additional benefits:
    1. It can ensure chronological processing for scalar reads which provides more efficient processing. Processing scalar reads out of order can result in additional work as reconciliation initial measurements will be created and processed to true up the consumption calculations.
    2. It can ensure measuring components are processed in order based on their relationships. For example, if a scalar channel is configured as the check channel

Copyright © 2026, Oracle and/or its affiliates.

for an interval channel, the scalar channel will be processed first so that its data is available for sum checks and any interval gap filling required.

- Configuring deferred processing involves adding the “IMD Monitor – Physical Devices” Batch Control as the Monitor Process on the Pending status to the following business objects:
  - D1-InitialLoadIMDInterval
  - D1-InitialLoadIMDScalar
- When initial measurement processing is deferred it will be necessary to schedule the “IMD Monitor – Physical Devices” Batch Control to run more frequently. It should be scheduled such that it runs with a higher thread count immediately after the core head end usage payload processing has completed and with fewer threads throughout the rest of day to pick up any other initial measurements that might be received (such as those received via On Demand Read commands).

Middleware Processing Recommendations

When processing payloads using middleware implementations, keep the number of devices to 2,000 per file for optimum usage and event processing through Oracle Service Bus. A lower number of devices, per file, will take more time for processing. A higher number of devices, per file, leads to high growth in garbage collection leading to waits and results in lower throughput. Please note, the optimum number of transactions per file may vary by head-end system.

VEE Recommendations

- The High-Low VEE Rule is very “heavy” in terms of performance. This rule has an intensive hit on performance of the Initial Measurement loading process. Customers are encouraged to use the Usage High-Low rule, which has a less significant performance impact. If the VEE rule version is being used, the historic pre window should be set as low as possible based on business needs.
- Each VEE Rule included in VEE processing can have an impact on overall performance. The table below summaries the relative costs in terms of performance for several commonly used VEE Rules. Using this table when designing your VEE groups and rules can help identify potential performance challenges.
- For example, using "Minimal Impact" rules will likely have little to no significant impact on performance, while using several "Moderate Impact" rules could result in performance challenges, and using the High Low Check rule is very likely to introduce noticeable impacts to VEE processing performance.

Relative Cost	VEE Rules
Minimal Impact (0% - 5%)	Interval Interpolation Interval Replacement Rule Interval Size Validation Interval Spike Check Negative Consumption Check Scalar Replacement Rule UOM Check

Copyright © 2026, Oracle and/or its affiliates.

Moderate Impact (5% - 20%)	Final Measurement Validation Interval Adjustment From Scalar Interval Averaging Multiplier Check Scalar Calculation From Interval Scalar Proration Sum Check
Significant Impact (20%)	High/Low Check

- Consumption sync processes should only be run on devices where the difference in consumption is outside the defined tolerance as dictated by the sum check tolerance. In a production environment, running consumption sync for two percent of devices or less is advisable.
- For cloud implementations, custom rules must be written using Groovy or Oracle Utilities Application Framework scripting.
- Run performance testing on all configuration changes performed that impact portions of Oracle Utilities Meter Data Management with high transactional processing.
- Any VEE Rules, whether provided as part of base product or custom developed, should be configured to query no more than 30 cumulative days of historical data (i.e. High/Low) to help optimize the data loading process
- A good way to troubleshoot a VEE Rule is to use the Trace section of the IMD Log. When an IMD is in Pending status, the **Trace On** button can be used to trace all VEE Rules fired during IMD processing. Once processing is complete, view the Log tab of the IMD and review the results in the **IMD Trace Log** section. For any custom developed VEE Rules, the average run time should be no longer than the average run time of other base product VEE Rules. Also, the impact of different configuration can be checked through this method as well.
- Load initial measurements and perform VEE processing every 4 to 6 hours to optimize performance.

Measurement Processing Recommendations

- Payload processing (in which usage and event data are loaded into the system) should be run separately from measurement processing (in which VEE is performed and final measurements are created). Do not run these processes simultaneously.
- When performing Periodic Estimation for the first time, avoid running estimation for large numbers of devices for a long periods of time. The Device Periodic Estimation (D1-DVCPE) batch process used to initiate periodic estimation is designed to perform estimations for a small number of devices per run, not the entire meter population in the system.
- Set the **Perform Changes in New Session** parameter on periodic estimation algorithms (including D1-CIITBIH and D1-ARSPE) to "Y" to improve processing when many initial measurements are being created in one run of these algorithms. When this parameter is set to "Y" each initial measurement will be created in a separate session and will be committed as they are completed. This improves performance since there are fewer changes being cached and ensures progress is saved in the event of errors during processing.

Copyright © 2026, Oracle and/or its affiliates.

Aggregation Recommendations

- When defining the **Aggregation Horizon** for aggregation processing, the horizon should ideally be around 7 days or less (barring specific requirements that call for an longer horizon, such as in the case of some final settlement calculations).

  For each day in the horizon the system aggregates each day that many times. For example, setting the Aggregation Horizon to 30 means that each day is aggregated 30 times. The benefit of aggregating a day multiple times is that each day the quality of the data could improve if initial measurements arrive late. However there is a cost associated with a large horizon because the system is aggregating a much larger set of measurement data to catch a statistically smaller amount of changes the further into the past a day is.

- A short **Aggregation Horizon** is recommended in particular when configuring Measurement Quality and Timeliness aggregations used with Oracle Utilities Analytics Visualization. Refer to Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization for more information.
  The **Aggregation Horizon** is defined on Master Measurement Aggregator Type (D2-MasterMsrmtAggregatorType).

- Dynamic Aggregation is intended to combine large populations of measuring components, devices, service points, or usage subscriptions into substantially fewer aggregation measuring components. Detailed aggregation, such as at the service point or usage subscription level, is supported for small populations but does not scale to large ones, and is generally **not recommended**.

  Each foundation or composite layer creates and stores its own measuring components and measurements. Before deployment, estimate the number created at each layer; configurations approaching the source population size can duplicate large volumes of data and cause excessive scanning, processing, storage, and re-aggregation.

Usage Transaction Recommendations

- Usage transactions should NOT be used as an extraction tool to perform the mass export of measurement data.
- Saving derived interval data and sending the data to an external system via the "Save Interval Vector" and "Extract Interval Data" options on the "Vector and Service Quantity Math" and "Get Interval Data" usage calculation rules can have a noticeable impact on usage transaction processing. Usage transaction processing can be improved by up to 45% if extraction of interval data is eliminated. Use of these options should be limited to situations in which they are absolutely required.
- Use the Usage High Low rule instead of the VEE High Low rule where possible.
- For Outbound Messages, the schema extraction for usage transactions should be converted into a RAW string operation to allow for more efficient Pre-Processing logic.

Usage Calculation Recommendations

This section includes best practices and recommendations regarding usage calculation processing.

- Oracle recommends including the Usage Calculation High/Low Rule rule in Post-Calculation Usage Calculation Groups to validate the current usage against historical

Copyright © 2026, Oracle and/or its affiliates.

usage, and ensure that any increase or decrease of the current usage relative to historical usage is within a defined tolerance.

User Interface Recommendations

- For better performance, user interface zones should be initially collapsed when not required for 90% or more of business processes. The initial state of zones (collapsed or not) can be controlled via the "Portal Preferences" tab on the **User** portal.
- The number of records returned to the user interface for a zone should be limited to 50 rows when building custom zones against large transactional tables.
- UI Map schemas should be specific to the displayed data. This will make sure that application will not visit all the elements that are not required for display.
- A good way to troubleshoot a screen in the user interface is to go to the **Preferences** in the top right hand corner of the application and choose the **Portal Preferences** tab. Choose the appropriate portal and set all zones to "Initially Collapsed". Next, navigate back to the screen that has performance issues, expand the zones one by one, and measure the execution time of each zone. This should be an accurate step-by-step representation of the full screen execution. This testing is especially important for **custom** user interfaces.
- For optimal user interface performance, Oracle Utilities Meter Data Management users should ensure their computer set to **high performance mode** so all CPU resources are used. Also, check if allocating additional memory through your browser's default settings helps improve performance.

Exception Handling Recommendations

- Don’t create To Do entries unless they are absolutely required
- Use the VEE Exception Handler (D1-VEERuleExceptionHandler) and Usage Exception Handler (D2-UsageRuleExceptionHandler) rules to limit the number of To Do entries created by VEE and usage rule processing.
- Do not create To Do entries for exceptions for which Service Investigative Orders are being creative.
- Purge To Do data on a regular basis (see Partitioning and Data Removal)

SQL Recommendations

- Avoid making unbounded SQLs statements with no boundary condition on date columns.
- Reduce CLOB searches and use physical columns wherever possible.
- Implement caching to pre-fetch data instead of issuing multiple SQLs.

Custom Rules and Processes

- When creating custom rules and processes, don’t use plug-in scripts for high volume transactions. Use Groovy or Java (on-premises implementations only).

Cloud Services

- Java is not allowed in cloud service environments, so implementations are encouraged to use Groovy when creating custom VEE and or usage calculation rules. Training regarding

Copyright © 2026, Oracle and/or its affiliates.

using Groovy with Oracle Utilities Application Framework applications can be found in the Oracle Utilities Learning Subscription available from Oracle University.

Testing Recommendations

- “Day in the Life” performance testing with production-level data volumes should be performed on a Test Environment before going live.

Java Recommendations — On-Premises Implementation Only

1. For areas where the transactional volume is high (such as Initial Measurement), use Hibernate SQL for non-CLOB fields whenever possible as opposed to reading entire Business Object. If a CLOB field must be retrieved, then either Entity or Lite BO should be used. Reading the full Business Object should be used as the last resort for high volume areas. This process should be considered for updating data as well when it provides the same functionality as updating the Business Object. These methods should only be considered if BO level validations, pre-processing, and post-processing aren't required.
2. For a simple SQL select statement needed in Java (not many joins and no complex logic), using Hibernate SQL provides a benefit over using a Business Service and a Zone since the entities are cached for Hibernate SQL.
3. "Lite" Business Objects are provided as a way to access the main fields for a BO without pulling back all of the information. Retrieving less information will speed up the process for reading the BO.

Referencing Master Data by Identifiers

- Understanding Referencing Master Data by Identifiers

Understanding Referencing Master Data by Identifiers

There are many places within admin configuration where direct references to master data can be made. Since master data relies on system generated keys this configuration often breaks once migrated to a new environment since the master data referenced does not exist in the target environment. To alleviate this issue, in each of these instances, a user defined identifier can be used instead of the system generated keys.

This has an added benefit for installations that support multiple time zones by enabling the identification of the master data to not only search by the user defined identifier but also the time zone of the instance data. This is advantageous because it allows one set of admin configuration to satisfy master data in multiple time zones since things like TOU Maps and profile measuring components will be identified at run time using the identifier and time zone.

The following types of data can be referenced by an identifier on admin configuration:

- **Device**: can be referenced by a device identifier type. An example of this can be seen on the Specification Lookup (D1-SpecificationLookup) extendable lookup.
- **Measuring Component**: can be referenced by a measuring component identifier. When multiple time zone support is enabled the measuring component's time zone will also be evaluated. There are two major patterns for this:

Copyright © 2026, Oracle and/or its affiliates.

  - *Profile Factors*: The profile factor itself will reference the measuring component by identifier and then the admin configuration will point to the profile factor. An example of this is the profile factor list on interval measuring component types.
  - *Direct Measuring Component references*: An example of this is the Final Values Overlay Profile (D1-FinalValuesOverlayProfile) extendable lookup.
- **TOU Map**: can be referenced by the TOU map template code. The appropriate TOU map is then found by searching for the appropriate TOU map that references a TOU map type for the supplied TOU map template. When multiple time zone support is enabled the TOU map type time zone will also be evaluated. An example of this is the default TOU map that can be configured in the display profile of a measuring component type.

During execution the identifier supplied will be used to find the appropriate master data entry. The following validation will occur:

When multiple time zone support is off: the search must find one and exactly one master data entry for the identifier type and value.

When multiple time zone support is on: the search must find one and exactly one master data entry for the identifier type, identifier value, and time zone. Note: some installations may want to use master data across time zones, this can still be achieved by supplying one and only one master data entry for the identifier type and value:

When the search fails to find the master data for the identifier type, identifier value, and time zone and there is only one entry for the identifier type and value that master data will be used (thus allowing one master data entry to be used across time zones)

When the search fails to find the master data for the identifier type, identifier value, and time zone and there are multiple entries for the identifier type and value an error will be encountered.

Smart Grid Gateway Native Implementations

When implementing Oracle Utilities Meter Data Management, Oracle Utilities Smart Grid Gateway, and Oracle Utilities Customer to Mater, Native implementations are recommended for both on-premises and Platform-as-a-Service (PaaS) implementations. See Smart Grid Gateway Adapter Native Implementations for more information.

Recommendations for Creating an On-Premises Production Environment

Oracle recommends that you do not clone the demonstration environment as a basis for a new production environment. The demonstration environment typically includes transactional data that will be irrelevant to your production environment and can cause unexpected issues if it is not purged correctly. The recommended process is to start a new production environment from a new installation and migrate "clean" system data (such as business objects and algorithms) and administrative data (such as sample activity types or other administrative entities) from the demonstration and/or test or development environments as applicable.

Your implementation can use bundling and/or the Content Migration Assistant to move system and administrative data. Instructions for using these tools are contained in the Bundling and Content Migration Assistant sections in the Oracle Utilities Application Framework *Administrative User Guide*.

Contact Oracle customer support if further assistance is required.

Copyright © 2026, Oracle and/or its affiliates.

Recommendations for High Volume Meter Processing

For implementations involving large volumes of meter data, Oracle recommends using Oracle Exadata. Oracle Exadata is a database machine and storage server designed to optimize enterprise-class databases and their workloads, comprising prepackaged database servers, storage servers, and storage networking.

Additional Resources

- How to Get Support
- Knowledge Base Articles
- Support Hot Topic Emails
- Embedded Help
- Leveraging Demonstration Data
- Customer User Groups
- Oracle Forums

How to Get Support

Oracle is deeply committed to technical support services for its products. The Oracle Support online portal provides a reliable, easy-to-use method for obtaining technical support for Oracle Utilities Meter Data Management. To register as a new user for Oracle Support, go to https://support.oracle.com and follow the steps shown on the screen.

Oracle policies for standard technical support services are available at http://www.oracle.com/us/support/policies/index.html.

Knowledge Base Articles

Once you have access to Oracle Support, finding Knowledge Base articles is an easy process.

1. Log in to https://support.oracle.com.
2. Select the **Knowledge** tab at the top of the Oracle Support portal.
3. Under the Knowledge Base section, type "**Oracle Utilities Meter Data Management**" as the product.
4. Enter key search topics into the **Enter search terms** box.
5. Click **Search**.

The search results will display all knowledge articles stored on Oracle Support related to Oracle Utilities Meter Data Management for your key topics.

- Important Articles

Important Articles

Copyright © 2026, Oracle and/or its affiliates.

While there is a vast library of important Knowledge Base articles on Oracle Support, the list below highlights a few of the key knowledge articles that should be helpful if you're looking for additional detailed information about Oracle Utilities Meter Data Management:

Article Name	Knowledge Base Article Number
Related Measuring Component Consumption Sync Technical Reference Paper	KB406197

Support Hot Topic Emails

Oracle Support provides a way to receive daily or weekly updates on new information posted for Oracle Utilities Meter Data Management.

1. Log in to https://support.oracle.com.
2. Select the **Settings** tab at the top of the Oracle Support portal. (Note: this tab may be hidden for you and may require clicking a dropdown on the right side of the tabs)
3. Once on the Settings screen, choose the **Hot Topics Email** from the set of options on the left.
4. Under Delivery Options, choose **Send With Selected Options**.
5. Choose your delivery frequency: either Daily or Weekly.
6. Add content which interests you: there are a number of options available. To subscribe to updates for Oracle Utilities Meter Data Management, hit the **Add...** button under Selected Products. Enter "Oracle Utilities Meter Data Management" for the Product then choose the types of information for which you'd like to receive an email.
7. Make sure you click **Apply** at the bottom once you've made all of your selections.

Embedded Help

Oracle Utilities Meter Data Management, like all Oracle Utilities Application Framework applications, provides extensive internal documentation. For example, detailed descriptions of system objects are included in the objects' maintenance portals. The lifecycle of each business object is described on the Lifecycle tab and depicted in flow diagrams on the Summary tab. This information is extremely useful for implementers and system administrators.

Embedded help is provided for all non-obvious fields in most portals and zones. If a field has associated help text, a question mark icon appears next to the field when the zone is displayed.

Leveraging Demonstration Data

One of the best ways to understand Oracle Utilities Meter Data Management is through a thorough review of the demonstration data (also referred to as "demo data") provided by Oracle. The demo data for Oracle Utilities Meter Data Management provides the following:

- An overall view of how to implement the product for common needs
- Examples of the productized solutions to solve key utility scenarios
- Some examples of adding "customer modifications" on top of Oracle Utilities Meter Data Management

Copyright © 2026, Oracle and/or its affiliates.

This demo data can be downloaded along with the Oracle Utilities Meter Data Management. It's recommended that this be installed into an isolated environment for reference purposes separate from other development or testing environments.

Once installed, the demo data scenarios can be found in two places within Oracle Utilities Meter Data Management:

- A zone named **Demo Scenarios** is shown in the Sidebar. This provides a quick method to link to the 360 Degree View for the specific demo meter.
- An extendable lookup named **Demo Scenarios Catalog** describes each of the scenarios in detail. This matches the scenarios displayed in the Demo Scenarios zone.

NOTE: Oracle recommends that you do not clone the demonstration environment as a basis for a new production environment. The demonstration environment typically includes transactional data that will be irrelevant to your production environment and can cause unexpected issues if it is not purged correctly. The recommended process is to start a new production environment from a new installation and migrate "clean" system data (such as business objects and algorithms) and administrative data (such as sample activity types or other administrative entities) from the demonstration and/or test or development environments as applicable. Instructions for using these tools are contained in the Bundling and Content Migration Assistant sections in the Oracle Utilities Application Framework *Administrative User Guide*.

Customer User Groups

A number of Customer User Groups have been established (one in the United States and others internationally) that are specific to sharing best practices and learnings about Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway (SGG). If you're interested in participating in a Customer User Group, contact the appropriate Product Manager.

Oracle Forums

Oracle Forums is a way for you to suggest ideas for Oracle Utilities solutions. Oracle Forums provides an easy way to submit feature ideas directly to product owners in Oracle Utilities. Once an idea is submitted, other users can comment on and vote for your idea to show their support.

Submitting an idea is simple. When you go to Oracle Forums, click **Sign In** to enter your Oracle account information or create a new account. Once you have signed in, select **New Post.** Select a solution category, title, and description for your idea. Click **Add Post** and product owners will be notified of your suggestion.

Customer To Meter Configuration

Oracle Utilities Customer to Meter employs functionality from Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management, and (optionally) Oracle Utilities Operational Device Management. While much of the configuration of the application is very similar to that used with the individual applications, the merging of functions requires some configuration that is unique to Customer To Meter.

This chapter describes topics specific to configuring Customer To Meter to support a seamless integration between Customer Care and Billing, Meter Data Management, and Operational Device Management functionality.

Copyright © 2026, Oracle and/or its affiliates.

- Cross Product Configuration Master Configuration
- Customer to Meter Consolidated Maintenance and Data Mapping
- Customer To Meter Table Setup Sequence
- Customer To Meter Batch Processing

Cross Product Configuration Master Configuration

The Cross Product Configuration master configuration provides a single point of access for configuration information related to consolidated data maintenance, data mapping, and other master configurations used with Customer To Meter.

To open the master configuration, select **Admin**, **General**, **Cross Product Configuration**. The master configuration opens in the **Cross Product Configuration** portal.

The **External System** field references the external system used for communication between Customer Care and Billing, Meter Data Management, and Operational Device Management including data synchronization requests, Service Order Management (SOM) queries, etc.

Consolidated Maintenance

The **Consolidated Maintenance** section documents objects that use some form of consolidated maintenance in Customer To Meter.

Merged Maintenance references the extendable lookup used to configure Merged Maintenance. Click **Joint Maintenance Lookup** to view values for this extendable lookup.

The **Consolidated Maintenance Objects** table lists objects that use Master Data Synchronization and Data Audit Maintenance for consolidated maintenance. Click on an object's name to view its maintenance object.

Customer and Device Value Mapping

The **Customer and Device Value Mapping** section lists objects used to map data between Customer Care and Billing and Meter Data Management.

SA Type Mapping references the extendable lookup used to map Customer Care and Billing service agreement (SA) types to Meter Data Management usage subscription types. Click **C2M SA Type Mapping Values** to view values for this extendable lookup.

The table lists fields that use Lookup Value Mapping to map values between Customer Care and Billing and Meter Data Management. Click the Description link to view and/or edit mapping values for a specific field. A plus sign (✚) indicates that mapping values have not been defined for that field. Click the icon to configure values for that field.

Device and Asset Value Mapping

The **Device and Asset Value Mapping** section lists objects used to map data between Meter Data Management and Operational Device Management.

The table lists fields that use Lookup Value Mapping to map values between Meter Data Management and Operational Device Management. Click the Description link to view and/or edit mapping values for a specific field. A plus sign (✚) indicates that mapping values have not been defined for that field. Click the icon to configure values for that field.

Copyright © 2026, Oracle and/or its affiliates.

Related Master Configurations

The **Related Master Configurations** section lists other master configurations used with Customer To Meter, including:

- Master Data Synchronization Configuration
- Seeder Sync Request Master Configuration
- Service Order Management Master Configuration
- Service Order Management Integration

Click the **Master Configuration** link to view a specific master configuration.

Customer to Meter Consolidated Maintenance and Data Mapping

Oracle Utilities Customer to Meter uses data from Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management, and (optionally) Oracle Utilities Operational Device Management. This chapter describes how data used by these applications can be maintained through a consolidated maintenance process, and how data in each application is mapped to similar data in other applications..

There are several types of consolidated maintenance and data mapping used by Customer to Meter, including:

- Merged Maintenance: maintenance of data in applications through use of a single portal.
- Master Data Synchronization: master data used by the applications is kept synchronized when data is created or updated.
- Data Audit Maintenance: creating specific types of Meter Data Management data triggers creation of corresponding Customer Care and Billing data via Audit algorithms
- Lookup Value Mapping: lookup values used by one application are mapped to corresponding values in other applications via extendable lookups
- Transactional Data Mapping: creating specific types of transactional data in one part of Customer to Meter triggers the creation of corresponding transactions in other parts.

- Merged Maintenance
- Customer To Meter Master Data Synchronization
- Lookup Value Mapping
- Data Audit Maintenance
- Transactional Data Mapping

Merged Maintenance

Merged maintenance allows data in multiple product—specific maintenance objects to be updated via a single maintenance portal.

Administrative Data Merged Maintenance

Merged maintenance is used with several types of administrative data in Customer Care and Billing and Meter Data Management that are used in nearly identical ways in both applications. These include:

Copyright © 2026, Oracle and/or its affiliates.

- Service Type
- SP Type (Service Point Type)
- Service Quantity Identifier
- Time of Use
- Unit of Measure

When a user creates or updates data of these types, the maintenance portal displays a superset of data elements from product-specific business objects based on a “superset” service script (defined in the Joint Maintenance Lookup extendable lookup (see below)). When the data is saved, the corresponding maintenance objects in Customer Care and Billing, Meter Data Management, and Operational Device Management are updated. For example, when a user navigates to the **Unit of Measure** portal and selects a record to update, the portal displays a combination of data from the X1-UnitOfMeasure and D1-UnitOfMeasure business objects. When the user saves the data, the changes are saved to both the UOM Customer Care and Billing maintenance object and the D1_UOM Meter Data Management maintenance object.

> **Note**
>
> Service Point Types (asset location types) are also used by Operational Device Management.

Service Point Merged Maintenance

In addition to the above types of administrative data, Customer to Meter service points are also maintained through merged maintenance. Service point merged maintenance does NOT make use of a “superset” service script, but does use product-specific business objects. Oracle Utilities Customer to Meter service points (asset locations in the case of Operational Device Management) are maintained through these product-specific business objects and extendable lookups.

The Customer to Meter base package uses the following service point business objects:

- Meter Data Management: X1D-ServicePoint
- Customer Care and Billing: X1-ServicePoint
- Operational Device Management: W2-ServicePoint

When a Customer to Meter service point is created or updated, the “Maintain CCB Service Point” (X1-CCBSPMNT) and “Maintain ODM Service Point” (X1-ODMSPMNT) Pre-Processing algorithms on the X1D-ServicePoint business object uses the C2M Mapping Field Values (X1–CCBMDM-MDMCCB-Lookup) and ODM-MDM Mapping Field Values (X1-ODMMDM-MDMODM-Lookup) extendable lookups to map lookup values between Customer Care and Billing and Meter Data Management and between Meter Data Management and Operational Device Management, and creates or updates an instance of the X1–ServicePoint and W2-ServicePoint business objects (which are used to update the their corresponding maintenance objects).

> **Note**
>
> Customer to Meter service point types should reference the “Service Point — MDM” (X1D-ServicePoint) business object.

Copyright © 2026, Oracle and/or its affiliates.

Joint Maintenance Extendable Lookup

The specific business objects and service scripts used by merged maintenance are defined in the Joint Maintenance Lookup (X1-JointMaintenance) extendable lookup. Records in this extendable lookup define the superset script that contains the union of the elements of the objects that exist in Customer Care and Billing and Meter Data Management, as well as the corresponding product-specific business objects.

Merged maintenance lookup values are defined by the following:

- **CCB Maintenance Object**: the unique identifier for the joint maintenance lookup value
- **Superset Script**: the superset script that contains the union of the elements of the objects that exists in Customer Care and Billing and Meter Data Management. Used for administrative data maintenance only.
- **CCB Business Object**: the business object used to update the Customer Care and Billing maintenance object
- **MDM Business Object**: the business object used to update the Meter Data Management maintenance object
- **ODM Business Object**: the business object used to update the Operational Device Management maintenance object

Customer To Meter Master Data Synchronization

This section describes the master data synchronization processes supported by Oracle Utilities Customer to Meter.

In Customer to Meter, Customer Care and Billing information is generally considered the data of record for account, customer, or service agreement-related data. Synchronizing master data between Customer Care and Billing tables and Meter Data Management tables ensures that all account, customer, and service point-related data is correct and up to date before usage transaction calculations are performed. Master data synchronization ensures that when specific types of Customer Care and Billing data are created or updated, corresponding Meter Data Management data is also created or updated.

Initial master data synchronization is performed via a data conversion and migration process. Initial Master Data Conversion for more information.

The table below summarizes the types of data ongoing master data synchronization supported in Customer to Meter:

Copyright © 2026, Oracle and/or its affiliates.

When Customer Care and Billing data of this type is created and/or updated:

Copyright © 2026, Oracle and/or its affiliates.

When Customer Care and Billing data of this type is created and/or updated:

Copyright © 2026, Oracle and/or its affiliates.

When Customer Care and Billing data of this type is created and/or updated:

Copyright © 2026, Oracle and/or its affiliates.

When Customer Care and Billing data of this type is created and/or updated:

Copyright © 2026, Oracle and/or its affiliates.

When Customer Care and Billing data of this type is created and/or updated:

Copyright © 2026, Oracle and/or its affiliates.

For example, when a Customer Care and Billing Service Agreement is created, the synchronization process creates a corresponding Meter Data Management usage subscription.

> **Note**
>
> Master data synchronization is a one-way process. When Customer Care and Billing data is created or updated, corresponding Meter Data Management data is created or updated, but changes made to Meter Data Management data are NOT synchronized to the Customer Care and Billing data. As a general rule, changes to the types of data outlined above should be made in the Customer Care and Billing entities instead of their Meter Data Management counterparts.

The Customer to Meter synchronization process is based on the Data Synchronization functionality of the Oracle Utilities Application Framework, and is supported through a set of maintenance objects, master configurations, business objects, and service scripts.

Maintenance Objects

Customer to Meter supports master data synchronization for the following maintenance objects.

Customer Care and Billing Maintenance Object	Meter Data Management Maintenance Object
BILL CYCLE (Bill Cycle)	D1-BILLCYCLE (Bill Cycle)
CONTRACT OPT (Contract Option)	D1-DOP (Dynamic Option)
CONTRACT OPT (Contract Option Event)	D1-DOPEVT(Dynamic Option Event)
PERSON (Person)	D1-CONTACT (Contact)
SA (Service Agreement)	D1–(Usage Subscription)

Master Configurations

The following master configurations are used to configure the sync process between Customer Care and Billing data and Meter Data Management data:

Master Configuration	Name/Description

Copyright © 2026, Oracle and/or its affiliates.

Lists all foreign key references that need resolution. Each one should dr

Copyright © 2026, Oracle and/or its affiliates.

e f e r e n c e t h e v i e w t h a t c o n t a i n s t h e e x t e r n a l k e y / p r o d u c t i o n k e y c r o s s -

Copyright © 2026, Oracle and/or its affiliates.

reference. If initial synchronization process areal so configured for ord

Copyright © 2026, Oracle and/or its affiliates.

a t a c o n v e r s i o n i n a d d i t i o n t o t h e o n g o i n g s y n c h r o n i z a t i o n p r o c e s s , t w o

Copyright © 2026, Oracle and/or its affiliates.

viewsarespecified.Forentitiesthatundergoingsynchronizat

Copyright © 2026, Oracle and/or its affiliates.

ion, an external system/IDtype mapping is specified to cater for entities

Copyright © 2026, Oracle and/or its affiliates.

s t h a t m i g h t b e s y n c h r o n i z i n g f r o m m o r e t h a n o n e e x t e r n a l s y s t e m .

Copyright © 2026, Oracle and/or its affiliates.

Lists the maintenance objects (contract, usage subscription, etc.) tha

Copyright © 2026, Oracle and/or its affiliates.

t
r
e
q
u
i
r
e
s
y
n
c
h
r
o
n
i
z
a
t
i
o
n
.
E
a
c
h
r
e
f
e
r
e
n
c
e
s
t
h
e
s
y
n
c
h
r
o
n
i
z
a
t
i
o
n
b
u
s
i
n

Copyright © 2026, Oracle and/or its affiliates.

essobjectthatneedstobeinstantiatedwhenprocessingasynchronizat

Copyright © 2026, Oracle and/or its affiliates.

ionrequestforthatmaintenanceobject.Formaintenanceobjectsthatu

Copyright © 2026, Oracle and/or its affiliates.

ndergobothinitialandtheongoingsynchronization,twobusinessobje

Copyright © 2026, Oracle and/or its affiliates.

ctsarespecified.

Outbound Data Synchronization Business Objects

When data is created for one of the above Customer Care and Billing maintenance objects, an outbound synchronization request is created to create the appropriate Meter Data Management data.

Synchronization of Customer Care and Billing data and Meter Data Management data uses the following outbound synchronization business objects:

Data Sync	OBhujseicnteNsasm0ebject

Copyright © 2026, Oracle and/or its affiliates.

Bill Cycle/Schedule > Bill Cycle/Schedule	MCDM2MBDiMl2lBCiylclCeyScylnecSRyenqcuReesqtuest
Contract Option > Dynamic Option	MCDM2MCDoMn2tCroanctrOapcttiOopntSSyymccRReeqquuessttt

Copyright © 2026, Oracle and/or its affiliates.

Contract Option Event > Dynamic Option Event	MCDIM2MCDoMn2tCroanctrOOpptt iEovnt ESvyencnctRSeyqnuceRsetquest
Person > Contact	MCDIM2MPDeMr2sPoenrSsyomncSRyenqcuReesqtuest

Copyright © 2026, Oracle and/or its affiliates.

Service Agreement > Usage Subscription	MCD1M2MSDAMS2ySnAcSRyenqcuReesqtuest

Inbound Data Synchronization Service Scripts

When outbound synchronization requests enter the “Send Request” state, Enter algorithms invoke service scripts to create corresponding inbound data synchronization requests.

Synchronization of Customer Care and Billing data and Meter Data Management data uses the following inbound data synchronization service scripts:

Data Sync	NSaemevicecScript

Copyright © 2026, Oracle and/or its affiliates.

Bill Cycle/Schedule > Bill Cycle/Schedule	BX i1 l- lI Cn yB cC lS ey In nS bS o u n d S y n c R e q u e s t
Contract Option > Dynamic Option	IX n1 b- oI un nS dy DD yy nO ap m i c O p t i o n S y n c

Copyright © 2026, Oracle and/or its affiliates.

Contract Option Event > Dynamic Option Event	IXmlbolunnSdyDDyynOapmEicOptionEventSync
Person > Contact	IXmlbolunnSdyCCoomttactSync

Copyright © 2026, Oracle and/or its affiliates.

Inbound Data Synchronization Business Objects

When outbound synchronization requests enter the “Send Request” state, service scripts create corresponding inbound data synchronization requests. Master data synchronization uses **ongoing synchronization requests** to continuously keep Meter Data Management data synchronized with corresponding Customer Care and Billing data. Ongoing synchronization requests can be used to maintain contacts, usage subscriptions, dynamic options, and bill cycles.

Ongoing synchronization of Customer Care and Billing data and Meter Data Management data uses the following data synchronization business objects:

Copyright © 2026, Oracle and/or its affiliates.

Data Sync

OBnugs oi in negs SsyOnbcj hercotnization Object Name

Copyright © 2026, Oracle and/or its affiliates.

IDn2b-oIunmbdoBuinldlBCiylcllCeyScylnecSRyenqcuReesqtuest

Copyright © 2026, Oracle and/or its affiliates.

Copyright © 2026, Oracle and/or its affiliates.

Copyright © 2026, Oracle and/or its affiliates.

Copyright © 2026, Oracle and/or its affiliates.

Service Agreement > Usage Subscription	UDs2a-gOenSguohisncgrSiypntciRoemqOunegsotiUnSgSyncRequest

Example Synchronization Process Steps

This section provides an overview of the processing that takes place when an ongoing synchronization request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

Step	POrbojceecstss

Copyright © 2026, Oracle and/or its affiliates.

1

Data in Customer Care and Billing is created or changed, triggering asyn

Copyright © 2026, Oracle and/or its affiliates.

chronization request. For example, a service agreement is updated, and

Copyright © 2026, Oracle and/or its affiliates.

the corresponding usage subscription must be updated.

Copyright © 2026, Oracle and/or its affiliates.

2

AO
nu
ot
ub
to
bu
on
ud
nS
dy
sn
yc
nh
cr
ho
rn
oi
nz
ia
zt
ai
to
in
oR
ne
rq
eu
qe
us
et
sb
tu
is
si
cn
re
es
as
to
eb
dj
.e
c
t
:
C
1
-
M
D
M
2
S
A
S
y
n
c
R
e
q

Copyright © 2026, Oracle and/or its affiliates.

u
e
s
t

Copyright © 2026, Oracle and/or its affiliates.

3

WhenbrotuhnedsSyymcchhrroomiizzaattioomrSeeqruveiscteSecnrtieprts:tXhe-“ISnendyURSequest”status,an

Copyright © 2026, Oracle and/or its affiliates.

Enter algorithm invokes the inbound synchronization service script to

Copyright © 2026, Oracle and/or its affiliates.

createaninboundsyncrequest.

Copyright © 2026, Oracle and/or its affiliates.

request. In this case, it's an ongoing synchronization request.

Copyright © 2026, Oracle and/or its affiliates.

Copyright © 2026, Oracle and/or its affiliates.

ea
dt
ba
yA
El
ng
to
er
ri
at
lh
gm
α
rD
i1
t-
hS
mR
s-
oP
mR
tE
hA
eD
sD
yR
ne
cs
ho
rl
ov
ne
iK
ze
ay
ts
i-
oO
mn
rg
eo
qi
un
eg
sS
ty
bn
uc
sA
il
ng
eo
sr
si
ot
bh
jm
e:
cD
t1
l-

Copyright © 2026, Oracle and/or its affiliates.

iRfEeScKyEcYlFeAsLtSatyneCsRe:

- questUpdateDataAlgorithm: D1-SR-UPDDAT

DataTransformed/BasicSchemaValidatedDetermine

Copyright © 2026, Oracle and/or its affiliates.

- SyncRequestBOSetupTransformedDataPre-AddedSyncRequestPre-Add

Copyright © 2026, Oracle and/or its affiliates.

- DataFKsResolved
- ResolveKeys-OngoingSyncUpdating
- SyncRequestUpd

Copyright © 2026, Oracle and/or its affiliates.

- Initial Master Data Conversion
- Operational Device Management Data Synchronization

Initial Master Data Conversion

This section describes the initial master data conversion processes supported when migrating data from Oracle Utilities Customer Care and Billing to Oracle Utilities Customer to Meter.

Initial data conversion facilitates creation of data in Meter Data Management and Operational Device Management based on corresponding data in Customer Care and Billing. After creating master data (and its corresponding administrative data) in Customer Care and Billing, batch processing can be used to create corresponding master data (and its corresponding administrative data) in Meter Data Management and Operational Device Management. The table below lists the specific types of master data that can be converted using this process.

Customer Care and Billing Data	Meter Data Management / Operational Device Management Data
Service Point	Service Point (Node)
Person	Contact
Service Agreement	Usage Subscription
Meter	Device (Asset)
Item	Device (Asset)
Service Point Item History	Install Event (Asset Disposition)
Service Point Meter History	Install Event (Asset Disposition)
Meter Reads	Measurements
Interval Profile	Measuring Component
Contract Options and Contract Option Events	Dynamic Options and Dynamic Options Events

CCB/MDM/ODM Data Migration Master Configuration

In addition to the above listed master data, the initial data conversion process also converts various administration data from Customer Care and Billing to Meter Data Management and Operational Device Management. The CCB/MDM/ODM Data Migration Master Configuration is used to identify how specific types of administration data are to be migrated, including:

- **Interval Meter Types**: a list of Customer Care and Billing interval meter types that will be converted to Meter Data Management device types
- **Interval Profile Types**: a list of Customer Care and Billing interval profile types that will be converted to Meter Data Management measuring component types
- **Tracked and Item Types**: a list of Customer Care and Billing meter types and item types that will be tracked as assets and components in Operational Device Management

Copyright © 2026, Oracle and/or its affiliates.

- **Out of Service Location Type**: the Out of Service Location Type used in creating Out of Service Location for meter and item data migration
- **Out of Service Location**: the Out of Service Location used on assets created from retired Customer Care & Billing meters
- **Service Agreement Types Pivot to Bill Determinants**: Allows you to navigate to the Service Agreement Type portal, to review and manually update the **Special Role** flag to “Bill Determinants Required” for those service agreement types that should be pivoted to require bill determinants (in other words, those that require service quantities created by Meter Data Management usage transactions).
- **Inflight FA Configuration**: options related to field activity related administration data, including whether or not to create Service Order Management field activity data, whether or not to include a Field Activity Type Profile on service point types, and mapping of Customer Care and Billing field activity types to Service Order Management field task types.

Refer to the embedded help for each of the above for more information about each.

Interval Profile Migration

The **Interval Profile** section of the master configuration should be configured to define the specific business objects to use when creating measuring component types and measuring components based on interval profiles. Interval profile measuring components should be created as "standalone" measuring components, Options include the following (values in parentheses are recommended settings):

- **Measuring Component Type BO**: Defines the measuring component type business object to use when creating measuring component types based on interval profile types (D1-StandaloneIntrvlMCType).
- **Sub Type/Measuring Component BO Mapping**: Maps Interval Profile Sub Types to specific measuring component business objects (D1-StandaloneInterval).
- **Migrate SA Profile**: Indicates if SA Profiles should be migrated.
- **Interval Profile Relationship Type Mapping**: Maps CCB Interval Profile Relationship Types to MDM Direct Channel Relationship Types (Standalone Data Request)

Data Conversion Batch Controls

Batch controls perform processing for data conversions, including such as allocating keys to data, resolving foreign keys, and loading data.

The table below lists the batch controls used for each type of data:

Customer Care and Billing Data	Meter Data Management Data	Batch Job
Administrative Data	Administrative Data	Migrate Admin Data (X1-MIGAD)
Service Point	Service Point	Migrate Service Point (X1-MIGSP)
Person	Contact	Migrate Person (X1-MIGP)
Service Agreement	Usage Subscription	Migrate Service Agreement (X1-MIGSA)
Meter	Device	Migrate Meter (X1-MIGMT)
Item	Device	Migrate Item (X1-MIGIT)
Interval Profile	Measuring Component	Migrate Interval Profile (X1-MIGIP)
SP Item History	Install Event	Migrate SP Item History (X1-MIGSI)

Copyright © 2026, Oracle and/or its affiliates.

Customer Care and Billing Data	Meter Data Management Data	Batch Job
SP Meter History	Install Event	Migrate SP Meter History (X1-MIGSM)
Meter Reads	Measurements	Migrate Meter Reads (X1-MIGMR)
Contract Options	Dynamic Options	Migrate Contract Options (X1-MIGCO)

Processing Modes

The data conversion batch controls listed above include parameters to control the manner in which data conversion processing is performed. These parameters include the following:

- Low ID Override Value: Used to specify an ID value to override the low ID in the thread ID range calculation. This, along with the High ID Override Value, can be used to limit the number of records processed by each batch run.
- High ID Override Value: Used to specify an ID value to override the high ID in the thread ID range calculation. This, along with the Low ID Override Value, can be used to limit the number of records processed by each batch run.
- Mode: Used to specify whether the batch process will Insert or Delete data.
- Migration Option: Used to specify the type of data migration being processed, One Time Migration or Staggered. Staggered mode inserts an ad hoc characteristic that is used to detect if the record has been changed since its initial creation (used by the “Delete” mode).

For implementations with relatively small data volumes, in which data conversion can be performed in a single set of batch processes, these jobs can be run in the “Insert” Mode using the “One Time Migration” Migration Option.

For implementations with larger data volumes, in which data conversion might require running multiple instances of each batch process, the Mode and Migration Option parameters can be used to adopt a “staggered” approach to migrating the data. This mode allows identification of data that may have been updated since its initial creation which can be deleted, and then re-inserted. Running batch processing in staggered mode involves running a series of batch process runs, as follows:

1. Mode: Insert / Migration Option: Staggered — This performs the initial data conversion/ migration, and flags records with an ad hoc characteristic noting the record’s version at the time of the migration, which is used to detect if the record has been changed since its initial creation.
2. Mode: Delete — This removes records that have changed in production (by comparing the versions recorded in the ad hoc characteristic) as well as records that have been deleted in production.
3. Mode: Insert / Migration Option: Staggered — This re-inserts records deleted in the previous step, including those that have changed in production since the initial migration, and inserts new records that were added in production since the initial migration.

Batch Process Sequence

The table below lists the sequence in which these migration batch processes should be executed.

Sequence	Maintenance Object
1	Admin
2	Item

Copyright © 2026, Oracle and/or its affiliates.

Sequence	Maintenance Object
3	Meter
4	Interval Profile and Interval Profile Data (see Interval Profile Migration and Converting Interval Profile Data Sets for more information.
5	Service Point
6	Service Point Item History
7	Service Point Meter History
8	Person
9	Contract Option
10	Service Agreement
11	Meter Read

Operational Device Management Data Synchronization

This section describes the master data synchronization processes used to synchronize data between Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management, including:

- Asset and Device Synchronization
- Install Event and Asset Disposition Synchronization
- Contact and Contact Type Synchronization
- Manufacturer Synchronization
- Field Task Type and Asset Activity Type Synchronization

Asset and Device Synchronization

When assets are created (or updated) in Operational Device Management, a related Meter Data Management device is created / updated. This synchronization is performed in a manner very similar to the process outlined in Customer To Meter Master Data Synchronization. The primary difference in how asset-device synchronization works is that instead of triggering the process based on the Asset maintenance object, the synchronization is triggered via the Generic Change Data Capture (W1-GCHG-CDCP) Audit algorithm on the parent business object (W1–TrackedAsset) of the following asset business objects:

- Item (W1-Item)
- Smart Meter (W2-SmartMeter)
- Analog Meter (W2-ManualMeter)
- Communication Component (W2-CommunicationComponent)

For asset-device synchronization, the specific synchronization objects used include the following:

- **Outbound Data Synchronization Business Object**: MDM Asset Synchronization Request (W1-MDMAssetSyncRequest)
- **Inbound Data Synchronization Service Script**: Inbound Device Sync (X1-InSyDvc)
- **Inbound Data Synchronization Business Object**: Device Ongoing Sync Request (D1-OngoingSyncRequestDevice)

Copyright © 2026, Oracle and/or its affiliates.

Install Event and Asset Disposition Synchronization

When a device is installed at a service point, the disposition of its related asset is updated to "Installed". The update to the device's install event triggers the update to the asset disposition, changing its **Status** and **Asset Disposition** to "Installed" and its **Location** to the service point with an effective date matching the install event's **Installation Date/Time**. This synchronization is performed in a manner very similar to the process outlined in Customer To Meter Master Data Synchronization. In this case, the synchronization is triggered via the Generic Change Data Capture (W1-GCHG-CDCP) Audit algorithm on the Install Event maintenance object (D1-INSTLEVT).

For install event / asset disposition synchronization, the specific synchronization objects used include the following:

- **Outbound Data Synchronization Business Object**: ODM Install Event Sync request (D1-InstallEventSyncRequest)
- **Inbound Data Synchronization Service Script**: Inbound Asset-Node Sync (X1-InSyAstNd)
- **Inbound Data Synchronization Business Object**: Asset-Node Ongoing Sync Request (W1-OngoingSyncRequestAssetNode)

Contact and Contact Type Synchronization

Contacts in Operational Device Management can be associated to asset locations (service points). Operational Device Management contacts are created when a Meter Data Management service point has a contact associated with it via the Maintain ODM Service Point (X1-ODMSPMNT) algorithm on the Service Point - MDM (X1D-ServicePoint) business object (see Service Point Merged Maintenance for more information about how service points are maintained).

Operational Device Management contacts and contact types are maintained via Pre-Processing algorithms on the following Meter Data Management business objects:

Meter Data Management Business Object	Pre-Processing Algorithm	Operational Device Management Business Object
Business (D1-Business)	Maintain ODM Business (X1-MNTODMBUS)	Business (W1-Business)
Person (D1-Person)	Maintain ODM Person (X1-MNTODMPER)	Person (W1-Person)
Contact Type (D1-ContactType)	Maintain ODM Contact Type (X1-MNTODMCTY)	Contact Type (W1-ContactType)

> **Note**
>
> Out of service locations reference Meter Data Management contacts.

Manufacturer Synchronization

When manufacturers are created or updated in Operational Device Management, corresponding data is created or updated in Meter Data Management.

When a manufacturer is created or updated in Operational Device Management, a corresponding Meter Data Management manufacturer is created / updated. This

Copyright © 2026, Oracle and/or its affiliates.

synchronization is performed via the Manufacturer Joint Maintenance (X1-MAINTMANF) Post-Processing algorithm on the W1-Manufacturer business object.

> **Note**
>
> Operational Device Management is considered the “system of record” for manufacturers and models in Customer To Meter.

Field Task Type and Asset Activity Type Synchronization

When Field Task Types are created in Service Order Management (as values for the Field Task Type extendable lookup), corresponding Operational Device Management Asset Activity Types are created. This synchronization is performed via the Maintain MDM/SOM field task type to a corresponding ODM activity type (X1-MNTFLDTSK) Post-Processing algorithm on the D1-FieldTaskTypeLookup business object.

Asset activity types created in this manner are based on the Service Order Management Activity Type (W1-SOMActivityType) business object,. The **Activity Business Object** for these asset activity types is Service Order Management Activity (W1-SOMActivity).

Lookup Value Mapping

Oracle Utilities Customer to Meter service points and other records reference field and lookup values from Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management, and Oracle Utilities Operational Device Management. This section describes how field and lookup values used by Customer Care and Billing are mapped to similar field and lookup values used by Oracle Utilities Meter Data Management, and how field and lookup values used by Oracle Utilities Operational Device Management are mapped to similar field and lookup values used by Oracle Utilities Meter Data Management

Customer Care and Billing / Meter Data Management Mapping

Mapping of field and lookup values between Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management is supported through the use of two extendable lookups:

- **C2M Mapping Field List** (X1-CCBMDM-MDMCCB-MappingField): used to capture the fields that require translation mapping between Customer Care and Billing and Meter Data Maintenance maintenance objects in Customer to Meter. Values in this extendable lookup are identified by a field or lookup used by Customer Care and Billing, such as Disconnect Location (DISCON_LOC_CD) or Life Support / Sensitive Load (LS_SL_FLG).
- **C2M Mapping Field Values** (X1–CCBMDM-MDMCCB-Lookup): used to capture the mapping values for the fields and lookups defined in the C2M Mapping Field List extendable lookup. Values in this extendable lookup are identified by a Customer Care and Billing field or lookup defined in the C2M Mapping Field List extendable lookup. Each value also defines the specific value mappings for the field or lookup.

  Examples

  Value Mappings for the Disconnect Location (DISCON_LOC_CD) field include the following:

CCB Value	MDM Value
Disconnected at meter	Device

Copyright © 2026, Oracle and/or its affiliates.

CCB Value	MDM Value
Disconnected at pole	Source

Value Mappings for the Activity Type (FA_TYPE_CD) lookup include the following:

CCB Value	MDM Value	MDM Value 2
Start Service	Enable Service	
Stop Service	Disable Service	
Re-read meter	Field Activity	Read Meter

Value Mappings for the Life Support / Sensitive Load (LS_SL_FLG) lookup include the following:

CCB Value	MDM Value
None	Not on life support
LS/SL	On life support

Operational Device Management / Meter Data Management Mapping

Mapping of field and lookup values between Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management is supported through the use of two extendable lookups:

- **ODM—MDM Mapping Field List** (X1-ODMMDM-MDMODM-MappingField): used to capture the fields that require translation mapping between Operational Device Management and Meter Data Management maintenance objects in Customer to Meter. Values in this extendable lookup are identified by a field or lookup used by Meter Data Management, such as the Life Support / Sensitive Load (LIFE_SUPPORT_SENS_LOAD_FLG) or Ok to Enter (OK_TO_ENTER_FLG).
- **ODM—MDM Mapping Field Values** (X1–ODMMDM-MDMODM-Lookup): used to capture the mapping values for the fields and lookups defined in the C2M Mapping Field List extendable lookup. Values in this extendable lookup are identified by a Operational Device Management field or lookup defined in the ODM—MDM Mapping Field List extendable lookup. Each value also defines the specific value mappings for the field or lookup.

  Examples

  Value Mappings for the Life Support / Sensitive Load (LIFE_SUPPORT_SENS_LOAD_FLG) lookup are as follows:

ODM Value	MDM Value
Life Support/Sensitive Load	LS/SL
None	None

Value Mappings for the Ok to Enter (OK_TO_ENTER_FLG) field are as follows:

ODM Value	MDM Value
No	False
Yes	True

Copyright © 2026, Oracle and/or its affiliates.

Data Audit Maintenance

Different functions in Customer Care and Billing and Meter Data Management use similar types of data when calculating usage and billing charges, and it's critical that the data be consistent between the applications. For example, both Usage Rules in Meter Data Management and Calculation Rules in Customer Care and Billing make use of TOU Groups, TOU Map Templates, TOU Map Types, etc., and when calculating usage and billing charges for an account, the data used by both systems should be effectively identical. The process of maintaining similar sets of data usable by each system is called Data Audit Maintenance.

When specific types of Meter Data Management data is created or updated, Audit algorithms create or update corresponding Customer Care and Billing data. For instance, when a TOU Map Type usable by Meter Data Management Usage Rules is created, data auditing creates a similar TOU Map Type usable by Customer Care and Billing Calculation Rules.

Data audit maintenance is used with several types of Meter Data Management data in Customer to Meter, including:

- TOU Group
- TOU Map Template
- TOU Map Type
- TOU Map
- Device Type (Item Type)

The table below outlines the business objects and algorithms used by data audit maintenance:

Data Type	Business Object	Audit Algorithm
TOU Group	D2-TOUGroup	Maintain CCB TOU Group (X1-MCCBTOUGP)
TOU Map Template	D2-TOUMapTemplate	Maintain CCB TOU Map Template (X1-MCCBTOUTM)
TOU Map Type	D2-TOUMapType	Maintain CCB TOU Map Type (X1-MCCBTOUMT)
TOU Map	D2-TOUMap	Maintain CCB TOU Map (X1-MCCBTOUMP)
Device Type (Item Type)	D1-ItemType	Maintain CCB Item Type (X1-MCCBITEMT)

Refer to the Business Object and Algorithm portals to view additional details regarding these audit algorithms.

Other Data Auditing

In addition to the objects listed above, Customer To Meter uses data auditing with other objects in Customer Care and Billing to ensure that changes to data reflected in related data in Meter Data Management. These objects use Audit algorithms on their maintenance objects to capture any changes and update the corresponding data. The objects that make use of this approach include the following:

- **Account**: When service agreement-related account data is changed in Customer Care and Billing, the audit algorithm creates a Service Agreement outbound data synchronization request to captures the change and update the corresponding usage subscription in Meter Data Management.

Copyright © 2026, Oracle and/or its affiliates.

- **Premise**: When a Customer Care and Billing premise is changed, the audit algorithm updates the address information in the service point in Meter Data Management.

The table below outlines the data types, maintenance objects, and algorithms used by these.

Data Type	Maintenance Object	Audit Algorithm
Account	ACCOUNT	C2M Account Change Data Capture (SA-Based) (X1-ACCTCDCSA)
Premise	PREMISE	Update MDM SP from CCB Premise (X1-UPSPFRPR)

Refer to the Maintenance Object and Algorithm portals to view additional details regarding these audit algorithms.

Transactional Data Mapping

Certain types of transactional data created in Customer Care and Billing result in related transactional data in Meter Data Management (and vice-versa). This section describes these transactional mappings.

Usage Requests / Usage Transactions

Customer Care and Billing creates usage requests during the billing process to request usage calculations in Meter Data Management. These requests in turn create Meter Data Management usage transactions, which perform the usage calculations.

Customer Care and Billing usage requests use the following business objects:

- MDM Cyclical Usage Request (C1–UsageRequestCyclicalBilling): Used with usage requests created from the cyclical or batch billing process.
- MDM Non-Cyclical Usage Request (C1–UsageRequestNonCyclical): Used with usage requests created from non-cyclical or online bill generation.
- MDM Usage Request (C1–UsageRequest): Parent business object that defines the lifecycle for Cyclical and Non-Cyclical usage requests.

When a usage request enters the “Send Request” state, the “Send Usage Request Message to MDM” (X1–SENDUSGR) Enter algorithm invokes the “Usage Transaction Request Inbound” (X1-UsgTrRqIn) service script which uses the Usage Transaction Seeder (D1–UsgTranSeeder) business object to create a usage transaction.

When a Meter Data Management usage transaction enters a “Subsequent Correction” state, the “X1-SEND-SUBC” Enter algorithm invokes the “Call Subsequent Correction Notification” (X1-SendSubC) service script to process a corrected read notification for Customer Care and Billing by finding the account associated with the corrected read and creating an off cycle bill generator to handle associated correction processing.

Refer to The Big Picture of Usage Requests for more information about usage requests and usage transactions.

Field Activity Requests / Service Order Activities

Customer Care and Billing creates Field Activity Requests to initiate service orders to perform tasks at a service point. Most field activity requests are created:

- When a customer service representative starts or stops service at a premise. When this happens, the system automatically creates field activity requests for each service point.

Copyright © 2026, Oracle and/or its affiliates.

- When service is cut due to lack of payment or reconnected for sufficient payment or arrangement.

When Customer Care and Billing creates a field activity request, a corresponding Service Order Activity is created to orchestrate the various activities required to fulfill the service order by continuously monitoring the status of the service point, device installation, and related measurements. This orchestration can include field activities to instruct a field crew to install, remove, or exchange a meter, as well as eligible smart meter command activities to connect or disconnect the meter, etc.

Service Order Field Activities are specific activities that typically exist only in Service Order Management and not Customer Care and Billing. However, field activity remarks notification, appointment notification (if required by the service order field activity), a missed appointment (which can create an adjustment) and create customer contact actions can be invoked in Customer Care and Billing when triggered from service order field activities.

Refer to Understanding Service Order Management for more information about service order activity processing.

Other Transactional Data

Data synchronization of interval register data, interval profiles, and field activities is performed via a data conversion and migration process. See Transactional Data Conversion for more information.

- Transactional Data Conversion

Transactional Data Conversion

This section describes transactional data conversion processes supported when migrating data from Oracle Utilities Customer Care and Billing to Oracle Utilities Customer to Meter or Oracle Utilities Customer Cloud Service, including conversion of interval register data, interval profiles, and “inflight” field activities.

- Converting Interval Register Data
- Converting Interval Profile Data Sets
- Converting Field Activities

Converting Interval Register Data

Interval register data from Oracle Utilities Customer Care and Billing can be converted into Measurement data for use in Customer to Meter (and Customer Cloud Service).

Note: Refer to **Maintaining Interval Data** and **Maintaining Interval Register Data** in the *Oracle Utilities Customer Care and Billing Business User Guide* for more information about interval register data in Oracle Utilities Customer Care and Billing.

Interval Registers should have been migrated to Interval Measuring Components as part of the Initial Master Data Conversion process (see Initial Master Data Conversion) . Once the registers have been converted, register reads (also known as Register Data Sets) can be converted to measurement data.

A batch control (see below) is used to convert register read data from Interval Register reads to the Customer to Meter (Customer Cloud Service) Measurement table. Processing can be defined for a specified year and month range, and optionally for a range of Register IDs.

Processing Notes:

Copyright © 2026, Oracle and/or its affiliates.

- Only Register Data Sets with a status of "Complete" will be converted. Data sets that have 'No Data' will be excluded from the conversion.
- No validation is performed on the interval data. All the interval data is assumed to be valid.
- If the **Data Set Type** is not defined for the register data set, the converted measurements will have a Condition of "Regular".
- Register data sets with a **Data Set Type** of "Estimated" will result in measurements with a Condition of "Estimated".
- If overlapping register data sets are processed, the values from the data set with the latest Set Data Time are selected for the conversion.
- The conversion process will not fill-in data for periods without interval data.

Creation of records in the Measurement table is be performed via an Insert Select of the Interval Register Data, for a month's worth of data at a time, which will be immediately committed.

Conversion should only be run for a given range of Register IDs for a given date range once. Attempting to convert the Register IDs for the same date range multiple times can result in errors.

Interval Register Data Conversion Batch Control

The Migrate Register Data Set (X1-MIGRD) batch control performs processing for interval register data conversion.

The registers to be converted for each batch run can be defined using the Register ID Low and Register ID High parameters. To process data for a single register, you can set both of these parameters to the same Register ID. If neither parameter is defined, the process will convert data for all registers.

The batch process can convert up to 1 year of data at a time, the specific year is designated by the Year parameter. The Start Month and End Month parameters can be used to further limit the time range for the conversion.

This batch control uses the following parameters:

- **Register ID Low**: Optional. Used to specify an ID value to override the low ID in the thread ID range calculation. This, along with the Register ID High be used to limit the number of records processed by each batch run. If not specified, the process will convert data for all registers whose ID is lower than the Register ID High parameter (if defined).
- **Register ID High**: Optional. Used to specify an ID value to override the high ID in the thread ID range calculation. This, along with the Register ID Low, can be used to limit the number of records processed by each batch run. If not specified, the process will convert data for all registers whose ID is higher than the Register ID Low parameter (if defined)
- **Year**: The year for which data will be converted. (Required)
- **Start Month**: The start month of the time range for which data will be converted. (Required)
- **End Month**: The start month of the time range for which data will be converted. (Required)

Examples:

To convert data for all registers from the year 2020, you would set the parameters as follows:

- Year: 2020
- Start Month: 01

Copyright © 2026, Oracle and/or its affiliates.

- End Month: 12

To convert data for all registers for the months of January through May of 2020, you would set the parameters as follows:

- Year: 2020
- Start Month: 01
- End Month: 05

To convert data for registers whose IDs range from 10001000 to 10009000 for from the year 2020, you would set the parameters as follows:

- Register ID Low: 10001000
- Register ID High: 10009000
- Year: 2020
- Start Month: 01
- End Month: 12

To convert data for registers whose IDs range from 1001000 to 1009000 for the months of January through May of 2020, you would set the parameters as follows:

- Register ID Low: 10001000
- Register ID High: 10009000
- Year: 2020
- Start Month: 01
- End Month: 05

Converting Interval Profile Data Sets

Interval profile data sets from Oracle Utilities Customer Care and Billing can be converted into measuring components and measurement data for use in Customer to Meter (and Customer Cloud Service).

Note: Refer to **Maintaining Interval Data** and **Maintaining Interval Profile Data** in the *Oracle Utilities Customer Care and Billing Administrative User Guide* for more information about interval profile data in Oracle Utilities Customer Care and Billing.

Interval Profiles should have been migrated to Interval Measuring Components as part of the Initial Master Data Conversion process (see Initial Master Data Conversion). Once the profiles have been converted, profile data can be converted to measurement data.

A batch control (see below) is used to convert interval profile data to the Customer to Meter (Customer Cloud Service) Measurement table. Processing can be defined for a specified year and month range, and optionally for a range of Register IDs.

Processing Notes:

- Only Profile Data Sets with a status of "Complete" will be converted. Data sets that have 'No Data' will be excluded from the conversion.
- No validation is performed on the interval profile data. All the interval profile data is assumed to be valid.
- If the **Data Set Type** is not defined for the profile data set, the converted measurements will have a Condition of "Regular".

Copyright © 2026, Oracle and/or its affiliates.

- Profile data sets with a **Data Set Type** of "Estimated" will result in measurements with a Condition of "Estimated".
- If overlapping profile data sets are processed, the values from the data set with the latest Set Data Time are selected for the conversion.
- The conversion process will not fill-in data for periods without interval data.

Creation of records in the Measurement table is be performed via an Insert Select of the Interval Profile Data, for a month's worth of data at a time, which will be immediately committed.

Conversion should only be run for a given range of Profile IDs for a given date range once. Attempting to convert the Profile IDs for the same date range multiple times can result in errors.

Data Conversion Batch Control

The Migrate Interval Profile Data Set (X1-MIGID) batch control performs processing for interval profile data set conversion.

The profiles to be converted for each batch run can be defined using the Interval Profile ID Low and Interval Profile ID High parameters. To process data for a single interval profile, you can set both of these parameters to the same Interval Profile ID. If neither parameter is defined, the process will convert data for all interval profiles.

The batch process can convert up to 1 year of data at a time, the specific year is designated by the Year parameter. The Start Month and End Month parameters can be used to further limit the time range for the conversion.

This batch control uses the following parameters:

- **Interval Profile ID Low**: Optional. Used to specify an ID value to override the low ID in the thread ID range calculation. This, along with the Interval Profile ID High be used to limit the number of records processed by each batch run. If not specified, the process will convert data for all interval profiles whose ID is lower than the Interval Profile ID High parameter (if defined).
- **Interval Profile ID High**: Optional. Used to specify an ID value to override the high ID in the thread ID range calculation. This, along with the Interval Profile ID Low, can be used to limit the number of records processed by each batch run. If not specified, the process will convert data for all interval profiles whose ID is higher than the Interval Profile ID Low parameter (if defined)
- **Year**: The year for which data will be converted. (Required)
- **Start Month**: The start month of the time range for which data will be converted. (Required)
- **End Month**: The end month of the time range for which data will be converted. (Required)
- **Clear Measurements**: Flag indicating whether or not to delete existing measurements within the date range

Examples:

To convert data for all interval profiles from the year 2020, you would set the parameters as follows:

- Year: 2020
- Start Month: 01
- End Month: 12

Copyright © 2026, Oracle and/or its affiliates.

To convert data for all interval profiles for the months of January through May of 2020, you would set the parameters as follows:

- Year: 2020
- Start Month: 01
- End Month: 05

To convert data for interval profiles whose IDs range from 10001000 to 10009000 for from the year 2020, you would set the parameters as follows:

- Interval Profile ID Low: 10001000
- Interval Profile ID High: 10009000
- Year: 2020
- Start Month: 01
- End Month: 12

To convert data for interval profiles whose IDs range from 100100 to 100900 for the months of January through May of 2020, you would set the parameters as follows:

- Interval Profile ID Low: 10001000
- Interval Profile ID High: 10009000
- Year: 2020
- Start Month: 01
- End Month: 05

Converting Field Activities

Completed and inflight (activities that are in the "Pending" state) customer event orchestration events and field activities from Oracle Utilities Customer Care and Billing can be converted into corresponding field activity requests and service order activities and service order field activities for use in Customer to Meter (and Customer Cloud Service).

> **Note**
>
> Refer to **Service Orders** in the *Oracle Utilities Customer Care and Billing Administrative User Guide* for more information about field activities Oracle Utilities Customer Care and Billing. Refer to Service Orders in the *Business User Guide* for more information about service order activities and service order field activities in Customer to Meter and Customer Cloud Service,

A batch control (see below) is used to create new field activity requests and corresponding Service Order Management activities. Processing can be defined for a specified date range and optionally for activities of a specific status, and/or a range of field activity IDs.

Administration Data

Before this conversion process can be run, the Migrate Admin Data (X1-MIGAD) batch process must be run. Note that the **Inflight FA Configuration** section of the CCB/MDM/ODM Data Migration master configuration must be configured to map Customer Care and Billing field activity types to Field Task Types before the Migrate Admin Data batch process is run. See

Copyright © 2026, Oracle and/or its affiliates.

Initial Master Data Conversion for more information about the CCB/MDM/ODM Data Migration master configuration.

The Migrate Admin Data process performs the following field activity related steps:

- Creates Customer Care and Billing field activity types used as customer event orchestration in Service Order Management (such as SOM-STRT (Start Service), SOM-STOP (Stop Service), and so on).
- Creates Service Order Management service order activity types and service order field activity types (Enable Service, Disable Service, Field Activity, and so on).
- Creates a "Service Order Management Profile" (SOM) Field Activity Type Profile mapping Customer Care and Billing field activity types for customer event orchestration events, such as Enable Service, Disable Service, Back to Back, Cut for Non Payment and Reconnect for Payment to corresponding Service Order Management orchestration activity types
- Creates and/or updates the Field Activity Type Value in the C2M Mapping Field Values extendable lookup mapping Customer Care and Billing customer event orchestration field activity types to corresponding Service Order Management orchestration activity types
- Copies the CCB FA Cancel Reason to MDM FA Cancel reason extendable look-up
- Copies the CCB FA Reschedule Reason to MDM FA reschedule reason extendable look-up
- Creates outbound and inbound communication types (Field Activity Outbound Communication Type and Field Activity Inbound Communication Type).

The following administration data and configuration must be in place before the conversion process can be run:

- Customer Care and Billing field activity types for customer event orchestration events, such as Enable Service, Disable Service, Back to Back, Cut for Non Payment and Reconnect for Payment have been created and configured on the Field Activity Type Profile. This is done by the Migrate Admin Data batch process (see above).
- Meter Data Management/Service Order Management service order activity types have been created and mapped to the corresponding Customer Care and Billing customer event orchestration field activity types in the Field Activity Type value of the C2M Mapping Field Values extendable lookup (accessible from the **Customer and Device Value Mapping** section of the Cross Product Configuration master configuration). This is done by the Migrate Admin Data batch process (see above).
- Customer Care and Billing field activity types for specific tasks (such as "Reread meter") are mapped to the corresponding field task types (such as "Read Meter") in the Field Activity Type value of the C2M Mapping Field Values extendable lookup (accessible from the **Customer and Device Value Mapping** section of the Cross Product Configuration master configuration).

Processing Notes

- The Customer Care and Billing field activity will be identified as either an orchestrator or specific field activity via its relationship to the service agreement/service point or a severance event.
- Existing Customer Care and Billing field activities in the "Pending" and "Completed" states will be recreated and used linked to service order activities. This is important for inflight field activities created using Customer Care and Billing field activity templates and steps, which are retained because they provide a link to the field work application.

Copyright © 2026, Oracle and/or its affiliates.

- When converting activities in the “Completed” state, only the service order activity or service order field activity will be created. The process does not create child activities or communications. For example, if converting a completed “Start Service” field activity, the result will be an “Enable Service” service order activity, but not its children service order field activities or outbound communications.
- The original Customer Care and Billing field activity will be captured on the corresponding service order activity and/or service order field activity to ensure a record of the original field activity that resulted in the creation of the new service order activity.
- The service order management communication business objects and algorithms have been updated to accommodate existing inflight Customer Care and Billing field activities. This includes providing for identifying outbound communications created based on the original Customer Care and Billing field activity, as well as receiving inbound communications and status updates based on the original Customer Care and Billing field activity.
- Customer Care and Billing field activities in the "Cancelled" and "Held" states will not be converted. If the batch process is submitted for Customer Care and Billing field activities in the "Held" state, the "Held" field activities will be deleted. Also, the conversion process will convert Customer Care and Billing field activities of types without a corresponding service order management activity type.

Conversion of field activities can be performed using a "staggered" approach by converting different type of field activities based on field activity status at different stage in the implementation. For instance, performing conversion of completed field activities first (though some implementations may opt not to migrate historical completed field activities at all), since these are static data, following by conversion of inflight field activities. In addition to status, field activity conversion may be staggered by field activity Id range or field activity Date range.

Data Conversion Batch Control

The Migrate Field Activity (X1-MIGFA) batch control performs processing for field activity conversion.

Field activities are converted based on their Status, and by date range. In addition, field activities to be converted for each batch run can be defined using the Low FA Id Override Value and High FA Id Override Value parameters. For instance, to convert a single activity, you can set both of these parameters to the same ID. If neither parameter is defined, the process will convert all field activities of the specified status for the specified date range.

This batch control uses the following parameters:

- **Low FA Id Override Value**: *Optional*. Used to specify an ID value to override the low ID in the thread ID range calculation. This, along with the High FA Id Override Value be used to limit the number of records processed by each batch run. If not specified, the process will convert all field activities whose ID is lower than the High FA Id Override Value parameter (if defined).
- **High FA Id Override Value**: *Optional*. Used to specify an ID value to override the high ID in the thread ID range calculation. This, along with the Low FA Id Override Value, can be used to limit the number of records processed by each batch run. If not specified, the process will convert all field activities whose ID is higher than the Low FA Id Override Value parameter (if defined)
- **FA Status (C/P/H)**: The status ("Completed", "Pending", or "Held") for which field activities will be converted. (Required)
- **Start FA Date**: The start date of the time range for which field activities will be converted. (Required. If not provided defaults to earliest date)

Copyright © 2026, Oracle and/or its affiliates.

- **End FA Date**: The start month of the time range for which field activities will be converted. (Required. If not provided defaults to the business date)

Examples:

To convert all Pending field activities for the month of July 2022, you would set the parameters as follows:

- FA Status (C/P/H): P
- Start FA Date: 2022-07-01
- End FA Date: 2022-07-31

To convert all Completed field activities for the months of April through June 2022, you would set the parameters as follows:

- FA Status (C/P/H): C
- Start FA Date: 2022-04-01
- End FA Date: 2022-06-30

To convert Pending field activities whose IDs range from 10001000 to 10009000 for the month of July 2022, you would set the parameters as follows:

- Low FA Id Override Value: 10001000
- High FA Id Override Value: 10009000
- FA Status (C/P/H): P
- Start FA Date: 2022-07-01
- End FA Date: 2022-07-31

To convert Completed field activities whose IDs range from 10001000 to 10009000 for the months of April through June 2022, you would set the parameters as follows:

- Register ID Low: 10001000
- Register ID High: 10009000
- FA Status (C/P/H): C
- Start FA Date: 2022-04-01
- End FA Date: 2022-06-30

Customer To Meter Table Setup Sequence

This section provides guidelines for setting up control tables and administrative data used with Oracle Utilities Customer To Meter.

- Customer Care and Billing Control Table Setup Sequence
- Meter Data Management Table Setup Sequence
- Operational Device Management Table Setup Sequence
- Understanding Online / Batch Loaders

Customer Care and Billing Control Table Setup Sequence

The topics in this section describe the order in which the control tables should be set up.

Copyright © 2026, Oracle and/or its affiliates.

- Core Control Table Setup Sequence
- Cross-Reference To The Remaining Chapters
- Open-Item Accounting Table Setup Sequence
- Fund Accounting Table Setup Sequence
- Payment Event Distribution Table Setup Sequence
- Loans Table Setup Sequence
- Quotes Table Setup Sequence
- Non-Billed Budget Table Setup Sequence
- Scripting Table Setup Sequence
- Reports Setup Sequence
- External Messages Setup
- Case Management Setup Sequence
- Umbrella Agreement Management Setup Sequence
- Outage Management Setup Sequence
- Prepaid Metering Setup Sequence
- Zone Set Up
- To Do Options Setup

Core Control Table Setup Sequence

To implement the system, you must set up your organization's business rules in "control tables". Setting up these tables is time-consuming because we allow you to tailor many aspects of the system to meet your organization's requirements. We strongly recommend that you take the time to document how you plan to set up all of these tables before you use the following roadmap to enter the core control data. Time spent understanding the interrelationships between this data will reap the rewards of a clean system that meets your current and long term needs.

While we describe the transactions and options in more detail in other sections of this manual, use the following chart (and the remaining sections of this chapter) as your roadmap. Here we list the order in which you perform tasks and the pages you'll use to set up your system. The order is important because some information must exist before other information can be defined (i.e., many dependencies exist).

> **Note**
>
> **Auto setup.** The Auto Setup column in the following table contains suggestions to save you time. It also indicates if a control table contains information when the system is installed.

> **Note**
>
> **You don't have to set up every control table.** You need only set up those control tables that govern functions that are applicable to your organization.

Copyright © 2026, Oracle and/or its affiliates.

Function	Menu	Auto Setup
Global Context		
Algorithm	Admin, Algorithm. You will need to set up an algorithm that populates global context values. The global context is used by various zones in the system to display relevant data. This algorithm is plugged-in on the installation record.	
Accounting Environment		
Country and State	Admin, Country	
Currency Codes	Admin, Currency Code	USD is automatically populated
Accounting Calendar	Admin, Accounting Calendar	
GL Division	Admin, General Ledger Division	
Security Environment		
Application Service	Admin, Application Service	All base package transactions are automatically populated
Security Type	Admin, Security Type	

Copyright © 2026, Oracle and/or its affiliates.

User Group	Admin, User Group Note, you won't be able to set up users at this point	One user group, ALL_SERVICES, is automatically setup. It references all other application services and a single user called SYSUSER.
Language	Admin, Language	ENG is automatically populated
Display Profile	Admin, Display Profile	Two display profiles are automatically setup: NORTHAM displays currencies and dates in a classic American format; EURO displays information in a classic European format
Data Access Role	Admin, Data Access Role	
Access Group	Admin, Access Group	

> **Note**
>
> The SYSUSER used is provided for use with on-premises implementations. It is disabled by default in cloud service implementations.

Copyright © 2026, Oracle and/or its affiliates.

User	Admin, User	SYSUSER is automatically set up. Note that you may import your users (and user groups) from an external source.
Return to User Group	You must return to your user groups and define all of their users	
Customer Class Environment		
Customer Class	Admin, Customer Class. At this point, you'll only be able to set up your customer class codes. You will return to these customer classes throughout the setup process to populate additional information.	
Financial Transaction Environment		
Work Calendar	Admin, Work Calendar	
CIS Division	Admin, CIS Division	
Revenue Class	Admin Revenue Class	
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs a distribution code's corresponding GL account when it is interfaced to the general ledger	
Distribution Code	Admin, Distribution Code	
Bank and Bank Accounts	Admin, Bank	
Billable Charge Template	Admin, Billable Charge Template. Note, if you want the system to default service quantities onto billable charges created using this template, you must setup the appropriate unit of measure code, time-of-use code and/or service quantity identifier.	
Billable Charge Upload Line Type	Admin, Billable Charge Line Type	
Algorithm	Admin, Algorithm. You will need to set up several algorithms. These algorithms: 1) retrieve a bill segment's consumption, 2) calculate a bill segment's bill lines, 3) construct a bill segment's financial transaction, 4) cancel previously estimated bill segments	
Bill Segment Type	Admin, Bill Segment Type	

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Bill Segment information that is displayed throughout the system for a specific Bill Segment Type. This algorithm is plugged-in on the Bill Segment Type.
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs a payment segment's financial transaction
Payment Segment Type	Admin, Payment Segment Type
Algorithm	Admin, Algorithm. You will need to set up the algorithm that constructs an adjustment's financial transaction
Algorithm	Admin, Algorithm. Several plug-in spots are available to perform additional logic when processing adjustments. For example, if you have the system calculate adjustments, you must set up an adjustment generation algorithm. Refer to Adjustment Type for other available plug-in spots that may be used by your implementation.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Adjustment information that is displayed throughout the system for a specific Adjustment Type. This algorithm is plugged-in on the Adjustment Type.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the Adjustment information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Adjustment Type	Admin, Adjustment Type
Adjustment Type Profile	Admin, Adjustment Type Profile

Copyright © 2026, Oracle and/or its affiliates.

Approval Profile	Admin, Approval Profile. Note, an approval profile references a To Do type and one or more To Do Roles; these must be set up before you can set up the approval profile. After the approval profile(s) are set up, they must be referenced on the adjustment types that they govern.
Cancel Reason - Bill	Admin, Bill Cancel Reason
Cancel Reason - Payment	Admin, Payment Cancel Reason
Cancel Reason - Adjustment	Admin, Adjustment Cancel Reason
Tender Type	Admin, Tender Type
Tender Source	Admin, Tender Source
A/P Request Type	Admin, A/P Request Type
Issuing Center	Admin, Issuing Center. You will need to set up issuing centers if your organization assigns document numbers to bills.
Installation	Admin, Installation Options - Framework and Admin, Installation Options. Many fields on the installation record impact the financial transaction environment. Refer to the description of the Billing and Financial Transaction tabs and the Messages tab in the Framework page for more information.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that distributes payments.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that handles overpayment situations.
Algorithm	Admin, Algorithm. You may need to set up an algorithm if specific customers can have individual bill due dates.
Algorithm	Admin, Algorithm. You may need to set up an algorithm if you want the system to delete bills that contain only information about historical payments.

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Admin, Algorithm. You may need to set up an algorithm if you want the system to levy a non-sufficient funds charge if a payment is canceled due to non-sufficient funds.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the bill information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the payment information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that defaults the amount when a payment is manually added. This algorithm also calculates the amount of an automatic payment for a bill for an account with an active auto pay option. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. Refer to Customer Class for other available plug-in spots that may be used by your implementation to perform additional logic when processing payments and bills.
Return to Customer Class	Admin, Customer Class. You will need to plug-in the algorithms defined above on your customer classes.
Budget Environment	
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: How To Calculated The Recommended Budget Amount, How To Periodically True Up A Customer's Budget Amount, The Circumstances When The System Should Highlight A Customer As Having An Anomalous Budget.
Budget Plan	Admin, Budget Plan

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Admin, Algorithm. Budget eligibility is set at the SA type level. You will need to set up an override budget eligibility algorithm if some service agreements for an SA type are not eligible for budget based on certain conditions.
Customer Environment	
Account Management Group	Admin, Account Management Group. Note, you will probably have to set up To Do Type and To Do Roles before you can setup account management groups. Refer to Assigning A To Do Role for more information on how account management groups may be used to define an entry's role.
Account Relationship	Admin, Account Relationship Type
Alert Type	Admin, Alert Type
Bill Message	Admin, Bill Message
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a bill in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your bill image software.
Bill Route Type	Admin, Bill Route Type
Contract Quantity Type	Admin, Contract Quantity Type
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a letter in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your letter image software.
Letter Template	Admin, Letter Template
Customer Contact Class	Admin, Customer Contact Class
Customer Contact Type	Admin, Customer Contact Type

Copyright © 2026, Oracle and/or its affiliates.

Conservation Programs	Admin, Conservation Program. You will need to set up conservation programs if your organization provides rebates to customers based on eligibility and verification of newly purchased appliances and hardware that are rated to conserve the demand for energy.
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if person ID's are in a predefined format.
Person Identifier Type	Admin, Person Identifier Type
SICs	Admin, SIC Code
Tax Exempt Type	Admin, Tax Exempt Type
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if phone numbers are in a predefined format.
Phone Type	Admin, Phone Type.
Person Relationship Type	Admin, Person Relationship Type.
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the person information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You will need to set up an algorithm to validate a person's name. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You can override the system's standard account information string by setting up an algorithm that produces this string of information. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a bill in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.

Copyright © 2026, Oracle and/or its affiliates.

Installation	Admin, Installation Options. Many fields on the installation record impact the Customer Environment. Refer to the description of the Main, Person, and Account tabs for more information.
Statements	
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a statement in a PDF (for the purpose of online display), you will need to create an algorithm that formats the extract records that are sent to your statement image software.
Statement Route Type	Admin, Statement Route Type
Statement Cycle	Admin, Statement Cycle
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a statement in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.
Automatic Payment (EFT) Environment	
Algorithm	Admin, Algorithm. You will need to set up an algorithm to create automatic payments. This algorithm is plugged-in on the installation record.
Tender Source	Admin, Tender Source Note: earlier, you created tender sources for the remittance processor and your cash drawers. At this point, you'll need to add at least one tender source for automatic payments. Why? Because automatic payments get linked to a tender control (which, in turn, gets linked to a tender source) when they are interfaced out of the system.
Algorithm	Admin, Algorithm. You will need to set up the appropriate automatic payment date calculation algorithm to populate the extract, GL interface and payment dates on automatic payments.

Copyright © 2026, Oracle and/or its affiliates.

Auto Pay Route Type	Admin, Auto Pay Route Type
Tender Type	Admin, Tender Type Note: earlier, you created tender types for things like cash, checks, etc. At this point, you'll need to add a tender type for each type of automatic payments (e.g., direct debt, credit card, etc.).
Work Calendar	Admin, Work Calendar. You need only set up additional work calendars if the auto pay sources (i.e., the financial institutions) have different working days than does your organization
Algorithm	Admin, Algorithm. If you need to validate the customer's bank account or credit card number, you will need to set up the appropriate validation algorithms.
Auto Pay Source Type	Admin, Auto Pay Source Type
Algorithm	Admin, Algorithm. You may need to set up an algorithm if your customers can define a maximum withdrawal limit on their autopay options.
Return to Customer Class	Admin, Customer Class. You should plug-in the Autopay Over Limit Algorithm in each appropriate customer class.
Deposit Environment	
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: The Definition Of A Good Customer, When To Refund A Deposit To A Customer, When To Recommend An Additional Deposit, How / When To Calculate Interest, How To Generate The Recommended Deposit Amount.
Deposit Class	Admin, Deposit Class
Non Cash Deposit Type	Admin, Non Cash Deposit Type
Field Work Environment	
Field Activity Type and Steps	Admin, Field Activity Type
Field Activity and Field Order Cancellation Reason	Admin, Fieldwork Cancel Reason
Field Activity Type Profiles	Admin, Field Activity Type Profile

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Field activities for Start/Stop will only be created if the SA Type linked to the service point has a SASP field work creation algorithm. Refer to the SA Type section below.
Algorithm	Admin, Algorithm. If you have software that's capable of reconstructing an image of a field order in a PDF for the purpose of online display, you will need to create an algorithm that renders this PDF. This algorithm is plugged-in on the installation record.

Credit & Collections Environment (if you collect on overdue bills (as opposed to overdue debt), you will NOT set up these tables; refer to Overdue Processing - Set Up Tasks for the list of control tables required to collect on overdue bills)

Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard collection events.
Collection Event Type	Admin, Collection Event Type
Algorithm	Admin, Algorithm. You may need to set up a collection process cancellation algorithm if your organization allows individual service agreements to be removed from a collection process if they are paid (rather than performing cancellation based on all SAs in a debt class).
Collection Process Template	Admin, Collection Process Template
Collection Class	Admin, Collection Class
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: Collection process cancellation criteria, Severance process cancellation criteria, and Override arrears due to pay plans.
Debt Class	Admin, Debt Class
Write Off Debt Class	Admin, Write Off Debt Class
Algorithm	Admin, Algorithm. You will need to set up Collection Condition algorithms.
Collection Class Control	Admin, Collection Class Control
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard severance events.

Copyright © 2026, Oracle and/or its affiliates.

Severance Event Type	Admin, Severance Event Type
Algorithm	Admin, Algorithm. You may need to set up a severance process cancellation algorithm if your organization allows a severance process to be canceled when the related service agreement is paid (rather than performing cancellation based on all SAs in a debt class).
Severance Process Template	Admin, Severance Process Template
Algorithm	Admin, Algorithm. You will need to set up several algorithms at this time: How to refer debt to a collection agency, How to transfer debt to another active service agreement, How to write down small amounts of debt, and How to refund credit balances to a customer.
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have non-standard write-off events.
Write Off Event Type	Admin, Write Off Event Type (Note, you'll have to wait until you have defined your SA Types before you can set up the Write Off Events because SA Type is a necessary parameter to write off debt).
Write Off Process Template	Admin, Write Off Process Template
Write Off Control	Admin, Write Off Control
Collection Agency	Admin, Collection Agency. Note, each collection agency references a person therefore you must set up a person for each agency before you can enter collection agency information.
Algorithm	Admin, Algorithm. You may need to set up algorithms if you have special logic that should be executed when a pay plan is canceled.
Pay Plan Type	Admin, Pay Plan Type
Payment Method	Admin, Payment Method
Third Party Payor	Admin, Third Party Payor. Note, you must create an account before you can create a third party payor.

Copyright © 2026, Oracle and/or its affiliates.

Installation	Admin, Installation. Several fields on the installation record impact the Credit & Collections Environment.	
Algorithm	Admin, Algorithm. You will need to setup an algorithm that's called when a user write-off debt real time.	
Return to Customer Class	Admin, Customer Class. You should plug-in the Autopay Over Limit Algorithm in each appropriate customer class.	
Services and Characteristics		
Service Type	Admin, Service Type	
Algorithm	Admin, Algorithm. If you have ad hoc characteristic types, you may need to set up the algorithms that control how they are validated	
Foreign Key Reference	Admin, FK Reference. If you have foreign key characteristic types, you may need to set up foreign key references to control how the user selects the characteristic values (and how the foreign key values are validated).	All base package FK references are automatically populated
Characteristic Type and Values	Admin, Characteristic Type	
Meter and Item Environment	Refer to the Meter Data Management Table Setup Sequence for control tables used with Meter Data Management, and configure the control tables as specified in this section.	
Premise and Service Point Environment		
Premise Type	Admin, Premise Type	
Algorithm	Admin, Algorithm. You will need to set up an algorithm to format the standard premise info that appears throughout the system. This algorithm is plugged-in on the installation record.	
Algorithm	Admin, Algorithm. You will need to set up an algorithm that formats the service point information that is displayed throughout the system. This algorithm is plugged-in on the installation record.	
Algorithm	Admin, Algorithm. You may need to set up the algorithms that determine if geographic ID's are in a predefined format.	
Geographic Type	Admin, Geographic Type	

Copyright © 2026, Oracle and/or its affiliates.

Service Point Type	Admin, SP Type. [Note - you won't be able to define the SP Type's SA Types until after you define the SA Types or the FA Type Profiles until after you define the Field Activity Type Profiles.]	
Bill and Service Cycle Environment		
Bill Cycle, Bill Cycle Schedule	Admin, Bill Cycle	
Bill Period, Bill Period Schedule	Admin, Bill Period	
Rate Environment		
Frequency	Admin, Frequency	
Service Quantity Identifier	Admin, Service Quantity Identifier	
Algorithm Type	Admin, Algorithm Type. If you create new pre-processing calculation rules, you must set up an algorithm type for each such rule (the algorithm type defines the types of parameters that are passed to the calculation rule).	All base package algorithm types are automatically populated
Bill Factor	Main, Rates, Bill Factor	
Algorithm Type	Admin, Algorithm Type. If you create new Register Rules you must set up an algorithm type for each such rule (the algorithm type defines the types of parameters that are passed to the register rule).	All base package algorithm types are automatically populated
Rate	Main, Rates, Rate Schedule	
Calculation Group	Main, Rates, Calculation Group	
Algorithm	Admin, Algorithm. If you use algorithms to dynamically change step boundaries, calculate prices, or implement calculation rule eligibility rules, you must set up these algorithms.	
Calculation Rule	Main, Rates, Calculation Rule	
Bill Factor Value	Main, Rates, Bill Factor Values	
Bill Factor Interval Values	Main, Rates, BF Interval Values	
Item Type SQ Estimate	Admin, Item Type SQ Estimate	
Degree Day	Admin, Degree Days	
Late Payment Environment		
Algorithm	Admin, Algorithm. You will need to set up the algorithm that determine if customers in a customer class are eligible for late payment charges	

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Admin, Algorithm. You will need to set up the algorithm that levies late payment charges for customers in a customer class
Return to Customer Class	Admin, Customer Class. You will need to plug-in the late payment charge algorithms set up above.
SA Configuration	
Algorithm	Admin, Algorithm. You will need to set up the algorithms that determine: • How to calculate the late payment charge amount for service agreements of a given type • Special criteria to be tested before a service agreement is severed. • How to create field activities for service agreements of a given type. • Special processing that should take place prior to the completion of a bill that references service agreements of a given type. • Special processing that should take place during completion of a bill that references service agreements of a given type. • Special processing that should take place when service agreements of a given type are created. • Special processing that should take place when a financial transaction is frozen for service agreements of a given type.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the SA information that is displayed throughout the system. This algorithm is plugged-in on the installation record.
Algorithm	Admin, Algorithm. You may want to set up an algorithm that formats the SA information that is displayed throughout the system for a specific SA Type. This algorithm is plugged-in on the SA Type.

Copyright © 2026, Oracle and/or its affiliates.

Algorithm	Admin, Algorithm. If you want a Control Central alert to highlight when the current account has any stopped service agreement(s), you will need to set up the algorithm that does this. This algorithm is plugged-in on the installation record.
Service Agreement Type	Admin, SA Type
Terms and Conditions	Admin, Terms and Conditions
SA Type Start Options	Admin, SA Type Start Option
Update SP Types with initial SA types and with FA Type Profiles	Admin, SP Type
SA Relationships	
SA Relationship Type	Admin, SA Relationship Type
Service Provider	Admin, Service Provider. Note, you must create a person before you can create a service provider. If you have financial relationships (you bill for them or they bill for you), you must also create an account and a financial settlement service agreement before you can create the service provider.
SA Type / SA Relationship	Admin, SA Type SA Relationship Type
Service Provider.	Admin, Service Provider. Note, you must create a person before you can create a service provider.
Sales and Marketing	
Order Hold Reason	Admin, Order Hold Reason
Order Cancel Reason	Admin, Order Cancel Reason
And more...	Refer to Campaign and Package Setup Sequence for additional setup requirements
Service Credit Membership	
Algorithm	Admin, Algorithm. You may need to set up algorithms for the service credit membership type and service credit event type to control behavior for the service credit membership and its events.
Credit Unit	Admin, Credit Unit. If your service credits record non-monetary units.
Service Credit Membership Type	Admin, Service Credit Membership Type

Copyright © 2026, Oracle and/or its affiliates.

Service Credit Event Type	Admin, Service Credit Event Type
Membership Inactive Reasons	Admin, SC Membership Inactive Reason
Wrap Up	
Algorithm	Admin, Algorithm. You will need to set up the algorithms that determine: #x2022; Special alerts on Control Central (assuming you have special alerts)
Installation Options	Admin, Installation Options - Framework and Admin, Installation Options. At this point, it's a good idea to double-check everything on the installation record.
Postal Default	Admin, Postal Code Default

If you have cash drawers you will also need to set up the following information:

- Create a person / account to which you will link your over / under service agreement. Refer How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under) for more information.
- Create a service agreement to which your over/under payments will be linked. This service agreement will reference your over / under SA type. Refer to Over / Under Cash Drawer Segmentation for more information.

If you upload payments from an external source (e.g., a remittance processor or lock box), you must set up the following information:

- Create a person and account to which the system will link payments with invalid account. Refer to Phase 3 - Create Payment Events, Tenders, Payments and Payment Segments for information about the process that books invalid payments to this account. Refer to How To Transfer A Payment From One Account To Another for how a user transfers the payment from the invalid account to the correct account (once known).
- Create a service agreement for this account. This service agreement will reference your payment suspense SA type. The system needs this service agreement so that it can distribute the invalid account's payment (and this is necessary so that cash will reflect the payment). Refer to Payment Upload Error Segmentation for more information.
- Update the tender source associated with the respective source of payments to indicate the service agreement created in the previous step should be used for payments with invalid accounts. Refer to Setting Up Tender Sources for more information.
- Because the payment upload process simply books payments that reference invalid accounts to the account associated with the suspense service agreement on the payment's tender source, this account should belong to a customer class with the appropriate payment distribution algorithms. This may entail creating a new customer class that will only be used on suspense accounts. This customer class would need the following algorithms:
- We'd recommend using a simple payment distribution algorithm like PYDIST-PPRTY (distribute payment based on SA type's payment priority).

Copyright © 2026, Oracle and/or its affiliates.

- We'd recommend using an overpayment distribution algorithm like OVRPY-PPRTY (distribute overpayment to highest priority SA type).

The remaining sections describe additional control tables that must be set up for specific functional areas.

Cross-Reference To The Remaining Chapters

The table in the previous section describes the order in which you should enter your control tables. These tables are described at length in the following chapters.

- Refer to Defining General Options Addendum and Defining General Framework Options for a discussion of the control tables associated with general functionality (e.g., country codes, state codes, etc.).
- Refer to Defining Financial Transaction Options for a discussion of the tables affecting your financial transactions (e.g., bill segment types, payment segment types, etc.)
- Refer to Defining Customer Options for a discussion of the control tables affecting persons, accounts and service agreements.
- Refer to Defining Service Order Options for a discussion of the control tables affecting fieldwork.
- Refer to Defining Credit and Collections Options for a discussion of the control tables affecting your collection activities.
- Refer to Defining Device Options for a discussion of the control tables affecting your meters and items.
- Refer to Defining Device Installation Options for a discussion of the control tables affecting your premises and service points.
- Refer to Defining Cycles and Schedules for a discussion of the control tables affecting your cyclical processes.
- Refer to Rates for a discussion of the control tables affecting your rates.
- Refer to Defining SA Type Options for a discussion of the control tables affecting your service agreement types.
- Refer to Defining Background Process for a discussion of the control tables affecting your background processes.
- Refer to Defining Algorithms for a discussion of the control tables affecting the algorithms referenced on many control tables.
- Refer to Defining SA Relationships for a discussion of the control tables affecting the relationships between service providers.
- Refer to Statements for a discussion of the tables affecting the statement setup options for your customers.
- Refer to Defining Service Credit Options for a discussion of the tables affecting the service credit membership setup options for your customers.

Open-Item Accounting Table Setup Sequence

Copyright © 2026, Oracle and/or its affiliates.

Open-item accounting tables need only be set up if your organization practices Open Item Accounting. Refer to Setting Up The System To Enable Open Item Accounting for a description of the tables that must be set up to enable this functionality.

Fund Accounting Table Setup Sequence

Fund accounting tables need only be set up if your organization practices Fund Accounting. Refer to Setting Up The System To Enable Fund Accounting for a description of the tables that must be set up to enable this functionality.

Payment Event Distribution Table Setup Sequence

Payment event distribution tables need only be set up if your organization opted to use the distribution rules method to create payment events. Refer to Setting Up The System To Use Distribution Rules for a description of the tables that must be set up to enable this functionality.

Loans Table Setup Sequence

Loans need only be set up if your organization offers loans to your customers. Refer to Setting Up The System To Enable Loans for a description of the tables that must be set up to enable this functionality.

Quotes Table Setup Sequence

Quotes need only be set up if your organization sends quotes to customers or prospects. Refer to Defining Quotation Options for a description of the tables that must be set up to enable this functionality.

Non-Billed Budget Table Setup Sequence

Non-Billed Budgets need only be set up if your organization allows your customers to pay set amounts at specified intervals (e.g. every two weeks). Refer to Setting Up The System To Enable Non-Billed Budgets for a description of the tables that must be set up to enable this functionality.

Scripting Table Setup Sequence

Scripts need only be set up if your organization opts to create scripts to step your users through business processes. Refer to Defining Script Options for information about scripting and the tables that must be set up to enable this functionality.

Reports Setup Sequence

Copyright © 2026, Oracle and/or its affiliates.

In order to use the reporting tool, you will need to set up reporting options. Refer to Configuring The System To Enable Reports for more information.

External Messages Setup

To configure the system to communicate with an external system, refer to to External Messages.

Case Management Setup Sequence

Case management options need only be set up if your organization uses cases to manage issues. Refer to Setting Up Case Types for more information.

Umbrella Agreement Management Setup Sequence

Umbrella agreement management options need only be set up if your organization uses umbrella agreements to manage contracts. Refer to the integration documentation for more information.

Outage Management Setup Sequence

Outage management options need only be set up if your organization interfaces with Oracle Utilities Network Management System. Refer to the integration documentation for more information.

Prepaid Metering Setup Sequence

Prepaid metering options need only be set up if your organization offers prepaid metering service to your customers. Refer to Defining Prepaid Metering Options for more information.

Zone Set Up

Most zones are delivered with the base-package and do not require any configuration. However, some zones are only available if configured by your implementation. Refer to Configuring Zones for more information.

To Do Options Setup

Refer to Setting Up To Do Options for more information on how to configure the system to match your organization's To Do management needs.

Copyright © 2026, Oracle and/or its affiliates.

Customer to Meter To Do Algorithm Configuration

Proper creation of To Do Entries when using Oracle Utilities Customer to Meter requires that the following algorithms be configured on the **Algorithms** tab of the **Installation Options - Framework** portal.

System Event	Sequence	Algorithm
To Do Pre-Creation	10	C1-TDCR-CHAR (Link Account/ Premise/Person/SA/SP to To Do
To Do Pre-Creation	20	D1-TDPRCRE (To Do Pre-Creation)
To Do Pre-Creation	30	F1-TDPRCRE(Customer to Meter To Do Pre-Creation)

Meter Data Management Table Setup Sequence

This section provides the suggested order for the setup of Meter Data Management administrative data.

Sequence	Entity	Description
1	Unit of Measure	Quantities measured and recorded by the system (CCF, KWH, KW, etc.).
1	Service Quantity Identifier	Used to further distinguish between measured quantities that have identical UOM/TOU combinations.
1	Time of Use	Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used (On-Peak, Off-Peak, etc.).
1	Factor	Centrally stored sets of values for use in validation rules, bill determinants calculations, and other processes.
1	Market	Defines jurisdictions or regulatory environments in which a Service Point participates.
1	Measurement Cycle	Defines the schedule for manual meter reading of devices at Service Points in that cycle
1	Measurement Cycle Schedule	Define the dates on which devices are scheduled to be read for a given measurement cycle.
1	Outbound Message	Defines messages sent to external systems.

Copyright © 2026, Oracle and/or its affiliates.

1	Processing Timetable Type	Defines types of schedules that can be referenced by different processes and objects.
2	External System	Defines External Systems with which Oracle Utilities Meter Data Management should be able to communicate.
3	Head End Systems External Applications Market Participants	External entities that serve various roles relative to the application (head-end systems, billing systems, market participants, outage management systems, etc.).
3	Contact Type	Defines properties of a class of entities (businesses, persons).
4	Activity Type	Defines properties common to a specific type of activity.
4	Communication Type	Define properties common to a specific type of communication.
4	Service Task Type	Defines specific types of tasks performed by external users (self-service meter reads, self-service outage notifications, etc.)
4	Dynamic Option Type	Defines information common to dynamic options of a specific type.
4	Manufacturer	Individual companies that makes devices. Manufacturers also reference models.
4	Exception Type	Defines properties common to VEE Exceptions of a specific type.
5	VEE Group	Collections of VEE Rules that are applied to initial measurement data.
6	VEE Rule	Standard and custom VEE Rules that perform checking and/or manipulation of initial measurement data.
7	VEE Eligibility Criteria	Dictates whether a VEE Rule can execute based on a set of defined criteria.
8	Measuring Component Type	Defines the most important properties of a measuring component.
8	Measuring Component Comparison Type	Defines details by which measuring component data can be compared to determine the days that most closely resemble a specific day being evaluated.

Copyright © 2026, Oracle and/or its affiliates.

9	Device Configuration Type	Defines properties of Device Configurations of a given type.
10	Device Type	Defines information about a class of devices. When setting up Device Types, device type codes should match corresponding asset type codes in Operational Device Management (see Operational Device Management Table Setup Sequence).
11	Service Point Type	Defines specific types of points at which service is delivered.
11	Service Point Quantity Type	Defines types of quantities that can be stored for a service point.
12	TOU Group	Used to limit the set of Time Of Uses that are usable in a TOU schedule.
13	TOU Map Template	Schedules used for TOU map data generation.
14	TOU Map Type	Define important properties of TOU maps of a given type.
15	Usage Transaction Exception Type	Defines properties common to Usage Transaction Exceptions of a specific type.
15	Usage Calculation Group	Collections of usage calculation rules that are applied to measurement data to calculate bill determinants for usage subscriptions.
16	Usage Calculation Rule	Defines rules that perform calculations on measurement data to generate bill determinants and other values used by external systems.
17	Usage Calculation Rule Eligibility Criteria	Dictates whether a usage calculation rule can execute based on a set of defined criteria.
18	Usage Subscription Type	Defines collections of properties common to a set of Usage Subscriptions.
19	Data Source	Defines the source of data for dynamic aggregation, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or measuring component sets
19	Dynamic Aggregation Measuring Component Type	Define the most important properties of a measuring component used with dynamic aggregation.

Copyright © 2026, Oracle and/or its affiliates.

19	Measuring Component Set	Define the dimensions and criteria by which dynamic aggregation will be performed.
19	Aggregation Group	Define the ordering of a series of related aggregations based on a set of configured measuring component Sets.
20	Processing Methods	Control various behaviors for external applications, head end systems, and market participants within the system such as which message is sent, how an external value is translated, among others.
20	Master Configuration	Configuration that applies to series of modules that acts as a central point of configuration rather than embedding repetitive configuration throughout a set of algorithms.
20	Information Lifecycle Management	Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle.
20	Analytics	Defines the extract parameters, the bucket configurations and configuration snapshots used for extracting data for Oracle Utilities Analytics

Operational Device Management Table Setup Sequence

This section provides the suggested order for the setup of Operational Device Management administrative data used with Asset Management features and functions.

> **Note**
>
> Asset Management features and functions are disabled by default. To enable these features, execute the Turn On ODM and C2M (D1-OnOdmC2m) Business Process Assistant (BPA) script.

Sequence	Entity	Description
1	Configuration Type	Defines the information required for various asset configurations with other assets, components, firmware, and other data.

Copyright © 2026, Oracle and/or its affiliates.

1	Service History Questions	Used to guide workers through corrective and preventive inspection processes as part of an asset's service history.
1	Service History Type	Defines the main attributes for service history of a certain type. Types might include maintenance, inspection, service, test, measurement, and so on.
2	Asset Location Type	Defines the details of asset or component locations of a certain type. Valid types of asset locations can include service points, other locations for installed assets, or any other location as defined by your organization.
2	Out of Service Location Type	Defines attributes of types of locations where assets that are out of service are stored and maintained. Examples of out of service locations include warehouses, storerooms, and repair shops.
2	Organization	Used for categorization and reporting for locations.
2	Configuration	Documents and manages valid setups for assets and components. This includes a specification for each asset or component in the configuration as well as the valid value for each parameter, such as the firmware version of each asset.
3	Out of Service Location	Defines physical locations where assets are stored or repaired such as a storeroom, warehouse, or service facility. When assets are changed to statuses such as "In Repair" or "Pending Disposition" the asset location is typically moved to an out of service location.
3	Configuration Reports	Provides a summary of currently installed configurations.

Copyright © 2026, Oracle and/or its affiliates.

4	Asset Type	Defines the attributes for assets and components of a certain type, including information such as the valid location types and service history types for the assets and whether or not such assets can have attached components. When setting up Asset Types, asset type codes should match corresponding device type codes in Meter Data Management (see Meter Data Management Table Setup Sequence).
5	Asset Activity Type	Defines the attributes for asset activities of a certain type, including scheduling information , exception handling, attachments, and related completion event types.

> **Note**
>
> Asset activity types are automatically created when Service Order Management Field Task Types are created (values for the Field Task Type extendable lookup).

Copyright © 2026, Oracle and/or its affiliates.

5	Asset Completion Event Type	Define various processes or events to be executed when completing an asset activity such as asset replication, service history creation, attachment creation, and activity creation.
5	Asset Inbound Communication Type	Defines the information required for the system to receive messages from external systems.

Understanding Online / Batch Loaders

Online / Batch Loaders enable the quick loading of data into the system. Data can be loaded for any business object as it uses the business object's schema to load data into the application.

Note: We recommend that this process not be used to load data for objects maintained through merged maintenance, such as Service Type, Unit of Measure, Service Quantity Identifier, Time of Use, Service Point Type, or Service Point as this process may not properly populate data for all maintenance objects supported by merged maintenance.

Two types of loaders are supported:

- **Online Loaders** easily upload smaller amounts of data. Use this type of upload process when uploading small numbers of records or data for business objects without complex lifecycles and business object rules such as Organizations. The data is loaded through the user interface to create the target objects.
- **Batch Loaders** should be used for larger volumes of data and for complex business objects such as Assets. Batch loaders load data from files on the server or attachments. Batch loaders use a two-step process that first creates the inbound sync request and uploads the file to a staging table and secondly, uploads the sync request from the staging table to either create the target objects or create another dedicated inbound sync request that handles the complexity of a particular object.

Processing uses the creation of a request type to facilitate the upload process:

- For online request types, the actual data is stored within and processed by the request.
- For batch request types, two options for batch load are supported; from a file located on the server or from an attachment in the application. Either the file name (file on server) or the attachment id (attachments) is stored on the request, the data is loaded into the generic inbound sync requests, and the request acts as a header record for all the generic inbound sync requests.

A csv file is the format used for loading data (see **Creating CSV Files**, below). The loader has very strict requirements as to the format of the file. Please see the embedded help in the **Data** section on the Online Loader Request business object for details and additional tips.

- Creating CSV Files
- Initiating Requests

Creating CSV Files

While data can be manually entered into the system, the most efficient method for uploading the data is to use one of the CSV data templates already available for all appropriate business objects.

Copyright © 2026, Oracle and/or its affiliates.

The first step is to create a .csv file (comma delimited) from the business object of the portal being loaded.

The following is an example using Organization (W1-Organization):

1. Navigate to **Admin**, select **Location**, select **Organization**, and then select **Search**.
2. Search for and select a record.
3. Expand the **Record Information** section and click the "Organization" **Business Object** link to navigate to the **Business Object** portal.
4. Click the **View Schema** link on the **Main** tab and then copy the elements from the business object to a CSV file to create your data.

1	Sequence	Primary Key	Field 01	Field 02	Field 03
2	1	Organization	Organization Type	Disposition	description
3	2	nodeId	nodeType	nodeDisposition	description
4	3		Group	W1AC	Group 1
5	4		Group	W1AC	Group 2

Notes:

- Columns in the top row match the elements in the list of the request business object (header).
- The row with the lowest sequence number (1) is the description of xpaths to help users read the file and is ignored by the processing.
- The business object processing treats the row with the second lowest sequence number (2) not as data, but as the xpath to move data to the target business object, in this case, W1-Organization.
  - Note that the configuration of some business object mean that this cell may start with a “+” sign. Also note that the file always treats this cell as a formula and adds an “=” sign to the front of the data string, as indicated by “#NAME?” (For example: =+nodeId/ nodeType). You must reformat the cell to “Text” and remove the “=” sign.
- Rows with sequence number three (3) and up are treated as data.
- For objects where the primary key is a system generated ID (such as the case with Organization), the Primary Key column cells blank.

Note: When adding records that reference look up values use the lookup field value, not the description. For instance, the Disposition field references the NODE_DPOS_FLG lookup, for which the "Active" field value is "W1AC". For fields that reference other objects, use the code, not the description.

Other Formatting Issues

The following information addresses known formatting issues in working with the CSV data template:

- Dates need to be formatted as yyyy-mm-dd (2020-10-27).
- Date and Time should be formatted as yyyy-mm-dd-hh.mm.ss.
- Columns based on lookup values that have leading zeroes will need to match the values in the CSV file.

Copyright © 2026, Oracle and/or its affiliates.

- A link needs to be created when you want to include the “parent”, which has a system generated ID, in the same file with the “child”. For example, if you want to populate the Parent Organization’s ID in the same file, you would need to insert a link (for example, @3, which is the @ symbol plus the Sequence #) in the row where the system generated ID would reside.
- If multiple rows contain the same Primary Key, only the last row is loaded. All values for the Primary Key need to be on the same row.

Initiating Requests

There are two steps involved in loading data using this process: creating the request type and the request.

Creating Request Types

1. Navigate to **Admin**, select **General**, select **Request Type**, and then select **Add**. The **Select Business Object** screen opens.
2. Select the request type business object that corresponds to your request type (“Online Loader Request Type” or “Batch Loader Request Type”) from the **Request Type Business Object** drop-down list and click **OK**. The **Online Request Type** or **Batch Request Type** screen opens (based on the business object selected).
3. Enter a code and meaningful description for the request type in the **Request Type** and **Description** fields. For example, "Online Loader Request Type" or "Batch Loader Request Type".
4. Enter a **Detailed Description** (optional).
5. For request types based on the Batch Loader Request Type business object, specify whether or not the contents from the attachment record will be deleted once the upload is finished (Yes or No) in the **Deleted Attachment Upon Finish** drop-down list.
6. Click **Save**.

Creating Requests

Online Requests

1. Navigate to **Menu**, select **Tools**, select **Request**, and then select **Add**. The **Select Type** screen opens.
2. Select your online loader request type and then click **OK**. The **Online Loader Request** screen opens.
3. Select the business object for your loader in the **Business Object** field.
4. Select your CSV file to upload or enter the data directly into the Data section and then click **Save**. Note: If examples of online requests are already available in your system, search for the appropriate request, click **Edit**, and either use the existing data if it applies, or click **CSV File To Upload** to upload your already formatted and populated CSV file. You can also click **Download To CSV File** and manually populate the data into the file and then click **CSV File To Upload** to upload the file. Your request will open in the **Request** zone in the "Created" state.
5. Click **Load Data** in the **Record Actions** section to submit the request. If successful, the status of the request will be "Loaded". Check the **Request Log** for requests that do not transition to a “Loaded” status. You will need to click **Back to Created** to reset the request back to the "Created" status and click **Edit** to fix the issue before resubmitting the request.

Batch Requests

Copyright © 2026, Oracle and/or its affiliates.

1. Navigate to **Menu**, select **Tools**, select **Request**, and then select **Add**. The **Select Type** screen opens.
2. Select your batch loader request type and then click **OK**. The **Batch Loader Request** screen opens.
3. Select an external system from the **External System** drop-down list.
4. Select the business object for your loader in the **Business Object** field.
5. Enter a commit frequency in the **Commit Frequency** field.
6. Enter the **File Name** or search for and select the **Attachment** that should be in the request and click **Save**.

- For **Batch File Upload** — The Filename is specified in the request. Comma delimited files should uploaded to or available on the server. Run the W1-GBUPL batch job to create the sync request. Parameter File Name and File Path should match the file name and path of the file. Once the job has ended and the request is loaded, run the F1-SYSRQ batch job to process the created sync request.
- For **Batch Attachment Upload** — The Attachment ID is specified in the request. Comma delimited file is uploaded as an attachment. Run the W1-GAUPL batch job to create the sync request. Parameter Request ID should match the created request. Once the job has ended and the request is loaded, run the F1-SYSRQ batch job to process the created sync request.
- Use the following parameters when running the F1–SYSRQ batch job:
  - Maintenance Object: F1–SYNCREQIN
  - Restrict by Batch Code: True
  - Restrict by Business Object: W1–GenericDataLoaderSync
  - Restrict by Status Code: PENDING

Customer To Meter Batch Processing

This section provides guidelines for configuring batch processing with Oracle Utilities Customer To Meter.

- Batch Process Dependencies

Batch Process Dependencies

The contents of this section illustrate the periodicity and dependencies between the various background processes described above.

- Batch Schedulers and Return Codes
- The Nightly Processes
- The Ongoing or Hourly Processes
- The Letter Processes
- The Periodic Processes

Copyright © 2026, Oracle and/or its affiliates.

Batch Schedulers and Return Codes

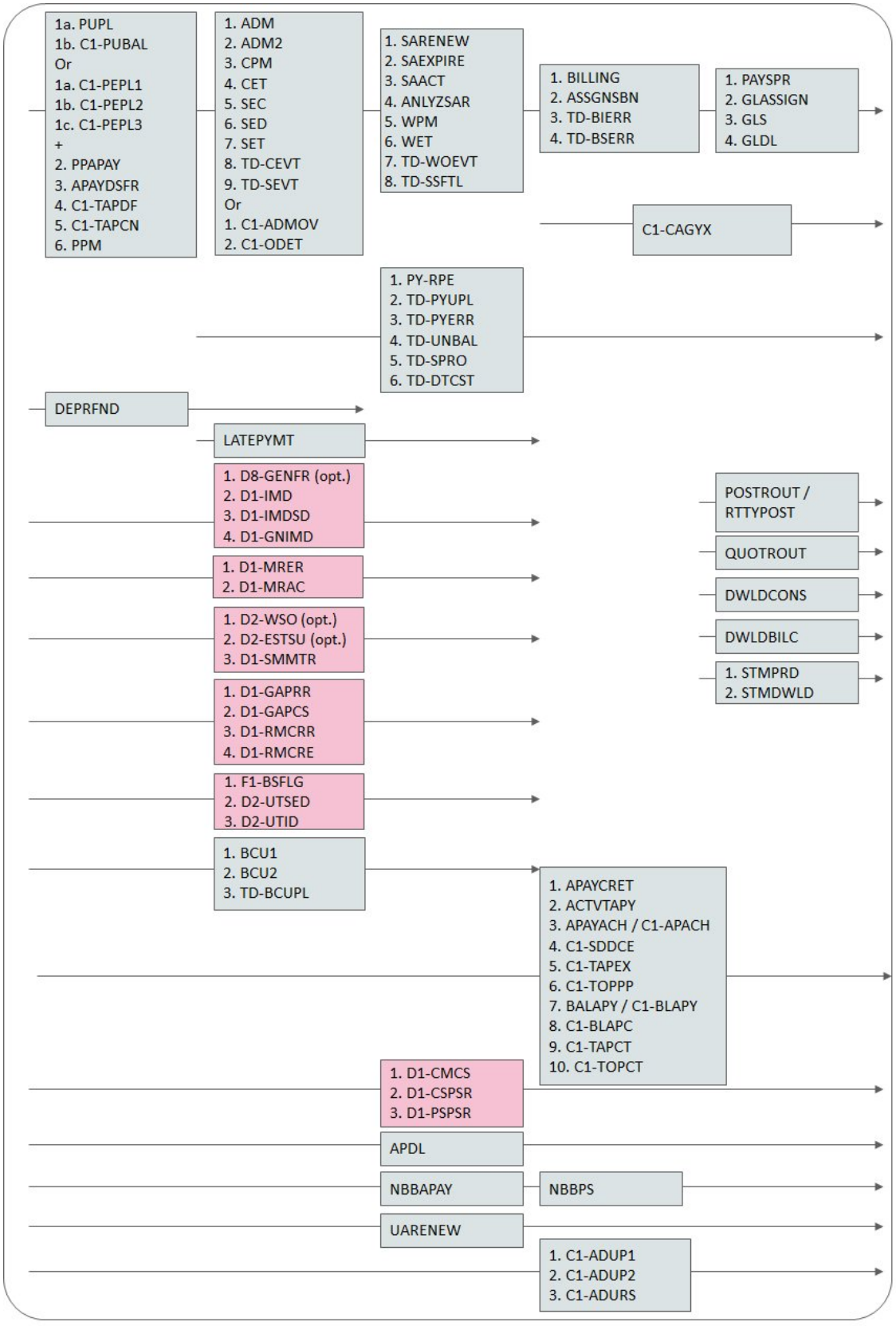
If you use a batch scheduler (e.g., Control-M, Tivoli) to control the execution of your batch processes, it will be interested in the possible values of each process's return code. The return code is a number that indicates if the process ended successfully. All product processes will return one of the following return code values:

- 0 (zero). A value of zero means the batch process ended normally.
- 2. A value of 2 means the batch process detected a fatal error and aborted.

The Nightly Processes

The following diagram illustrates the dependencies between the batch processes.

Copyright © 2026, Oracle and/or its affiliates.

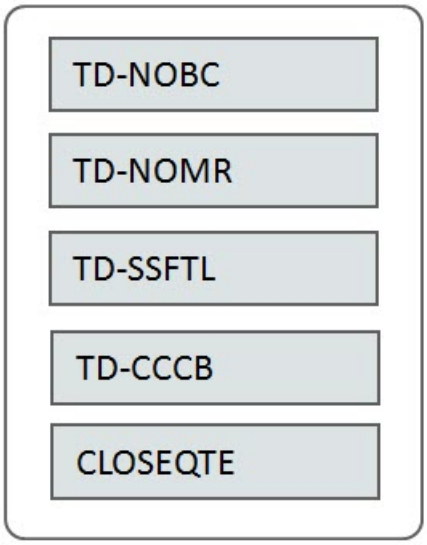


The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially. When multiple boxes exist on a timeline, all processes in an earlier box must execute before the subsequent

Copyright © 2026, Oracle and/or its affiliates.

box is executed. Those timelines that appear beneath the Main Job Stream's timeline indicate when the timeline's respective processes can be executed in respect of the Main Job Stream.

The following diagram illustrates the daily batch processes for which there are no dependencies.



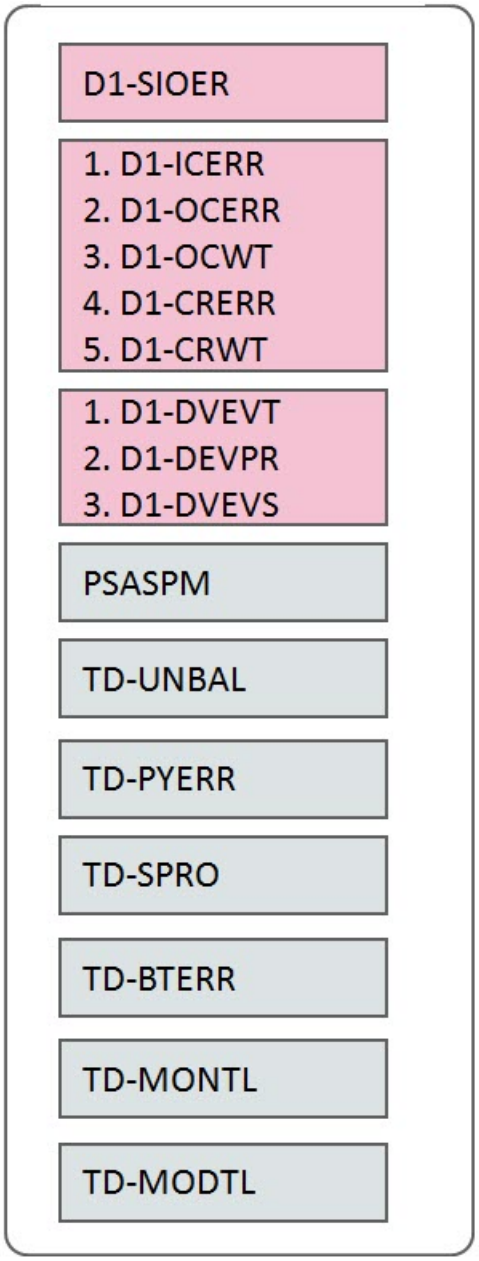
The mnemonics in the boxes refer to the individual batch processes described above.

> **Note**
>
> **No dependencies exist.** As you can see, there are no dependencies between the boxes (meaning they may be run in parallel).

The Ongoing or Hourly Processes

The following diagram illustrates the dependencies between the ongoing or hourly batch processes.

Copyright © 2026, Oracle and/or its affiliates.



The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially.

> **Note**
>
> **No dependencies exist.** As you can see, there are no dependencies between the boxes (meaning they may be run in parallel).

Throughout the course of the day, you will likely want to run jobs to bring measurement and event data in from the various metering / head-end systems and other external systems. To accompany this, you should consider running the following batch processes:

- **Ongoing Master Data Sync Processing**

Copyright © 2026, Oracle and/or its affiliates.

With the Oracle Utilities Customer to Meter deployment, data synchronization is used to keep certain objects synchronized. Refer to the C2MIntegration section for further details. Cleaning out synch staging records in error by running D1-SIOER may be necessary.

- **Command Processing**

  If using Oracle Utilities Smart Grid Gateway to process device commands, it's important to keep communications flowing to and from smart meters to provide the most accurate picture of the state of a given meter. This would include:

  – Retrying inbound communications in error (D1-ICERR)

  – Retrying outbound communications in error (D1-OCERR)

  – Processing outbound communications waiting for a response (D1-OCWT) to see if they should be timed out.

  – Processing command request activities in error (D1-CRERR) and those that are waiting (D1-CRWT)

  > **Note**
  >
  > Base package business objects for communications and command activities are designed to trap any processing errors encountered and transition the object into an Error state. To deal with unexpected errors that can't be trapped, which could leave communications / command activities in unmonitored states, implementations can choose to configure their own batch controls based on the delivered D1-ACTVY and D1-COMM batch controls, restricting records processed by business object or maintenance object as needed

- **Initial Measurement Processing**

  For initial measurement processing, the following batch processes should be scheduled on an ongoing basis per business requirements:

  – Processing initial measurements in the pending state (D1-IMD).

  – Processing initial measurements seeders in error (D1-IMDSD).

  – Processing initial measurements in the exception state (D1-GNIMD).

  These processes are also highlighted in the Nightly Processes stream to reflect the need to bring in as much measurement data as possible prior to running billing, both up-to-date data from late-reporting meters and corrected readings from the head-end system.

- Event Processing

  The base package is configured such that device events are processed immediately upon receipt, since they might need to be sent to some other application such as an outage system. This can be changed by configuring a monitor process on the device event business object to stop records in a specified state, and then use a batch process to process the events all at once. Beyond this type of batch-oriented processing for events, other event processing could include:

  – Re-processing device event "seeders" in error (D1-DVEVS)

  – Processing all device events that can come in as pairs (D1-DEVPR)

  – Picking up device events for processing if they've stopped in any state (D1-DVEVT)

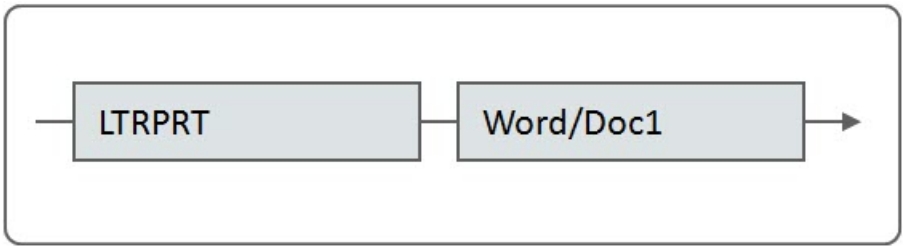
  If device events are configured to be held from being sent onto downstream applications, such as to prevent "flicker" outage events (an outage event and a restore event received in rapid succession) from being sent, device event monitoring (D1-DEVPR and D1-DVEVT) should be set up to be run periodically to ensure timely transmission of events.

Copyright © 2026, Oracle and/or its affiliates.

The Letter Processes

To extract information for your various letters, only one background process, LTRPRT, is required regardless of the different types of letters you have. This process simply calls an algorithm plugged-in on the respective letter template to construct its flat-file content.

The following diagram illustrates the dependencies for the letter background process. While this process should be run at least on a daily basis, you may want to consider running it more frequently (depending on how frequently you produce letters).



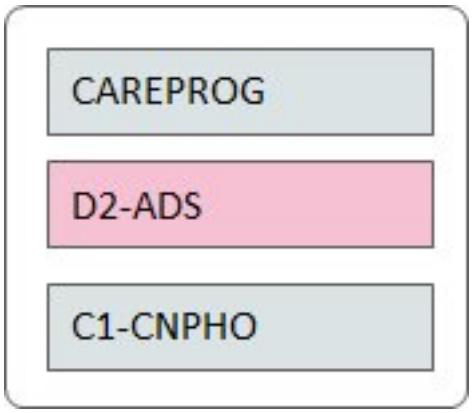
The mnemonics in the boxes refer to the individual batch processes described above. When a box contains multiple processes, these processes must be run sequentially. When multiple boxes exist on a timeline, all processes in an earlier box must execute before the subsequent box is executed.

The Periodic Processes

The following diagrams illustrate the dependencies between the periodic background processes. You may want to consider running these processes more or less frequently than suggested (depending on business requirements).

Periodic Processes: Weekly

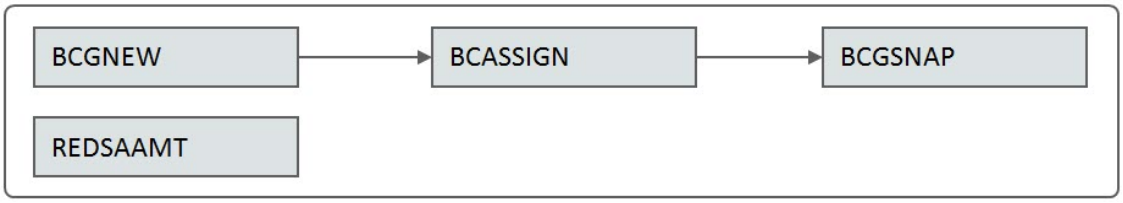
The following processes should be run roughly once a week, or as per business requirements.



Periodic Processes: Monthly

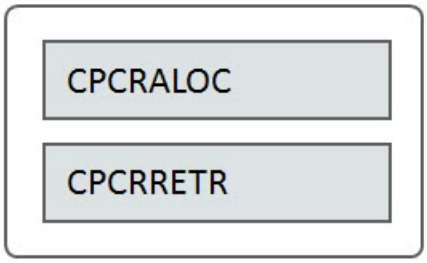
The following processes should be run at least once a month, or as per business requirements.

Copyright © 2026, Oracle and/or its affiliates.



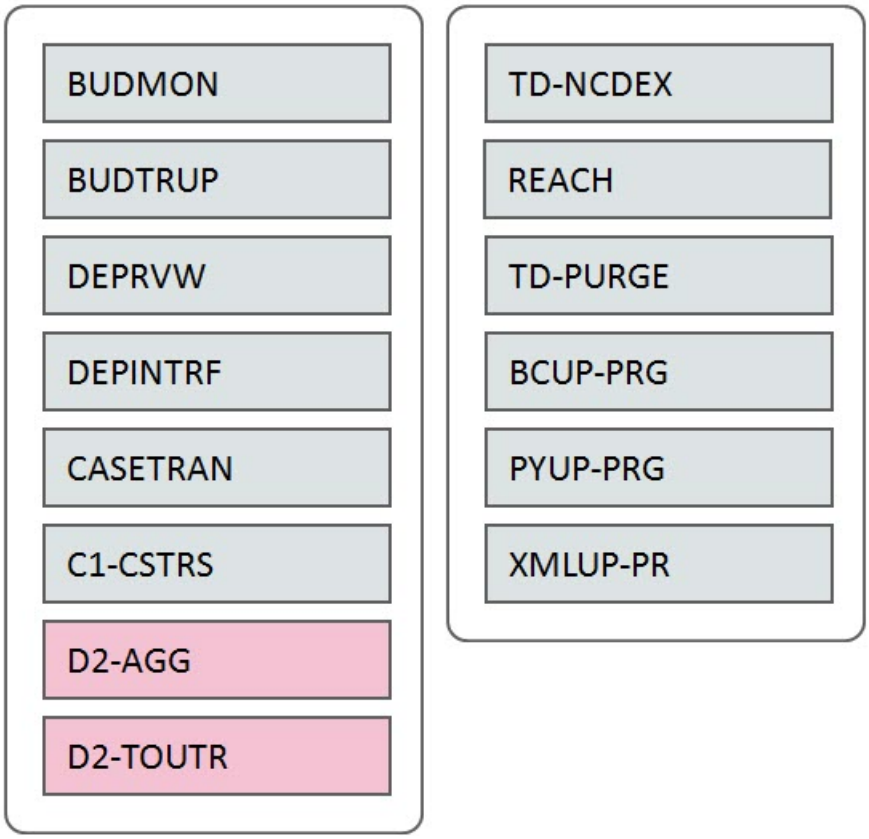
Periodic Processes: Annually

The following processes should typically be run once a year, or as per business requirements.



Periodic Processes: Per Business Requirements

The following processes should be run as frequently as dictated by business requirements.



The mnemonics in the boxes refer to the individual batch processes described above.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Few dependencies exist.** As you can see, there are few dependencies between the boxes (meaning they may be run in parallel).

TOU Map Data Generation

TOU map data must be in place for all TOU maps used in usage calculations. Generation of this data is performed using the “Time of Use Map Data Generation Monitor” batch control (D1-TOUTR). This process can be performed for long time periods, such as a year, generating data for all time-of-use maps for the entire following year, or it could be done more frequently, such as whenever schedules are updated via the TOU map templates

Defining General Options

This section describes control tables that are used throughout Oracle Utilities Customer Care and Billing.

- Defining Installation Options
- Defining Customer Languages
- Defining Accounting Calendars
- Defining General Ledger Divisions
- Defining Banks & Bank Accounts
- Defining Issuing Centers
- Defining Service Tasks Options
- Defining Customer Questions
- Agent Contact Scripting

Defining Installation Options

The topics in this section describe the various installation options that control various aspects of the system that are specific to the Oracle Utilities Customer Care and Billing product.

> **NOT_SUPPORTED**
>
> Refer to Installation Options - Framework for options that are common to products on the same framework.

- Installation Options - Main
- Installation Options - Person
- Installation Options - Account

Copyright © 2026, Oracle and/or its affiliates.

- Installation Options - Billing
- Installation Options - C&C
- Installation Options - Financial Transaction
- Installation Options Framework - Algorithms

Installation Options - Main

Select **Admin**, and then **General**, and then **Installation Options** and use the **Main** page to define system-wide installation options.

Description of Page

Use **Quick Add Tender Type** to define the tender type defaulted on payments added using the Payment Quick Add transaction.

Use **Starting Balance Tender Type** to define the tender type of the starting balance recorded on your tender controls (this will almost always be the tender type associated with "cash"). This value is used during tender control balancing as a separate balance is required for each tender type in order to balance a tender control. Refer to The Lifecycle Of A Tender Control for more information.

> **NOT_SUPPORTED**
>
> For more information, refer to Setting Up Tender Types.

Turn on the **Create Field Activity Start Stop** if field activities should be created when a start or stop is recorded (as opposed to shortly before the start/stop date). You might want to turn this switch off if it's possible for the state of the service point (or its meter / item) to change between the time service is requested and the actual service date. Why? Because the state of the service point and the state of its meter / item affects the type of field activity that is created. For example, if a customer wants to start service and there is no meter at the metered service point, an "install meter" field activity is created. However, if by the time the install date comes around, a meter has been installed by some other means; this field activity is inappropriate. This is why you might want to setup the system to wait until shortly before the service date to create the field activity (i.e., it reduces the likelihood that an inappropriate field activity is created). Refer to Starting Service and Field Activities for more information.

If you use orders to create new customers, define the **Campaign** that should be defaulted on orders created when the order transaction is opened for a new customer. Refer to Real time Marketing of Services to a New Customer for more information.

Use the **Premise Geo Type** to indicate whether at least one geographic identifier (e.g., GPS coordinate) is Required or Optional on a premise. Refer to Defining Geographic Types for more information.

The **Alternate Representation** flag should be set to None unless your organization uses multiple character sets for a person's main name and / or a premise's address. Alternate representations are typically only used in countries where multiple character sets are used. For example,

- In Hong Kong, a person's name may be written in both Chinese characters and in English.
- In Japan, a person's name may be written in both Kanji and Katakana.

Copyright © 2026, Oracle and/or its affiliates.

In both of the above situations, users need to be able to use both representations to find a customer or a premise.

> **Note**
>
> **Spouses.** If your organization doesn't use multiple character sets, you might want to consider using this functionality for spousal relationships. For example, rather than setup a person for each party in a spousal relationship, you could simply define one party using the person's main name and the spouse using the alternate name. While this is a bit of a "hack", it might be sufficient for your implementation as it will be much easier for an end user to use.
>
> **Alerts that should appear adjacent to a person's name or address.** If your organization doesn't use multiple character sets, you might want to consider using this functionality to implement critical person or premise alerts. For example, if you have a customer who's supported by a specific account representative, you could enter the account rep's name as the person's alternate name. If you do this, the account rep's name would appear in parenthesis following the customer's name. In addition, you can search for the customers supported by the account rep on Control Central by entering the account rep's name. This is a bit of a "hack", but it might prove useful for a variety of functions.

If your organization uses alternate representations of person name or address, set this flag to one of the following values:

- Use a value of Address if you only use alternate representations for premise addresses.
- Use a value of Name if you only use alternate representations for a person's primary names.
- Use a value of Name & Address if you use alternate representations for both premise addresses and person names.

The following points describe the ramifications of this flag in the system:

- If you support alternate representations of a person's primary name,
- The name grid on Person - Main allows you to specify an Alternate name for the person.
- If you use the base package name formatting algorithm, a person's name will be shown throughout most of the system in the format AAA (BBB), where AAA is the person's primary name and BBB is the person's alternate name. Note, this format does not apply to names that appear in search results (i.e., the alternate name is not concatenated to the main name in search results; however you can search for information using the alternate name).
- Most of the system's person name-oriented searches will allow users to use both a person's primary and alternate names to search for information.
- If you support alternate representations of a premise's address,
- A new tab is available on the Premise page that allows a user to define an alternate address for a premise.
- If you use the base package address formatting algorithm, a premise's address will be shown throughout most of the system in the format AAA (BBB), where AAA is the premise's primary address and BBB is the premise's alternate address.
- Most of the system's premise-oriented searches will allow users to use both a premise's primary and alternate addresses to search for information.

Copyright © 2026, Oracle and/or its affiliates.

If the CTI Integration flag has been enabled, set the **CTI Integration** flag to Yes if your organization integrates with an external computer telephony integration (CTI) system that supports a "get next caller in the queue" function.

The **Control By CIS Division** flag affects system behavior in many areas. This flag should be set if your business operates in different jurisdictions and CIS Division is used to delineate accounts and premises that users can access. When this flag is set, all users must be associated with at least one CIS division. See Users and CIS Division for more information.

The following system behaviors are affected by this flag:

- On Person, CIS Division is presented and is required when adding an account. Customer class is defaulted after CIS Division is set.
- On Order, CIS Division is presented when a new account is being added and is required. Customer class is defaulted after CIS Division is set..
- On Person, Order, Account and Premise, the list of CIS Divisions that can be chosen is restricted to the CIS Divisions associated with the user. The user's primary CIS Division is defaulted.
- On Account, CIS Division is presented before customer class. Customer class is defaulted after CIS Division is set.

The **Premise/SP Search** flag is used to provide access security to premises and service points that are not linked to an account.

- When the flag is set to Not Restricted, access to premises and service points that are not linked to an account is not secured.
- When the flag is set to Restricted by CIS Division, access to premises and service points that are not linked to an account is secured by the user's CIS Divisions. If the premise has a CIS Division, then only users that have access to that CIS division can access that premise or its service points. See Users and CIS Division for more information. Premise and service points belonging to premises without a CIS Division are not secured. A Premise Type setting can control whether premises of that type require a CIS Division.

> **Note**
>
> Premises and service points that are linked to an account are secured by the account access group

> **Warning**
>
> In order to improve response times, installation options are cached the first time they are used after a web server is started. If you change this field's option and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. Refer to Caching Overview **for information** on how to clear the system login cache (this is the cache in which installation options are stored).

Installation Options - Person

Copyright © 2026, Oracle and/or its affiliates.

Select **Admin**, and then **General**, and then **Installation Options** and use the **Person** page to define person-specific installation options.

Description of Page

Use the **Person ID Usage** to indicate whether or not at least one form of identification is Required or Optional when a new person is added. If required, indicate when the identifier is required based on the person's association with account(s).

Each form of identification has an identifier type. For persons that are humans (as defined by the person type), the system defaults the identifier type defined in **Identifier Type (Person)**. For persons that are businesses (as defined by the person type), the system defaults the identifier type defined in **Identifier Type (Business)**.

The **Use Name Separator** and **Person Name Separator Character** are used when formatting person names for persons (not businesses). The person name format algorithm type (NMFM-VALFMT) that is provided with the system uses these settings.

Installation Options - Account

Select **Admin**, and then **General**, and then **Installation Options** and use the **Account** page to define account-specific installation options.

Description of Page

When a new account is added, the system requires it have a customer class. If the main customer linked to the account is a human (as defined by the customer's person type), the system defaults the customer class defined in **Customer Class (Person)**. For persons that are businesses (as defined by the person type), the system defaults the customer class defined in **Customer Class (Business)**. For more information, refer to Setting Up Customer Classes.

In addition to requiring a customer class when a new customer is added, the system also requires a "main customer" (i.e., a reference to a person who is identified as the main customer for the account). Enter the default **Account Relationship Type Code** to be used to define the main customer relationship. For more information, refer to Setting Up Account Relationship Codes.

Enter the default **Bill Route Type** to be used to define how bills should be routed to a customer. For more information, refer to Setting Up Bill Route Types.

Enter the default **Quote Route Type** to be used to define how quotes should be routed to a customer. For more information, refer to Setting Up Quote Route Types .

If the number of pending start and pending stop service agreements exceeds the **Start Stop Detail Threshold** for an account, it is considered a large account for start stop purposes. Refer to Start/Stop Maintenance for more information.

Installation Options - Billing

Select **Admin**, and then **General**, and then **Installation Options** and use the **Billing** page to define billing-specific installation options.

Description of Page

The **Bill Segment Freeze Option** controls when a service agreement's balance and the general ledger is affected by bill segments and certain types of adjustments. Refer to

Copyright © 2026, Oracle and/or its affiliates.

Preventing SA Balances And The GL From Being Impacted Until Bill Completion to understand the significance of this option.

The **Accounting Date Freeze Option** controls how the accounting date defined on financial transactions is populated. Refer to Forcing The Freeze Date To Be Used As The Accounting Date to understand the significance of this option.

Define the **Rollover Threshold Factor** used by billing to determine if a register's consumption is sensible. This value is used as follows:

- Whenever billing calculates a meter's register's consumption, it compares it to a value equal to X times the register's maximum capacity (where X is the Rollover Threshold Factor).
- If consumption exceeds this value, a bill segment error is generated. If this consumptive value is correct, a user will need to override the consumption value billed on the bill segment (billing will never use such a read).

Define the **Minimum Amount for Final Bill**. If a final bill is less than this amount, the bill is still produced; it's just not printed.

Typically, the system sets a bill's Bill Date equal to the date on which it is completed. If you want to be able to specify a bill's Bill Date when you complete a bill, turn on **User Can Override Bill Date**. You would only want to override the bill date if you are setting up sample bills from historical period whose bill date needs to reflect the respective historical period.

Turn on **Use High Low Failures on Bill** if the system should mark meter reads that fail high / low checks as billable. Turn off this switch if such reads should not be used by billing. Users may override this default value on a specific read. Refer to Validations for more information.

Base Time is used by billing algorithms to determine the effective start and end times for a given period. The **Start Day Option** further defines how to use the base time, indicating whether the base time is for the Current Day or for the Previous Day.

Turn on **Use Alternative Bill ID** if your implementation uses assigned document numbers or sequential bill numbers. In the **Alternative Bill ID Option** list:

- Select **Document Numbers** if you require a system-assigned document number for each bill in addition to the Bill Id, which is a system-assigned random number used as the bill's primary identifier. Refer to Document Numbers for more information.
- Select **Sequential Bill Numbers** if you require a system-assigned unique sequential number for each bill in addition to the Bill Id, which is a system-assigned random number used as the bill's primary identifier. Refer to Sequential Bill Numbers for more information

> **Note**
>
> **Document Number Algorithms.** In addition to turning on **Use Alternative Bill ID** and specifying the **Alternative Bill ID Option**, the Document Number and Document Number Details algorithms must be enabled on the Installation record. These algorithms contain the logic used by the system to assign a document number to a bill.

The **Bill Correction** option lets you control whether your implementation uses Credit Notes or Correction Notes. Select the **Credit Note** option if you require bill segment cancellation details to be presented to the customer on a separate bill (referred to as a credit note). Refer to Credit Notes for more information. Select the **Correction Note** option if you require bill segment cancellation details and bill segment rebill details to be presented to the customer on a separate bill (referred to as a correction note). Refer to Correction Notes for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Credit Notes or Correction Notes.** The Bill Correction option on the Installation record controls whether Credit Notes or Correction Notes are allowed. If your implementation uses Correction Notes, the override label on the following should be customized accordingly:
>
> Lookup value CRNT on the customizable lookup field TXN_FLTR_TYPE_FLG (this lookup value is used on the Match Event Page and Account Bill History transactions)
>
> Lookup value CR on the customizable lookup field PYCAN_SYS_DFLT_FLG (this lookup value is used on the Pay Cancel Reason transaction)
>
> Metadata field CR_NOTE_FR_BILL_ID (this field is used on the Bill Search Page)

The **Autopay Creation Option** controls when automatic payments are created, distributed, and frozen. This option allows you to control when automatic payments will affect customer's balances and when their financial impact affects the general ledger. Refer to How And When Are Automatic Payments Created to understand the significance of this option.

Installation Options - C&C

Select **Admin**, and then **General**, and then **Installation Options** and use the **C&C** page to define credit and collections-specific installation options.

Description of Page

When you look at an account or service agreement's debt, the system shows the respective age of each piece of outstanding debt. The **Oldest Bucket Age (Days)** defines the debt age after which the system groups all outstanding debt together. For example, if this field is 180:

- The exact age of each element of debt that is less than 180 days old would be shown as a separate line item in the aged debt information.
- All debt older than 180 days would be amalgamated into a single "bucket".

Oldest Bucket Age (Days) also has another use – it defines the age of financial transactions that are considered by the background processes that mark old debt as "redundant". These batch processes are referred to by the batch codes C1FTRDBF and C1FTRDOI.

> **Warning**
>
> If you change the value of **Oldest Bucket Age (Days)** after debt has been marked as "redundant" by C1FTRDBF or C1FTRDOI, the system will NOT re-age the old debt (i.e., once a financial transaction has been marked as "redundant", it is "redundant" forever).

Enter what you consider to be an excellent credit rating in **Beginning Credit Rating**. Collection events can cause an account's credit rating to decrease. When an account's credit rating falls below a certain level, different collection processes may ensue.

Use **Beginning Cash-Only Score** to define the cash-only score for accounts with a perfect payment history (i.e., one without non-sufficient funds). When you cancel a payment tender

Copyright © 2026, Oracle and/or its affiliates.

and use a cancellation reason marked as NSF, the system will cause the account's cash-only score to increase by the value on the payment cancellation reason.

Use **Credit Rating Threshold** to define when an account's credit rating becomes risky. When an account's credit rating falls beneath the Credit Rating Threshold, the system will:

- Assuming you've enabled the Control Central alert algorithm, C1-CRRT-ACCT, an alert displays when an account's credit rating falls below the credit rating threshold on the installation record. This algorithm is plugged-in on the installation record.
- Subject the account's debt to different collection criteria. For more information, refer to Designing Your Collection Class Control Overrides.

Use **Cash-Only Threshold** to define the number of cash-only points a customer must have before the system warns the CSR accepting payments that the account is cash-only.

Installation Options - Financial Transaction

Select **Admin**, and then **General**, and then **Installation Options** and use the **Financial Transaction** page to define financial transaction installation options.

Description of Page

Use **GL Batch** to define the batch process that is used to interface your financial transactions to your general ledger. The process is snapped on FT download records by the GLS background process.

Use **A/P Batch** to define the batch process that is used to interface your check requests (initiated with adjustments with an adjustment type that reference an A/P request type) to you accounts payable system.

Use **Fund Accounting** to indicate if fund accounting is Practiced or Not Practiced at your organization.

Use **Alternate Currency** to indicate if your organization accepts customer payments in currencies other than the account's currency.

> **NOT_SUPPORTED**
>
> Refer to Alternate Currency Payments to understand the significance of this option.

Installation Options Framework - Algorithms

Select **Admin**, and then **General**, and then **Installation Options - Framework** and use the **Algorithms** page to define specific system events.

The following table describes each **System Event**.

System Event	Optional / Required	Description

Copyright © 2026, Oracle and/or its affiliates.

Account Information	Optional	We use the term "Account information" to describe the basic information that appears throughout the system to describe an account. The data that appears in "account information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Account information".
Adjustment Information	Optional	We use the term "Adjustment information" to describe the basic information that appears throughout the system to describe an adjustment. The data that appears in "Adjustment information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Adjustment information". Note: This algorithm may be further overridden by an "Adjustment information" plug-in on the Adjustment Type. Refer to Adjustment Type for how algorithms of this type are used.
Appointment Information	Required	We use the term "Appointment information" to describe the basic information that appears throughout the system to describe an appointment. The data that appears in "appointment information" is constructed using this algorithm.
Auto Pay Creation	Required if you allow customers to pay automatically	This algorithm is executed to create automatic payments whenever the system creates automatic payments. Refer to How And When Are Automatic Payments Created for the details.
Bill Cycle Selection	Optional	This algorithm is only used if your organization has specific rule for assign a bill cycle to an account. This plug-in spot is invoked by the SA Activate/ Stop common routine. The spot is called right after the system has updated the account.

Copyright © 2026, Oracle and/or its affiliates.

Bill Information	Required	We use the term "Bill information" to describe the basic information that appears throughout the system to describe a bill. The data that appears in "bill information" is constructed using this algorithm.
Bill Segment Information	Optional	We use the term "Bill segment information" to describe the basic information that appears throughout the system to describe a bill segment. The data that appears in "bill segment information" is constructed using this algorithm.
Case Information	Optional	We use the term "Case information" to describe the basic information that appears throughout the system to describe a case. The data that appears in "case information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "Case information". Note: This algorithm may be further overridden by a "Case information" plug-in on the Case Type. Refer to Case Type for how algorithms of this type are used.
Collection Agency Referral Information	Optional	We use the term "Collection Agency Referral information" to describe the basic information that appears throughout the system to describe a collection agency referral. Plug an algorithm into this spot to override the system default "collection agency referral information".
Collection Process Information	Optional	This algorithm displays additional information related to a collection process in a special field on the collection process main page.

Copyright © 2026, Oracle and/or its affiliates.

Control Central Alert	Optional	These algorithms produce the alerts shown on the Alert Zone. Additionally, system-generated alerts can be show on the alert zone if enabled. (See the Alert Zone for more information and the list of system-generated alerts.)
Credit Rating 'Created By' Information	Required	The data that appears in the credit rating "created by" information is constructed using this algorithm. Refer to Account - C&C for more information about the credit rating.
Credit Rating History Information	Optional	We use the term Credit Rating History information to describe the basic information that appears throughout the system to describe a credit rating history entry. Plug an algorithm into this spot to override the system default "credit rating history information".
Digital Signature	Optional	This algorithm is only used if your organization generate and store digital signatures for each bill. Refer to Digital Signatures for the details.
Document Number	Optional	If document numbers have been enabled on the installation record, this algorithm type assigns a document number to a bill or payment event.
Document Number Details	Optional	If document numbers have been enabled on the installation record, this algorithm type is responsible for returning the details used to construct the document number.
Determine Open Item Bill Amounts	Required if you use overdue functionality to collect on bills	This algorithm is responsible for determining the unpaid amount of an open-item bill. It can also be used to return the unpaid amount for a specific SA on a bill.
Document Sequence Number	Optional	If document numbers have been enabled on the installation record, this algorithm type is responsible for generating document sequence number only.

Copyright © 2026, Oracle and/or its affiliates.

FA Additional Information	Optional	This algorithm displays additional information related to a field activity in a special field called Additional Info on the field activity main page. For example, contact information linked to the field activity's field order may be displayed.
FA Information	Required	We use the term FA information to describe the basic information that appears throughout the system to describe a field activity. The data that appears in "FA information" is constructed using this algorithm.
Item Information	Required if you have items	We use the term "Item info" to describe the basic information that appears throughout the system to describe an item. The data that appears in "Item info" is constructed using this algorithm.
Meter Information	Required if you have meters	We use the term "Meter info" to describe the basic information that appears throughout the system to describe a meter. The data that appears in "Meter info" is constructed using this algorithm.
Meter Read High Low Limits	Not used with Customer To Meter	
Online Bill Display	Optional	This algorithm constructs or retrieves a PDF that contains the image of a bill. This algorithm is executed when the Display Bill button is clicked on the Bill page. Refer to Technical Implementation of Online Bill Display for more information.
Online Field Order Display	Optional	This algorithm constructs a PDF that contains the image of a field order.
Online Letter Display	Optional	This algorithm constructs or retrieves a PDF that contains the image of a letter. This algorithm is executed when the Display Letter button is pressed on the Customer Contactportal. Refer to Technical Implementation of Online Letter Display for more information.

Copyright © 2026, Oracle and/or its affiliates.

Online Quote Display	Optional	This algorithm constructs a PDF that contains the image of a quote. This algorithm is executed when the Display Quote button is pressed on Quote - Main. Refer to Technical Implementation of Online Quote Display for more information.
Online Statement Display	Optional	This algorithm constructs a PDF that contains the image of a statement. This algorithm is executed when the Display Statement button is pressed on Statement - Main. Refer to Technical Implementation of Online Statement Display for more information.
Override Proration Factors	Optional	This algorithm is only used if your organization has unusual rate proration requirements that necessitate the overriding of the base package proration logic. For example, you may have certain calculation rules whose charges should never be prorated. Refer to Overriding Proration Factors for more information.
Override Seasonal Proration	Optional	This algorithm is only used if your organization has unusual method of determining the seasons for your calculation rules. For example, you may determine the seasonal boundaries for a calculation rule based on the scheduled meter read date associated with the bill cycle.
Payment Amount Calculation	Required	This algorithm is executed to calculate the amount of an automatic payment for a bill for an account with an active auto pay option. Refer to How And When Are Automatic Payments Created for more information on automatic payments. This algorithm is also executed to default the amount of a manually added payment. Refer to How To Add A New Payment Event for more information on adding a payment manually.

Copyright © 2026, Oracle and/or its affiliates.

Payment Event Information	Optional	We use the term "payment event information" to describe the basic information that appears throughout the system to describe a payment. The data that appears in "payment information" is constructed using this algorithm.
Payment Information	Required	We use the term "payment information" to describe the basic information that appears throughout the system to describe a payment. The data that appears in "payment information" is constructed using this algorithm.
Person Information	Required	In most parts of the system, a person's Main name is displayed to describe a person. However, several transactions do not use this method. Rather, these transactions call the algorithm that's plugged into this spot to construct the person's name. Refer to the description of the **Alternate Representation** flag on the **Main** tab for a list of these transactions and for the rationale behind this algorithm.
Person Name Format	Required	The format of names entered on Person - Main and Order - Main is validated using this algorithm.
Premise Information	Required	We use the term "premise info" to describe the basic information that appears throughout the system to describe a premise. The data that appears in "premise info" is constructed using this algorithm.

Copyright © 2026, Oracle and/or its affiliates.

SA Information	Optional	We use the term "SA information" to describe the basic information that appears throughout the system to describe a service agreement. The data that appears in "SA information" is constructed using this algorithm. Plug an algorithm into this spot to override the system default "SA information". Note: This algorithm may be further overridden by an "SA information" plug-in on the SA Type. Refer to SA Type - Algorithms for how algorithms of this type are used.
Severance Process Cancellation	Optional	This algorithm is executed to perform additional processing when the system cancels a severance process. Note: This algorithm is executed before the Severance Process Template - Post Cancel Algorithm is executed. Canceling a severance process on-line manually does not execute this algorithm.
SP Information	Required	We use the term "SP info" to describe the basic information that appears throughout the system to describe a service point. The data that appears in "SP info" is constructed using this algorithm.

Defining Customer Languages

As described under Defining Languages, you define the language in which each user see the system. In addition to defining each user's language, the system allows you to define each customer's preferred language. For example, one customer can receive bills in English whereas another customer could receive their bills in Chinese.

Each customer's language is defined by the language code on their person record . Bills, adjustments and other system-generated records will then be done in the language of the main customer of the account. In addition, the language code is passed on to all customer-facing interfaces, such as letter requests and bill print.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can define Rates in multiple languages - when a bill is generated, the line-item descriptions are generated and stored in the account's main customer's language of choice. Anyone who subsequently views these bills can only see the descriptions in that language.
>
> To support bills and other correspondence, you must also provide translations of standard bill stock and letters. This must be handled by your printing software vendor.

Defining Accounting Calendars

Accounting calendars determine the accounting period to which a financial transaction will be booked. The following points describe how the system determines a financial transaction's account period:

- Every financial transaction references an accounting date and its service agreement
- Every service agreement references a service agreement type
- Every service agreement type references a GL division
- Every GL division references an accounting calendar
- The accounting calendar contains the cross-reference between the accounting date specified on the financial transaction and related accounting period in your general ledger

> **Warning**
>
> This information must be the same as the information in your financial database.

To add or review an accounting calendar, choose **Admin**, and then **General**, and then **Accounting Calendar**, and then **Search.**

Description of Page

Enter a unique **Calendar ID** and **Description** for the calendar.

Enter the **Number Of Periods** for the calendar. Don't count the adjustment period, if you use one, or any special "system" periods.

Specify the **Fiscal Year**, each **Accounting Period** in that year, a **Period Description**, the **Begin Date** and the **End Date**.

When you enter begin and end dates, you can define monthly calendar periods or any fiscal period that matches your accounting calendar (weekly, bimonthly) as long as the begin and end dates of successive periods do not overlap. Every day of the year must be included in a period; do not leave gaps between period dates.

For each fiscal period, enter the **Open From Date** and **Open To Date**. These dates define when that particular business dates are open for posting financial transactions to that fiscal period. For example, you might calculate a bill on Sept 1 for usage recorded on 31 August. To post this financial transaction in the August period, you must keep it open through Sept 1.

Copyright © 2026, Oracle and/or its affiliates.

As time passes, you will need to return to this transaction to manually enter ensuing years. You can enter several years at a time or incorporate the task into end-of-year system maintenance.

Defining General Ledger Divisions

There are two types of Divisions referenced in the system: a CIS Division and a GL Division. This is a rather powerful structure, but it can be confusing.

- General Ledger divisions typically comprise individual entities (e.g., companies) in your general ledger. You must set up a GL division for each such entity. The GL division's sole purpose in the system is to define the accounting period associated with financial transactions linked to service agreements associated with the GL division (service agreements are associated with GL divisions via their SA type). The system cares about accounting periods in order to prevent a user from booking moneys to closed periods. It also uses accounting periods when it produces the flat file that contains the consolidated journal entry that is interfaced to your general ledger (refer to The GL Interface for more information).

> **Note**
>
> When determining how many GL Divisions you need, be sure to consider your general ledger and how your chart-of-accounts is structured. You will typically have one GL division for each "company" in your general ledger.

- A CIS division is typically associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.

> **NOT_SUPPORTED**
>
> Refer to Setting Up CIS Divisions for information about CIS Divisions.

To define a general ledger division, select **Admin**, and then **General**, and then **General Ledger Division.**

Description of Page

Enter a unique **GL Division** for the general ledger division.

Enter a **Description** of this general ledger division.

Define the accounting **Calendar ID** that controls how to convert an FT's accounting date into an accounting period. Refer to Defining Accounting Calendars for more information.

You may define a **Currency Code** for the GL division.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The GL Download Extract (GLDL) batch process uses the currency code from GL division, by default. For multi-currency support, setting the Use FT Currency Code batch parameter to Y would cause the FT's currency code to be used instead. Refer to the GLDL Detail Record Layout for more details.

Defining Banks & Bank Accounts

The topics in this section describe how to maintain your implementation's bank accounts.

- Bank - Main
- Bank - Bank Account

Bank - Main

To review Banks choose **Admin**, and then **Financial**, and then **Bank**, and then **Search.**

Description of Page

Enter a unique **Bank** and **Description** for the bank.

The **Bank Accounts** collection displays the bank accounts currently linked to this bank code. Use the drill down button to view more details or to modify the bank account details. Alternatively, you may navigate to the Bank Account tab and scroll to the desired bank account.

Bank - Bank Account

To review Bank Accounts for a Bank, choose **Admin**, and then **Financial**, and then **Bank**, and then **Search** and then navigate to the **Bank Account** page.

Description of Page

Use the **Bank Accounts** tab to define the attributes of each bank account. For each account, enter the following information:

- Enter a **Bank Account Key** to identify an Account at a Bank. You may have more than one account at a given bank, and you may have accounts at more than one bank. This code will allow the system to easily identify a specific account.
- Enter a **Description** to appear on prompt lists, inquiries, and reports.
- Enter the **Account Number**, **Check Digit** and if needed, the **Branch ID** of the bank where the account is held.
- Enter the **Currency Code** for the currency in which the account is denominated.
- Use **DFI ID** to define the Depository Financial Institution ID that is interfaced to the automatic payment-processing agent as part of the automatic payment interface.
- Enter the **Distribution Code** to be used for cash GL distributions when a payment is frozen or canceled.

Copyright © 2026, Oracle and/or its affiliates.

Defining Issuing Centers

This section provides information about defining issuing centers that are used to assign document numbers to bills and payment events. An issuing center should be configured for each location that issues bills. The installation record Document Number and Document Number Details algorithms contain the logic used by the system to assign a document number to a bill. To set up an issuing center, open **Admin**, and then **General**, and then **Issuing Center**, and then **Add.**

> **NOT_SUPPORTED**
>
> Refer to Document Numbers for information about document number assignment.

> **Note**
>
> **This section is only relevant for some organizations.** The information in this section is only relevant if your organization indicated on the installation record that it uses document numbers as an alternative bill id. If your organization does not use document numbers as an alternative bill id, then no other setup is required.

The topics in this section describe the base-package zones that appear on the Issuing Center portal.

- Actions
- Issuing Center List
- Issuing Center
- Issuing Center Log

Actions

This is a standard actions zone.

If the issuing center is in a state that has valid next states, buttons to transition to each appropriate next state are displayed.

Issuing Center List

The Issuing Center List zone lists every issuing center. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent issuing center.
- Click the Add link in the zone's title bar to add a new issuing center.

Copyright © 2026, Oracle and/or its affiliates.

Issuing Center

The Issuing Center zone contains display-only information about an issuing center, including its current and historic branches. This zone appears when an issuing center has been broadcast from Issuing Center List zone or if this portal is opened via a drill down from another page.

Please see the zone's help text for information about this zone's fields.

Issuing Center Log

This is a standard log zone.

Defining Service Tasks Options

This section describes concepts and procedures related to service tasks.

- About Service Tasks
- About Service Task Types
- Defining Service Task Types
- Base Package Service Task Types

About Service Tasks

Service tasks are records that can be used to perform a variety of tasks. Examples include:

- Task-related records performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service.
- Task-related records performed by Oracle Utilities Customer Care and Billing to manage specific processing, such as net energy metering true-ups, or prepaid billing to create bill segments for smart meter prepaid service agreements.

The main attribute used to define a service task is service task type. The service task type defines properties common to specific types of service tasks.

For more information, see Searching and Viewing Service Tasks.

About Service Task Types

Service task types define properties common to specific types of service tasks. Service task types represent different types of tasks that can be performed by:

- Users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service. An example of a service task includes self service meter reads, in which users enter their own meter reads via the Customer Self Service application.
- Oracle Utilities Customer Care and Billing to manage specific processing. Examples include net energy metering true-ups, or prepaid billing to create bill segments for smart meter prepaid service agreement.

Copyright © 2026, Oracle and/or its affiliates.

Service task types can be defined by the following attributes:

- **Service Task Type**: the name of the task type.
- **Service Task Type Business Object:** the business object that defines the behavior of the service task type.
- **Service Task Business Object:** the business object instantiated when service tasks of this type are created.
- **Service Task Class:** the category used to define service task types for reporting purposes (self-service, miscellaneous, etc).
- Other data based on the specific type of service task (such as minimum days in true-up period, billing frequency, or default payment method.)

For more information about defining service task types, see Defining Service Task Types.

For more information about the service task types delivered with the application, see Base Package Service Task Types.

Defining Service Task Types

To maintain existing service task event types, select **Admin**, and then **Customer**, and then **Service Task Type,** then use standard actions to edit, duplicate, or delete a service task type.

To define a new service task type, follow these steps:

1. Select **Admin**, and then **Customer**, and then **Service Task Type.**.
2. Enter a name and a meaningful description for the service task type.
3. If needed, select the business object to use when creating service tasks of this type.
4. Select the service task class applicable to service tasks of this type, if applicable.

   Some service tasks types used solely by Oracle Utilities Customer Care and Billing have a default class of **Miscellaneous**.
5. Enter a detailed description for the service task type.
6. Complete the remaining fields and sections, as needed.
7. If applicable, select a To Do type and corresponding To Do role to use when creating To Do entries related to service tasks of this type.
8. Click **Save**.

This service task type can now be used when service tasks are received from other Oracle Utilities applications, such as Oracle Utilities Customer Self Service, or service tasks created specifically by Oracle Utilities Customer Care and Billing.
For more information about service task types, see About Service Task Types.

For more information about the service task types delivered with the application, see Base Package Service Task Types.

Base Package Service Task Types

This section provides descriptions of the service task types provided with the base package.

Service Task Type	Business Object	Detailed Description	Related Transaction Business Object

Copyright © 2026, Oracle and/or its affiliates.

Appointment Notification Task Type	C1-NotifyAppointmentTaskType	This business object is used to capture the information to use in appointment notification processing.	Appointment Notification Task
Create Customer Contact Task Type	C1-CreateCustContactTaskType	This business object is used to capture the information to use in create customer contact task processing.	Customer Contact Request Task
Direct Debit Task Type	C1-DirectDebitMandateTaskType	This business object defines configuration information that is used for processing Single Euro Payments Area (SEPA) direct debit transactions.	Direct Debit Mandate Task
FA Completion Task Type	C1-FACompletionTaskType	This business object captures the attributes used in field activity completion task processing.	FA Completion Task
Field Activity Remark Task Type	C1-FieldActivityRemarkTaskType	This business object is used to capture the information to use in create field activity remark task processing.	Field Activity Remark Task
Form Task Type	C1-FormTaskType	This business object captures attributes relevant for a particular form type to be used when a self-service customer creates a form.	Form Task
Missed Appointment Notification Task Type	C1-NotifyMissedApptTaskType	This business object is used to capture the information to use in missed appointment processing.	Missed Appointment Notification Task
Notification Task Type	C1-NotifyTaskType	This business object captures information about a notification that is made available for sign-up by self-service users.	Notification Preferences

Copyright © 2026, Oracle and/or its affiliates.

Net Energy Metering Task Type	C1-NEMTrueUpTaskType	This business object is used to maintain the various configuration options that are used by the true up monitor (TUM) business object's algorithms. It defines the length of the true up period as well as the adjustments types used during the true up process. For true up reversals, the adjustment cancel reason to use is also captured here.	True Up Monitor Task
Outage Call Type	C1-OutageCallType	This business object defines the behavior of an outage call type used to support trouble calls captured in CCB and integrated to NMS.	Outage Call
Payment Arrangement Task Type	C1-PATaskType	This business object defines the expected behavior for when a self-service user requests a payment arrangement.	Payment Arrangement Task
PPB Payment Notification Task Type	C1-PPBPaymentNotifyTaskType	This business object captures information about the Prepaid Billing Payment Request notification that is made available for sign-up by self-service users.	Notification Preferences
Prepay Biller Task Type	C1-PrepayBillerTaskType	This business object defines the behavior of a prepay biller task type used in smart meter prepay billing.	Prepay Biller Task
Product Offer Publish	C1-ProductOffer	This business object contains the product offer elements that are relevant when publishing product offers to Siebel Energy. It is used to read product offer information when building the initial and final snapshots for the sync request.	Not defined

Copyright © 2026, Oracle and/or its affiliates.

Product Offer Task Type	C1-ProductOfferTaskType	This business object is used to define the behavior of a product offer. Product offers are configured in CCB and published to Siebel Energy.	Not defined
SA Creation Rule Publish	C1-SACreationRule	This business object contains the SA creation rule elements that are relevant when publishing product offers to Siebel Energy. It is used to read SA creation rule information when building the initial and final snapshots for the sync request.	Not defined
SA Creation Rule Task Type	C1-SACreationRuleTaskType	This business object defines the behavior of the SA creation rules for a product offer. Each SA creation rule indicates the type of service agreement to create for the product offer.	Not defined
Service Request Task Type	C1-ServiceRequestTaskType	This business object is used as a parent service task type business object for service request related task types such as appointment notification, missed appointment and customer contact creation requests.	Not defined
Siebel Customer Maintenance Task Type	C1-CustomerMaintenanceTaskType	This business object is used in the Siebel integration. When a customer maintenance request is received from Siebel, an inbound service determines the service task type to use for the event.	Siebel Customer Maintenance Task
Start/Stop Task Type	C1-StartStopTaskType	This business object defines the expected behavior for when a self-service user requests to start, stop or transfer service.	Start/Stop Request Task

Copyright © 2026, Oracle and/or its affiliates.

Defining Customer Questions

A customer question is used to capture specific information related to a request. For example:

- A customer can be asked questions about a service call being reported.
- A customer can be asked one or more questions during a start, stop or transfer service request.

The responses to customer questions are captured on the Response object that is linked to either a customer service request or a service task.

You can access the portal from **Admin**, and then **General**, and then **Customer Question**. You are brought to a query portal with options for searching for a specific question. Once a question is selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones appear as part of the portal's **Main** tab page:

- **Customer Question**: This zone provides information on question and answers. Refer to the zone's embedded help for more details.
- **Service Code List**: This zone displays a list of associated service codes. It is hidden if no service code is associated to the current question.

Agent Contact Scripting

There are processes in the system where agents or CSRs need specific instructions or need to provide specific information to customers. This functionality is referred to as "Agent Contact Scripting" and the system is delivered with specific places in these processes where custom text can be added. These places are called "Script Points" and are maintained as extendable lookup values. For example, the "Date of Birth Explanation" script point for start service process, displayed in the Customer Identification panel of the start service process flow, can be used to explain how the customer's date of birth is used when starting service.

To display custom text for a script point, add a Field with the **Help Text** set to the custom text and associate the field to the script point.

If your implementation has added custom elements through data area extensions, those extensions can reference custom script points. To implement this, you need to ensure your map fragment's class attribute is set to "contact-script-point" and the scriptPoint attribute is set to the CM script point extendable lookup value. Here is the sample syntax: .

Defining System-Wide Options

- Installation Options - Framework
- Feature Configurations
- Time Zones

Installation Options - Framework

Copyright © 2026, Oracle and/or its affiliates.

- Configuring Installation Options - Framework

Configuring Installation Options - Framework

Installation options define the individual applications installed on your system and identify algorithms used to implement core system functions. These options also define global parameters such as the administrative menu style (alphabetical or functional), the country, language, currency code, as well as the base time zone to use for this implementation.

Installation options are stored in the installation record for your system. Use the Installation Options - Framework portal to configure these options. This portal is part of the OUAF and is described in detail in the Framework documentation.

Base Time Zone

The time zone setting of the Installation Options - Framework determines the time zone for all date/times stored within the system. Each date/time, based on the configuration of that field, is stored in either standard or legal time within this base time zone.

Refer to the Glossary of Terms in the *Oracle Utilities Meter Solution Business User Guide* for definitions Standard Time and Legal Time

Note: The installation record does not dictate the server time zone, but rather must match it.

Installation Algorithms

Installation algorithms implement global system functions and can be customized for each implementation. The base package supports the following installation options for Meter Data Management-related system events:

- Geocoding Service: Responsible for geocoding an address (converting an address to a geocode latitude/longitude pair).
- Global Context: Sets global contexts (displayed in the Global Context Sidebar zone) based on the value of existing global contexts. For example, if the Service Point is specified, this algorithm sets the Device by finding the most recently installed Device on the service point. It then sets the Measuring Component by finding the most effective Device Configuration and retrieving any measuring component linked to it. It then sets the Usage Subscription by finding the most recent active usage subscription linked to the service point. The contact is set by finding the main contact for the usage subscription.
  Additional detail on how global context is populated can be found in the detailed description of the D1-GBCTX-DF Algorithm Type.
- To Do Pre-creation:
  - In Meter Data Management, the D1-TDPRCRE (To Do Precreation) algorithm associates a To Do entry via characteristics to the related the related Device, Measuring Component, Service Point, Contact, Usage Subscription, Activity Type and Activity based on the drill keys of the To Do entry.
  - In Customer to Meter, the following algorithms are used to associate a To Do Entry via characteristics with related objects:
    * C1-TDCR-CHAR (Link Account/Premise/Person/SA/SP to To Do)
    * C1-TDCR-DFRL (Default To Do Role via Account/Premise)
    * D1-TDPRCRE (To Do Precreation)
    * X1-TDPRCRE (Customer to Meter To Do Pre-Creation)

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on how the To Dos are associated to related data can be found in the detailed descriptions of these algorithms type.
>
> See Installation Options in the Oracle Utilities Application Framework *Administrative User Guide* for related information on the installation portal.

Feature Configurations

- Configuring Feature Configurations

Configuring Feature Configurations

Some of the features in Oracle Utilities Application Framework based applications are configured by populating options on a "feature configuration". For example, if your implementation uses Oracle Utilities Customer Care and Billing's batch scheduler, you must populate a variety of options on the batch scheduler's feature configuration.

> **Note**
>
> Refer to Defining Feature Configurations in the Oracle Utilities Application Framework *Administrative User Guide* for additional information.

Oracle Utilities Meter Data Management uses the following types of feature configurations (please note that more information can be found on each of these options by viewing their detailed description in the Feature Configuration portal):

Measurement Data Options

Measurement Data Options are used to define behavior related to periodic estimation of initial measurement data, including:

- **No of Hours in Past to Retrieve Last Usable Measurement**: This option is leveraged by scalar periodic estimation to restrict how far into the past it will search for existing measurements when the last contiguous measurement is being initialized on the measuring component. This is to ensure that the first time scalar period estimation is executed on a large number of measuring components that have not been initialized for periodic estimation the system does not execute a large number of unbounded queries into the past which would result in poor performance.

Business Intelligence Configuration

Business intelligence configuration is used to define external data source indicators used when Oracle Utilities Meter Data Management is integrated with Oracle Utilities Business Intelligence. External data source indicators allow business intelligence extracts to properly link the external identifiers to the source external system. The Value of the Data Source Indicator option should match the Environment ID on the Installation Option of the external system.

Copyright © 2026, Oracle and/or its affiliates.

General System Configuration

This feature configuration is owned by Oracle Utilities Application Framework but there are several important option types that have been created specific to Oracle Utilities Meter Data Management:.

- **Multi Time Zone Support**: By default the system assumes an installation operates within a single time zone. In order to enable multiple time zone functionality this option must be defined and set to "D1YS".
- **CCB Link URL**: this option is used to provide the destination URL for hyperlinking into Oracle Utilities Customer Care and Billing.
- **Trace On Flag**: can be set to automatically trace any initial measurement being processed. Note: if the initial measurement has had trace explicitly turned off then no tracing will occur. This is useful for tracing initial measurements created through automated processes where you are unable to set the trace flag directly.
- **System Override Date**: this option type is provided by OUAF but it is highly useful in testing prior to production. When set it will override the system date/time for all users. The format must be entered as: YYYY-MM-DD. For example January 1st 2010 would be 2010-01-01.

> **Note**
>
> Refer to Configuring Multi-Time Zone Support for additional information on setting up multi-time zone support.

Time Zones

- Configuring Time Zones
- Configuring Multi-time Zone Support

Configuring Time Zones

To support businesses spanning across multiple time zones, the system stores all date and time information in a single common time zone known as the base time zone or the server time zone. Furthermore, date and time information is stored in either standard time (i.e. independent of any Daylight Savings Time adjustments, if applicable, in that time zone) or legal time (i.e. shifted according to Daylight Savings Time).

The system also allows data to be entered and displayed in a different time zone in legal time (i.e. adjusted for Daylight Savings Time, managing the conversion back and forth between the data entry time zone and the storage time zone).

Entities associated with a physical location such as measurements (initial and final), measuring components, device configurations, devices, installation events, service points, and usage subscriptions are entered and displayed in the specific time zone where they occur, the entity time zone. The rest of the application uses the base time zone to display date and time information.

When configuring time zones the following fields are of high importance:

Copyright © 2026, Oracle and/or its affiliates.

- **Time Zone Name**: identifies the Olson time zone and as such defines the appropriate offset from Greenwhich Mean Time as well as the schedule for shifting into and out of Daylight Savings Time.
- **Default Time Zone Label**: will be appended to date times that do not fall within Daylight Savings Time
- **Shifted Time Zone Label**: will be appended to date times that do fall within Daylight Savings Time

> ⚠ **Warning**
>
> The server time zone must be correctly specified on the installation options record for the system to work properly.

For additional information see Defining Time Zones in the *Oracle Utilities Application Framework Administrative User Guide*.

Configuring Multi-time Zone Support

As a default the system assumes operations are in a single time zone. This has a few high level impacts:

- All master data must have a time zone that is equal to the base time zone. As such all master data will have the same time zone and that time zone will be the base time zone
- Certain logical complexities are avoided during high volume processing given the knowledge that all master data time zones are equal to the base time zone.

If your implementation resides within multiple time zones then the Multi Time Zone Support feature configuration must be enabled. Doing so will allow master data to be defined with any time zone configured in the system. It will also enable logic to convert time zones between the master data time zones and the base time zones.

You can access the feature configuration portal from the **Admin > General > Feature Configuration**.

From the list of results returned select the feature name for the feature type General System Configuration. For the option type Multi Time Zone Support supply "D1YS" as the value to turn multiple time zones on.

Refer to Multiple Time Zone Support for information about how functional processing is impacted by multiple time zone support.

Defining Financial Transaction Options

Bills, payments and adjustments share one very important trait - they affect how much your customers owe. This section explains the financial design of the system and describes how to set up the tables that control the financial impact of these transactions.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The tables in this section are the first of many that must be set up before you can create bills and apply payments. In this section, we limit the discussion to those tables that control the financial impact of bills, payments and adjustments. In later sections, we describe the tables that control other billing-related functions like meter reading and rates. It is only after all of these tables are set up that you will be able to generate the various financial transactions.

- The Financial Big Picture
- Service Agreement Type Controls Everything
- Designing and Defining Budget Plans
- Tender Management
- Automatic Payment Options
- One-Time Payment Options
- Third Party Payment Processing Options
- Payment Advices
- Credit Card Payments
- Non CIS Payments
- Alternate Currency Payments
- Payment Event Distribution
- Cancel Reasons
- Miscellaneous Financial Controls
- Payables Cash Accounting
- Deferred Accrual Accounting
- Open Item Accounting
- Fund Accounting
- Bill Taxation Threshold
- Other Financial Transaction Topics

The Financial Big Picture

This section provides an overview of the relationship between an account and the various financial transactions that influence how much a customer owes.

Copyright © 2026, Oracle and/or its affiliates.

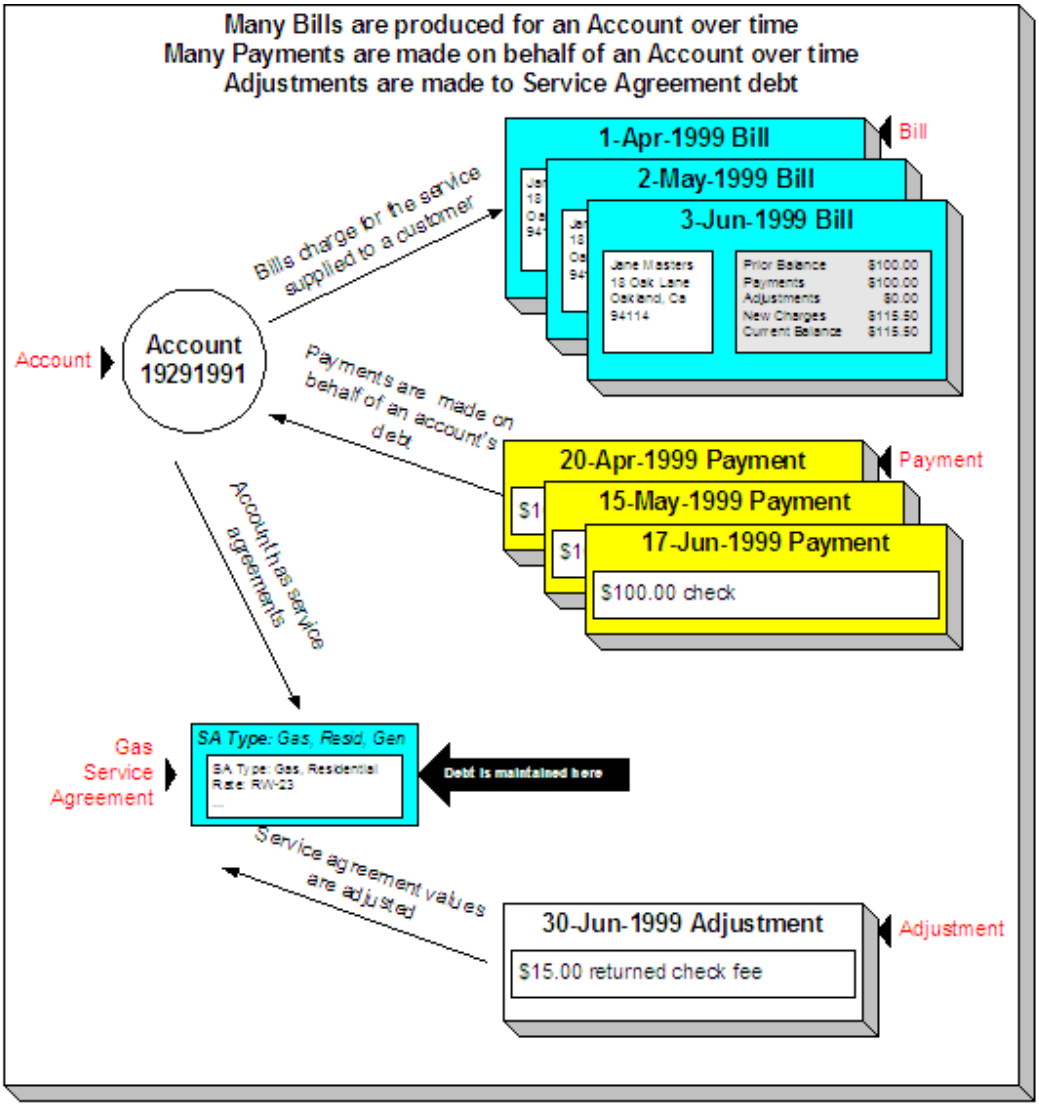
> **Warning**
>
> If your organization practices cash accounting for payables (i.e., you only pay the taxing authority when you get paid), refer to Payables Cash Accounting. If you organization practices open-item accounting (i.e., payments must be matched to bills), refer to Open Item Accounting.

- Bills, Payments & Adjustments
- Bill Details
- Payment Details
- Adjustment Details
- Current Amount versus Payoff Amount
- Financial Transactions Created Between Bills
- Financial Transactions And Aged Debt
- Preventing SA Balances And The GL From Being Impacted Until Bill Completion
- Forcing The Freeze Date To Be Used As The Accounting Date
- How Late Payment Charges Get Calculated

Bills, Payments & Adjustments

The following diagram illustrates the relationship between an account and its financial transactions:

Copyright © 2026, Oracle and/or its affiliates.



The following concepts are illustrated above:

Bills are produced for accounts Over time, many bills may be produced for an account. For more information about a bill, see Bill Details .

Payments are made for accounts Over time, many payments may be applied to an account's debt. For more information about a payment, see Payment Details .

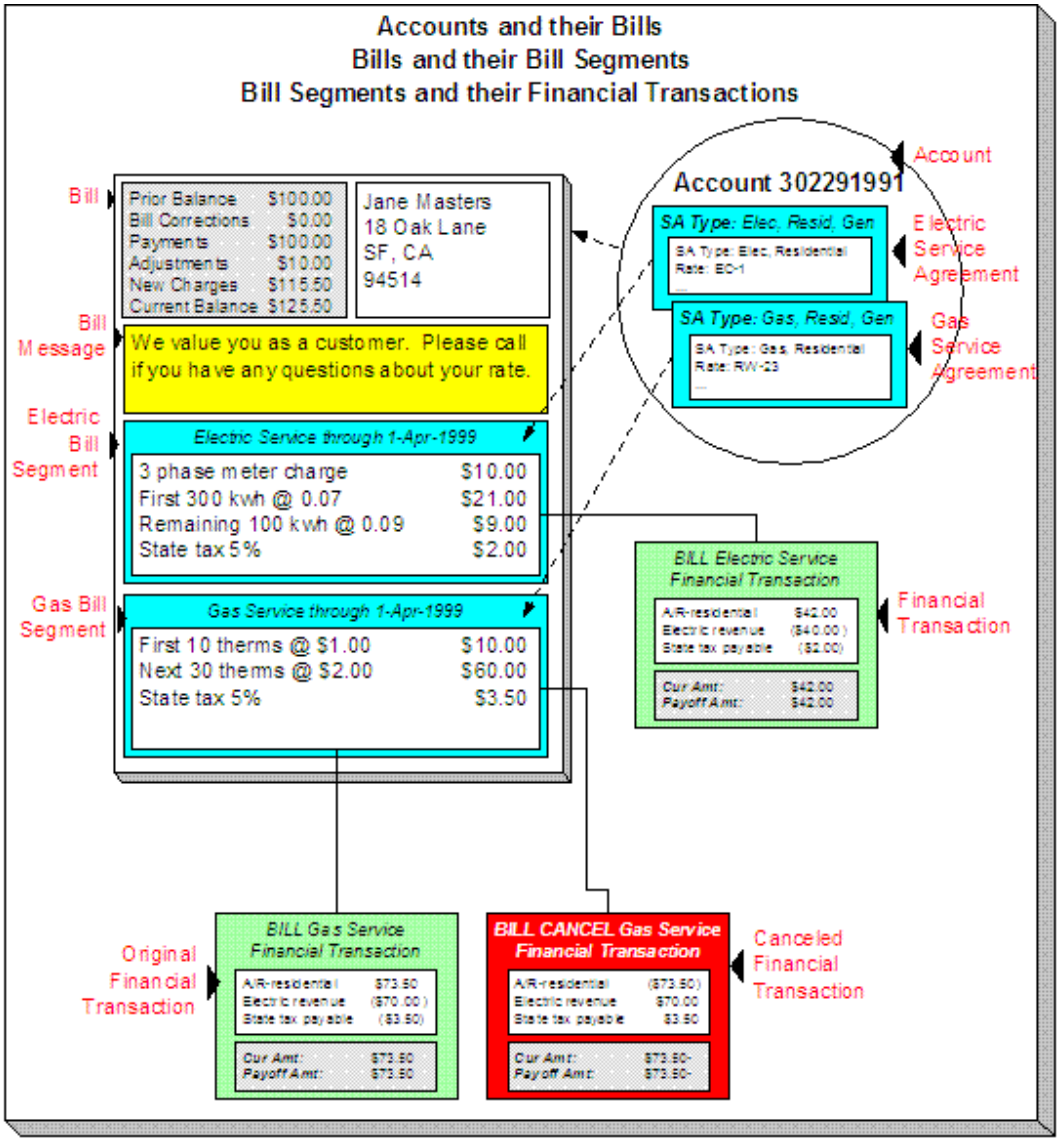
Service agreements have debt The system maintains debt on each individual service agreement. An account's debt is the sum of its service agreements' debt.

Service agreements are adjusted Over time, the debt that is stored on an account's service agreement(s) may be adjusted. For more information about an adjustment, see Adjustment Details .

Bill Details

The following diagram illustrates the relationship between an account and its bills:

Copyright © 2026, Oracle and/or its affiliates.



The following concepts are illustrated above:

A bill is produced for an account Over time, many bills are produced for an account. Bills charge for the services supplied to a customer. The above illustration shows a single bill.

Bills contain bill segments A bill typically contains one bill segment for every active service agreement linked to its account. The only time this is not true is when service agreements for different frequencies exist. For example, an account with a monthly and a quarterly service agreement will only have 4 bills a year that contain both bill segments; the other months' bills will contain a single bill segment for the monthly service agreement.

Bill segments contain calculation details A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

A bill segment has a financial transaction A bill segment has a related financial transaction. A financial transaction contains the financial effects of the bill segment on the service agreement's current and payoff balances and on the general ledger.

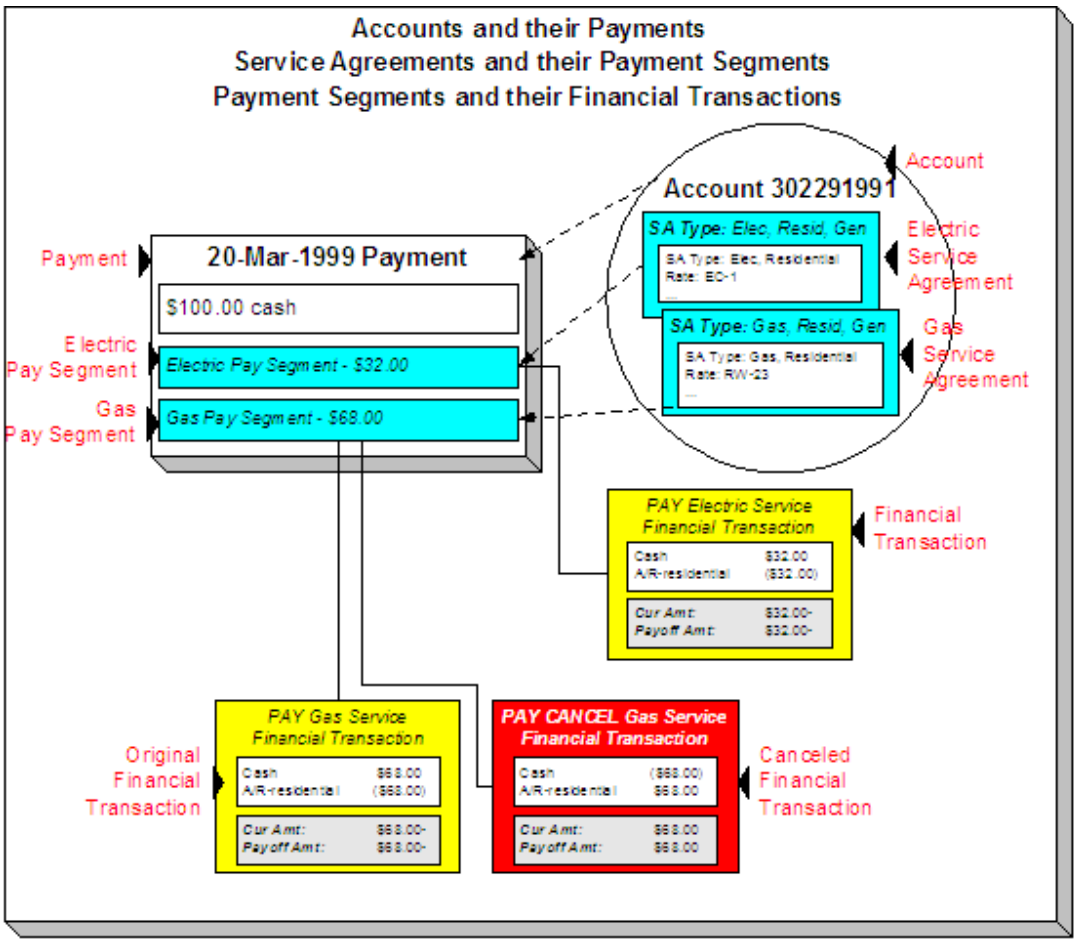
Canceling a bill cancels the financial tran. If the bill segment is eventually cancelled, another financial transaction will be linked to the bill segment to reverse its original financial

Copyright © 2026, Oracle and/or its affiliates.

transaction. The cancellation financial transaction appears on the next bill produced for the account as a bill correction.

Payment Details

The following diagram illustrates the relationship between an account and its payments:



The following concepts are illustrated above:

Payments are made for accounts Over time, many payments may be applied to an account's debt. The above illustration shows a single payment.

Payments contain payment segments A payment contains one payment segment for every service agreement to which the payment is distributed. For a customer who pays in full, the number of payment segments will coincide with the number of bill segments on the bill being paid.

A pay. segment has a financial transaction A payment segment has a related financial transaction. A financial transaction contains the financial effects of the segment on the service agreement's current and payoff balances and on the general ledger.

Canceling a payment cancels the fin. tran. If the payment is eventually cancelled, another financial transaction will be linked to the related payment segment(s) to reverse their financial effect. The cancellation financial transaction appears on the next bill produced for the account as a negative payment.

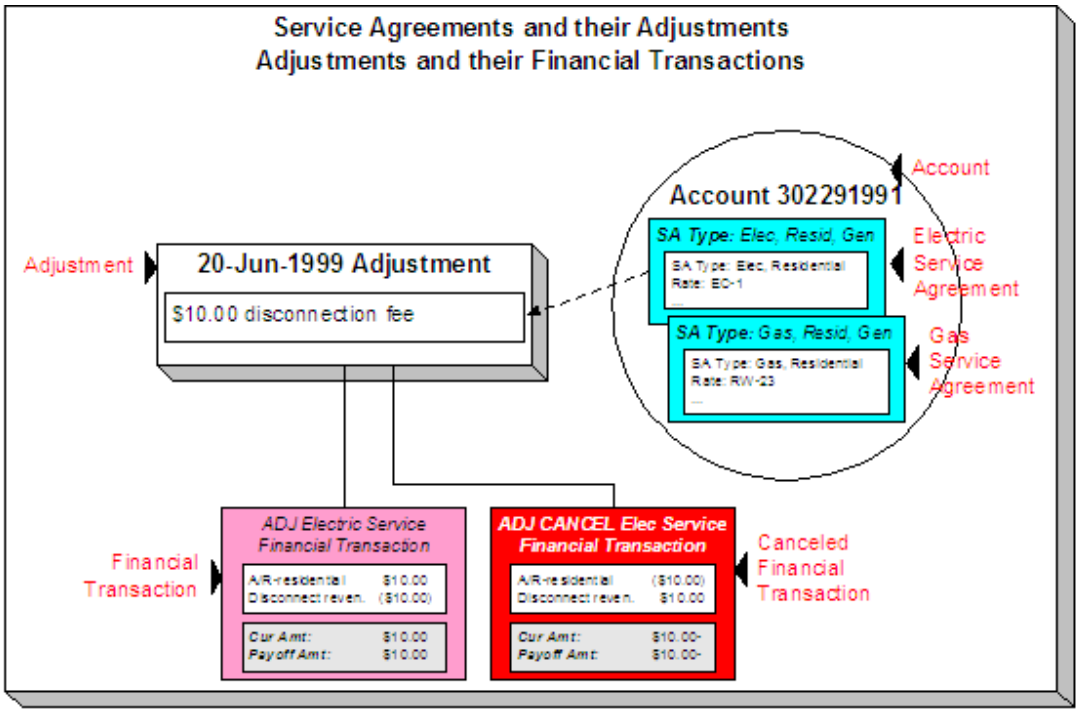
Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

A payment cannot be applied to an account's debt without an associated payment event. Refer to The Big Picture of Payments for more information.

Adjustment Details

The following diagram illustrates the relationship between an account and its adjustments:



The following concepts are illustrated above:

Service agreements have adjustments Over time, a service agreement may have many adjustments. The above illustration shows a single adjustment to one of the account's service agreements.

An adjustment has a financial transaction An adjustment has a related financial transaction. The financial transaction contains the financial effects of the adjustment on the service agreement's debt and on the general ledger.

Canceling an adjust. cancels the fin. tran. If the adjustment is eventually canceled, another financial transaction will be linked to the adjustment to reverse its financial effect. The cancellation financial transaction appears on the next bill produced for the account as an adjustment.

Current Amount versus Payoff Amount

A financial transaction contains two very important attributes: payoff amount and current amount. These attributes contain the grand total of how much the customer owes.

Copyright © 2026, Oracle and/or its affiliates.

- Current amount contains how much the customer THINKS THEY OWE.
- Payoff amount contains how much the customer REALLY OWES.

You may be wondering when these two values can be different? Well, for most financial transactions, these values are the same. These values differ under the following situations:

- When a bill segment charges a customer for a charitable contribution, payoff amount will be zero because the customer doesn't really owe anything (they don't have to contribute if they don't want to). Current amount will be equal to the agreed charitable contribution amount (the customer thinks they owe the contribution).
- When a bill segment charges a customer for a deposit, payoff amount will be zero because the customer doesn't really owe anything (billed deposits are typically not viewed as being a receivable). Current amount will be equal to the amount billed (the customer thinks they owe the deposit amount).
- When a bill segment charges a customer who participates in a levelized payment program (e.g., budget billing or non-billed budgets) the two "amounts due" will contain different values. Payoff amount is equal to how much the customer really owes for the service they consumed; current amount is equal to how much they think they owe in accordance with their monthly budget.

A perhaps easier way to view these two attributes is to consider payoff amount as the "cash out amount", i.e., the amount the customer would owe the utility if they wanted to clear up all debt. The current amount contains the amount the customer thinks they owe. If you're still struggling with the difference, think about your monthly Visa bill: it contains a monthly minimum payment and the total amount owed. The minimum payment is the current amount; the total amount owed is the payoff amount.

The topics in this section provide more information about these two fields.

- What Controls What Gets Booked To Current And Payoff Amount?
- Arrears
- GL Accounting Information
- A Complicated Example

What Controls What Gets Booked To Current And Payoff Amount?

As described in Bill Details , every bill segment has a sibling financial transaction. The financial transaction defines the bill segment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the bill segment's bill segment type.

NOT_SUPPORTED

For more information, refer to Billing - Current Balance versus Payoff Balance and Designing and Defining Bill Segment Types.

As described in Payment Details, every payment segment has a sibling financial transaction. The financial transaction defines the payment segment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the payment segment's payment segment type.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to Payment - Current Balance versus Payoff Balance and Setting Up Payment Segment Types.

As described in Adjustment Details, every adjustment has a sibling financial transaction. The financial transaction defines the adjustment's affect on current and payoff amounts due. The system populates these two fields as per the Financial Transaction Algorithm defined on the adjustment's adjustment type.

NOT_SUPPORTED

For more information, refer to Adjustments - Current Balance versus Payoff Balance and Setting Up Adjustment Types.

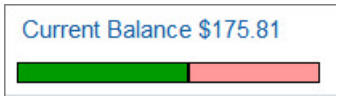
Arrears

The system keeps track of the age of each customer's debt to the day. For example, if a customer hasn't paid their last two bills, the customer's aged debt might look as follows:

- $124.50: 22 days old
- $213.41: 51 days old

Please be aware that it is the current balance (i.e., what the customer thinks they owe) that is aged. Also keep in mind that the moment an FT is frozen, it impacts a customer's current balance.

The system represents aged debt in a variety of ways of the various transactions in the system. On the Current Customer Zone and the Financial Information Zone arrears are shown in a colorful bar (where each color corresponds to different aged buckets):



Whereas on Service Agreement - Main, aged debt is shown in a grid:

Debt Class Arrears

Days Old	Arrears Amount
4	$99.45
32	$76.36

The grid method is used on many pages throughout the system. The following rows *may* appear in the grid:

- A row labeled New Charge highlights all debt that hasn't started aging yet. For example, if you've created a late payment charge and it hasn't appeared on one of the customer's bills, it will be classified as a New charge until the next bill is completed for the customer

Copyright © 2026, Oracle and/or its affiliates.

(unless a user overrides the late payment charge's arrears date by drilling into the financial transaction).

- A row with a label containing n Future (where n is the number of days) appears if there is "future debt". Future debt is very rare and can only exist if a debit financial transaction has a future arrears date. Financial transactions can receive a future arrears date if a bill is completed with a future date or if a user overrides a financial transaction's arrears date with a future date).
- A row that contains a number (and nothing else) represents debt that has started aging. The number is the age of the respective debt. In the above example, the customer has 1 day old debt, and debt that is more than 150 days old. Notice that the 150 day old debt is prefixed with a +. This means that the related debt is more than 150 days old. This age limit is controlled by a field on Installation Options - CC called "Oldest Bucket Age". This field limits the number of days the system will age debt. For example, if you set this field to 150, the system will never age an FT more than 150 days (and all debt that's older than 150 days will be classified as 150 day old debt). Also note, the aged debt bar that appears on Current Customer Zone only ages debt a maximum of 60 days.
- A row with a label of Disputed appears if the account is an open-item customer and this customer has disputed financial transactions.

NOT_SUPPORTED

Refer to Financial Transactions And Aged Debt for more information.

GL Accounting Information

Be aware that if payoff amount is non-zero, the financial transaction has general ledger detail lines.

There are unusual financial transactions whose payoff amount is zero, but still affect the general ledger:

- Bill segments for company usage do not impact payoff amount (because your organization doesn't really owe itself anything). However, the GL is affected.
- Payment segments for charitable contributions (created when your customers contribute extra money to a charity) do not impact payoff amount. Why? Because payoff amount is never debited when a charitable contribution is billed (the customer doesn't truly owe you for this receivable). It's only when the customer pays the contribution that the GL is impacted (debit cash, credit charitable contribution payable).
- If the SA has a special role of Loan, the financial transaction algorithms supplied with the base package transfer the current amount between the long-term receivables and the short-term receivables in the GL. This allows the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Refer to Payoff Balance and Current Balance for Loans for more information.

The effect on your GL is controlled by the financial transaction algorithm defined on your bill segment and payment segment types.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The GL Interface for how GL account information is interfaced to the general ledger.

A Complicated Example

The financial ramifications of a revolving charge account are predictable (if you're an accountant). The following table outlines the different financial events and their impact on the general ledger, arrearage history, and the amounts due (both current and payoff).

Event	GL Accounting	Arrearage Rule	Effect On Payoff Amt	Effect On Current Amt	Payoff Balance	Current Balance
Merchandise purchased	A/R 1000 Revenue <1000>	n/a (current amount is zero)	+1000	0	1000	0
Monthly bill	A/R 10 Int. Rev <10>	$120 aged accordingly	+10	+120	1010	120
Payment received	Cash 120 A/R <120>	$120 relieved accordingly	-120	-120	890	0

The following points describe the events in the above table:

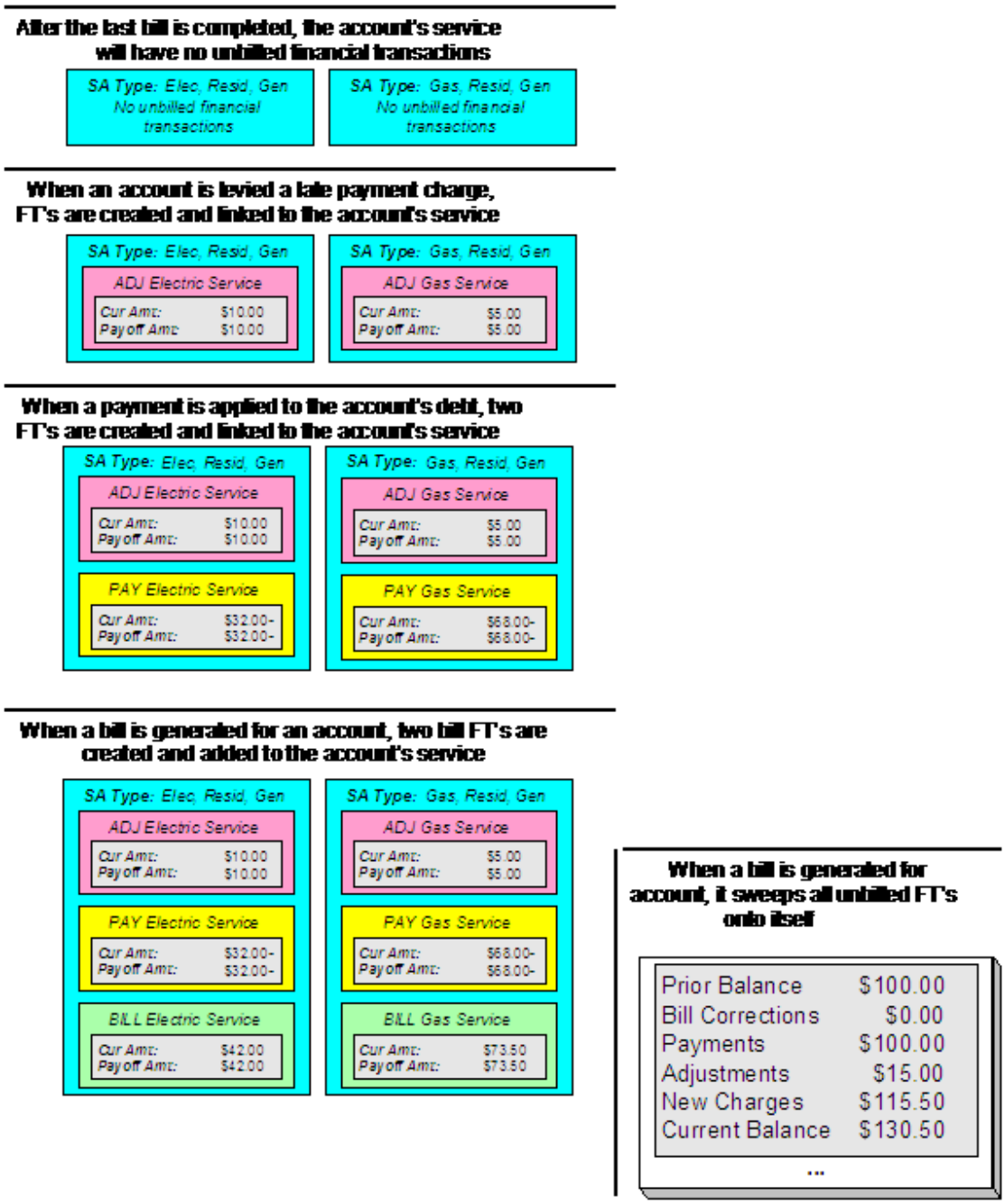
- **Merchandise purchased.** When a customer purchases an air conditioner:
  - The system generates an adjustment to book the purchase.
  - The customer doesn't really think they owe the entire $1,000 (because they've purchased it on credit), therefore no moneys are booked to current amount. However, if the customer wanted to cash out, they would owe your organization $1,000, therefore the entire amount of the purchase is booked to payoff amount.
  - Because no money was booked to current amount, this event has no impact on the arrearage history.
- **Customer billed.** Monthly, the system calculates how much the client owes. In this example, interest is calculated to be $10 and the minimum monthly payment is set at $120.
  - The interest is posted to the GL, but principal isn't since it was booked when the merchandise was purchased.
  - The customer really thinks they owe the minimum payment amount, $120. Therefore, current amount is affected. However, if the customer were to cash out, they would owe your organization $1,000 + $10 (the interest); therefore payoff amount is affected by only $10.
  - Because current amount changed by $120, arrearage history is affected accordingly.
- **Payment received.** With any luck, the client will pay the $120 that was billed (note, they could obviously pay more).
  - The payment has a normal affect on the GL (debit cash, credit A/R).

Copyright © 2026, Oracle and/or its affiliates.

- The amount the customer thinks they owe decreases by $120, therefore current amount is affected by the payment amount. And, if the customer was to cash out, they would owe the utility $120 less, therefore payoff amount is affected by the payment amount.
- Because current amount changed by $120, arrearage history is affected accordingly.

Financial Transactions Created Between Bills

The following diagram illustrates how frozen financial transactions (FT's) accumulate between bills and are swept onto the next bill produced for the account (when the bill is completed). This example assumes



Copyright © 2026, Oracle and/or its affiliates.

When any type of financial transaction is frozen, it impacts the related service agreement's current and payoff balances. If you do not want adjustments and bill segments to affect the customer's balance until they appear on the customer's next bill, refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion .

Notice the balances in the financial summary of the above bill:

- The **Prior Balance** is the ending balance from the customer's prior bill.
- The **Bill Corrections** portion is blank. It contains a value if you cancel / rebill a bill segment that appeared on an earlier bill.
- The **Payments** portion shows payment financial transactions (both new payments and cancellations) that have been created since the last bill.
- The **Adjustments** portion shows adjustment financial transactions (both new adjustments and cancellations) that have been created since the last bill.
- The **New Charges** portion shows bill financial transactions that were created when the bill was created.
- The **Current Balance** is the total amount owed.

NOT_SUPPORTED

If you practice Open Item Accounting, refer to Open Item Versus Balance Forward Accounting for more information about financial transactions and bills.

Financial Transactions And Aged Debt

The system keeps track of how old a service agreement's current balance is in order to determine if the customer is in arrears (and therefore credit and collections processing should start).

A financial transaction (FT) impacts the related service agreement's current and payoff balances the moment it is frozen. However, some types of frozen FTs have no impact on a customer's aged debt until the next bill is completed for the account associated with the service agreement.

As described in the previous section, a frozen financial transaction (FT) waits in limbo until the customer's next bill is produced. This limbo period could be several weeks if the customer is billed infrequently. When the customer's next bill is completed, all such frozen FT's are linked to the bill. It is important to stress the following in respect of these FT's:

- If the FT decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the FT appears on a bill.
- If the FT increases the amount of debt, the amount the customer owes from an aged debt perspective may or may not be affected by the FT. There is a switch on an FT called New Charge that controls the arrears behavior.
  - If this switch is on, the amount of debt will be reflected as a "new charge" when you look at the customer's aged debt. This amount will remain classified as a "new charge" until the FT is swept onto a bill. The moment the FT is swept onto the customer's bill, the debt starts aging. This logic exists because you probably don't want to start aging an FT until the customer has actually seen it.

Copyright © 2026, Oracle and/or its affiliates.

- If this switch is off, the date on which the FT starts aging must be defined in the Arrears Date field. The Arrears Date is used to compute how many days old the debt is.

> **Note**
>
> **Aged debt limitations.** It's important to be aware that there's a field on Installation Options - CC called "Oldest Bucket Age" that limits the number of days old the system will age debt. For example, if you set this field to 360, the system will never age an FT more than 360 days (and all debt that's older than 360 days will be classified as 360 day old debt). Also note, the aged debt bar that appears on Control Central - Account Information only ages debt a maximum of 60 days.

> **NOT_SUPPORTED**
>
> If you practice Open Item Accounting, refer to Open Item Versus Balance Forward Accounting for information about how open-item FT's affect aged debt.

Preventing SA Balances And The GL From Being Impacted Until Bill Completion

It's important to understand that when any type of financial transaction is frozen, the related service agreement's balance is affected. For example:

- When a payment is frozen, the customer's balance is reduced.
- When an adjustment is frozen, the customer's balance is impacted.
- When a bill segment is frozen, the customer's balance is increased (typically).

For payments, there is no issue. However, for bill segments and certain types of adjustments, you may NOT want the customer's balance to be impacted until the next bill is completed. Consider the following scenarios:

- Late payment charges:
  - You can setup the system to create a late payment charge (i.e., an adjustment) say 5 days after an unpaid bill is due.
  - If the related adjustment is frozen, the customer's balance will be impacted. However, its impact will not affect aged debt until the next bill is completed. In other words, the amount of the frozen adjustment segment will appear as a "New Charge" until the bill is completed.
- Batch billing:
  - If a customer has multiple service agreements, it's possible for one of the service agreements to have a bill segment that's in error and the other service agreement's bill segment to be error-free.
  - If this happens and you have setup the bill cycle schedule to freeze bill segments if they're error-free, then you could have one bill segment frozen and another in error.

Copyright © 2026, Oracle and/or its affiliates.

- The frozen bill segment will impact the customer's balance. However, its impact will not affect aged debt until the bill is completed (and a bill cannot be completed until all of its bill segments are error-free). In other words, the amount of the frozen bill segment will appear as a "New Charge" until the bill is completed.

> **Warning**
>
> We'd like to stress that while a frozen financial transaction impacts a customer's balance the moment it is frozen, the amount of the financial transaction appears as a "New Charge" when viewing a customer's aged-debt. This amount will remain classified as a "New Charge" until the next bill is completed (i.e., the customer's debt doesn't start aging until the next bill is sent to the customer).

While aged-debt isn't impacted by frozen FT's, the general ledger is. This is because a financial transaction is marked for interface to the general ledger when it is frozen. This can be problematic if you have a long period between FT freeze and bill completion (you could impact the general ledger but not impact the customer's balance). If this is unacceptable, you can setup the system to not allow certain types of FT's to be frozen until the next bill is completed. This means that neither the customer's balance nor the general ledger will be impacted until bill completion time. To do this:

- Choose the Freeze At Bill Completion option on Installation Options - Billing.
- Examine each of your adjustment types. Select Freeze At Bill Completion for those that should not impact the customer's balance or the general ledger until the next bill is completed. Select Freeze At Will for those that should impact the customer's balance and the GL when they are frozen. Typically, the only adjustment types for which you'd choose Freeze At Will option are those that cause a customer's balance to be reduced, those that are used to refund money to a customer, and those that are created at bill completion. Adjustment types for adjustments created during bill completion (e.g., by a bill completion algorithm) must have their adjustment freeze option set to Freeze At Will. Otherwise (i.e., if the option is Freeze At Bill Completion) they will not be frozen until a subsequent bill is completed.

Please be aware of the following in respect of the Freeze At Bill Completion options:

- If you turn on Freeze At Bill Completion on Installation Options - Billing:
  - Users will not be allowed to freeze bill segments online. This means that the freeze button will be disabled on Bill - Main , Bill - Bill Segments and Bill Segment - Main.
  - The Billing background process will not freeze bill segments until all segments on a bill are error free (and permission has been granted on the bill cycle schedule to complete bills).
  - Bill segments will exist in the freezable state until the bill is completed.
- If you turn on Freeze At Bill Completion on Adjustment Type - Main:
  - Users will not be allowed to freeze adjustments of this type online. This means that the freeze button will be disabled on Adjustment - Main .
  - Background processes that create adjustments will not freeze this type of adjustment. Rather, the adjustments will be frozen when the next bill is completed.
  - Adjustments of this type will therefore exist in the freezable state until the next bill is completed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Alerts highlight freezable FT's.** Please be aware that messages appear in the Account Information - Financial Information Zone and in the Sidebar - Financial Information Zone to highlight the existence of freezable financial transactions.

Please be aware of the following in respect of the Freeze At Will options:

- If you turn on Freeze At Will on Installation Options - Billing:
  - Users will be allowed to freeze bill segments online. This means that the freeze button will be enabled on Bill - Main , Bill - Bill Segments and Bill Segment - Main.
  - The Billing background process will freeze bill segments when the individual segment is error-free (and permission has been granted on the bill cycle schedule to freeze bill segments).
  - Bill segments will exist in the frozen state regardless of whether the bill is completed.
  - The frozen bill segment's FT will be interfaced to the GL when the interface next runs.
  - All adjustment types must be also be set to Freeze At Will (otherwise they wouldn't get frozen).
- If you turn on Freeze At Will on Adjustment Type - Main:
  - Users will be allowed to freeze adjustments of this type online. This means that the freeze button will be enabled on Adjustment - Main .
  - Background processes that create adjustments will freeze this type of adjustment.
  - Adjustments of this type will exist in the frozen state prior to bill completion.
  - The frozen adjustment's FT will be interfaced to the GL when the interface next runs.

Forcing The Freeze Date To Be Used As The Accounting Date

Every financial transaction references an accounting date. The accounting date controls the accounting period to which the financial transaction is booked as described below:

- Every financial transaction references an accounting date and a service agreement
- Every service agreement references a service agreement type
- Every service agreement type references a GL division
- Every GL division references an accounting calendar
- The accounting calendar contains the cross-reference between the accounting date specified on the financial transaction and the related accounting period in your general ledger

The accounting date is populated on financial transactions when they are initially generated. The following points describe the source of the accounting date:

- The user who creates or cancels a bill segment online defines the accounting date as part of the generation/cancel dialog (note, the current date defaults).
- Bill segments that are produced by the BILLING background process have their accounting date defined on the bill cycle schedule that caused the bill to be created.

Copyright © 2026, Oracle and/or its affiliates.

- The user who creates or cancels an adjustment online defines the accounting date as part of the generation / cancel dialog (note, the current date defaults).
- Payments are unusual in that their financial transaction is only created when they are frozen (rather than when the payment is first distributed amongst the account's service agreements). At payment freeze time, the accounting date is set to the current date.

For payments, there is no issue because the accounting date is only populated on the financial transaction when a payment is frozen. However, for bill segments and adjustments, your business practice may dictate that the freeze date should be used as the accounting date rather than the original accounting date. Alternatively, your business practice may dictate that the accounting date that's originally stamped on bill segments / adjustments should be used (unless this associated period is closed at freeze time). It's really a question of the interpretation of the local accounting rules. After you've decided on your approach, populate the **Accounting Date Freeze Option** on Installation Options - Billing with one of the following values:

- Choose Always change if the accounting date on your financial transactions should be populated with the freeze date (i.e., the current date when the financial transaction is frozen).
- Choose Change if period is closed if the accounting date defined when the financial transaction is generated should be used (unless the associated accounting period is closed).

Please be aware of the following in respect of your choice:

- If you choose Always change:
  - When a user freezes a bill segment online, they will be prompted to supply an accounting date. The current date will default, but the user can override this value.
  - When a user freezes an adjustment online, they will be prompted to supply an accounting date. The current date will default, but the user can override this value.
  - The BILLING background process will use the current business date as the accounting date on bill segments that it freezes.
  - Also note, if you have chosen the Freeze At Bill Completion **Bill Segment Freeze Option** on the installation record, bill segments and certain types of adjustments are frozen when a bill is completed. This means that the accounting date on the related financial transactions will be set to the completion date (because the completion date is the freeze date with this setting). Refer to Preventing SA Balances And The GL From Being Impacted Until Completion for more information.
- If you choose Change if period is closed:
  - When a user freezes a bill segment online, they will only be prompted to supply an accounting date if the related accounting period is closed (because the accounting period closes after the bill segment is generated but before it's frozen). The current date will default, but the user can override this date.
  - When a user freezes an adjustment online, they will only be prompted to supply an accounting date if the related accounting period is closed (because the accounting period closes after the adjustment is generated but before it's frozen). The current date will default, but the user can override this date.
  - The BILLING background process will use the accounting date defined on the related bill cycle schedule as the accounting date on the bill segments that it creates and freezes.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The above installation option only controls the final accounting date for GL recording purposes. Rate and bill factor value selection based on accounting date uses the date as initially determined.

How Late Payment Charges Get Calculated

Late payment charges are system-generated adjustments used to penalize a customer for late (or no) payments. This section describes how to set up the tables that control how and when late payment charges are generated. The following points describe how and when late payment charges are calculated.

- When a bill is completed, the system marks it with the date on which late payment charges will be calculated if the bill is not paid.
  - This date is calculated by adding grace days to the bill's due date. Grace days are defined on the account's Customer Class / Division.
  - This date will be zero if the account's Customer Class / Division indicates the account is not eligible for late payment charge processing.
- The late payment charge background process (referred to by the batch ID of **LATEPYMT**) selects all bills on or after their late payment charge date.
  - For each such bill, the system determines if its account satisfies the late payment charge eligibility criteria defined on the account's Customer Class / Division. The eligibility criteria are defined in an algorithm and can therefore be as flexible as required.
  - If an account is eligible for late payment charges, the system checks each of the account's service agreements to determine if it is eligible for late payment charges (as defined on the service agreement's SA Type).
  - If a service agreement is eligible for late payment charges, the system calls the SA type's late payment charge calculation algorithm. This algorithm should calculate the late payment charge amount, if applicable and return the calculated amount and an appropriate adjustment type to use. If this algorithm returns this information, an adjustment is generated to levy the late payment charge.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Customer Classes for more information about how to set up an account's due days and grace period. Refer to SA Type - Main Information for more information about enabling late payment charges calculations for your service agreements.
>
> You can update the **Late Payment Charge Details** section on the Bill - Main Information page to indicate if and when late payment charges may be levied. For more information, see Bill - Main Information.

Copyright © 2026, Oracle and/or its affiliates.

Service Agreement Type Controls Everything

The previous section illustrated three important concepts:

The true financial impact of the three financial events - bills, payments, adjustments - is at the service agreement level, not at the account level. This means that bills and payments are meaningless on their own. It's the service agreements' bill segments, payment segments and adjustments that affect how much a customer owes.

- Every bill segment, payment segment, and adjustment has a related financial transaction. These financial transactions contain the double-sided journal entries that will be interfaced to your general ledger. They also contain the information defining how the customer's debt is affected by the financial event (i.e., current amount and payoff amount).
- A single bill can contain many bill segments, each of which may have a different frequency. For example, a bill could contain future charges, monthly retroactive charges based on measurement cycle, quarterly charges that must end on a quarter-end boundary.

You control the financial effects of the various financial events using a single field on the service agreement. This field is called the service agreement (SA) Type. In this section, we describe many of the tables that must be set up before you can create a SA type.

> **Note**
>
> **Foreshadowing.** You will notice that we don't explain how to set up SA types in this section. This is because SA type controls numerous aspects of a service agreement's behavior in addition to its financial behavior. The non-financial aspects are discussed in later chapters. It's only after you have set up all of the control tables in this manual that you'll be able to finally define your SA types. Refer to Setting Up SA Types for more information.

> **Warning**
>
> Take the time to define how you will record the various financial events in your general ledger before you attempt to set up these control tables. If you have simple accounting needs, this setup process will be straightforward. However, if you sell many services and use sophisticated accounting, this setup process will require careful analysis.

- Understanding CIS Divisions
- Setting Up CIS Divisions
- Setting Up Revenue Classes
- Setting Up Distribution Codes
- Setting Up Billable Charge Templates
- Designing and Defining Bill Segment Types
- Designing and Defining Deposit Classes
- Setting Up Payment Segment Types

Copyright © 2026, Oracle and/or its affiliates.

- Designing And Defining Adjustment Types

Understanding CIS Divisions

There are two types of Divisions referenced on a SA type: a CIS Division and a GL Division. This is a rather powerful structure, but it can be confusing.

- General Ledger divisions typically comprise individual entities (e.g., companies) in your general ledger. You must set up a GL division for each such entity. The GL division's sole purpose in the system is to define the accounting period associated with financial transactions linked to service agreements associated with the GL division (service agreements are associated with GL divisions via their SA type). The system cares about accounting periods in order to prevent a user from booking moneys to closed periods. It also uses accounting periods when it produces the flat file that contains the consolidated journal entry that is interfaced to your general ledger (refer to The GL Interface for more information).
- A CIS division is associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.
- CIS division is also referenced on service agreement, premise and account.
  - The CIS division on SA is actually part of the SA's SA type. Because SA type controls many business rules, all business rules that are on the SA type can be thought of as being defined for a given jurisdiction and SA type combination. For example, you could define your valid rates for electric residential service in California which differ from the valid rates for electric residential service in Nevada. Refer to Defining Service Agreement Types for more information. In addition to controlling the business rules defined on the SA's SA type, the SA's CIS division also controls the type of collection criteria used to determine if and how to collect overdue debt. Refer to Setting Up Collection Class Control for more information.
  - The CIS division on premise defines the jurisdiction in which the premise is located. This jurisdiction controls the types of service agreements that can be associated with the premise's service points (e.g., you can only link California-oriented service agreements to premises governed by the California jurisdiction). You can also set up your field activity types to execute special algorithms when a field activity is completed at a service point located in specific jurisdiction.
  - The CIS division on account when combined with the account's customer class defines the jurisdiction that governs financial business rules (e.g., the bill's due date, when and how late payment charges are calculated, etc.). Refer to Setting Up Customer Classes for more information about these rules. The CIS division on account can also play a part in the addressee of To Do entries associated with the account. To assign To Do entries to a role based on the division, simply link the To Do type to the division. Refer to To Do Entries Reference A Role for more information.
- CIS Division Overrides
- Users and CIS Division
- Restricting CIS Division
- Using CIS Division for Operational Segmentation

Copyright © 2026, Oracle and/or its affiliates.

CIS Division Overrides

Some configuration values in the system, such as those defined on the installation record, are global and applied to the entire system. CIS Division allows some of these values to be specified by CIS division, overriding the system-wide default configuration. The system processes check the CIS Division for the override before checking the default configuration. CIS Division allows for overrides to be specified for **customer class** (by person and business), and **credit rating** and **cash-only** beginning and threshold values.

The bill route type and quote route type are used to define the background process and extract algorithm for each route type. CIS Division allows overrides to be specified for the extract batch control and/or extract algorithms. These can be defined for any or all each bill route types and quote route types. While bill route type and quote route type allow many of each type to be configured, one of each type is defaulted from the installation record. This override allows CIS Division specific value for a common route type.

Users and CIS Division

When the installation option Control By CIS Division flag is set, it is necessary to associate the system users with CIS Division. This feature is applicable if your organization expects users to work on designated CIS Divisions only (as opposed to allowing users to serve customers across all CIS Divisions). Users can be associated with CIS Division(s) even if the above flag is not set. Some system functions take advantage of this such as defaulting CIS Division on control central searches. User - CIS Division is maintained in the user portal.

To navigate there, select **Admin**, and then **User**, and then **Search** and navigate to the **Miscellaneous** page.

How Does the System Use User CIS Division

Refer to Installation Options – Main for information on how this is used in conjunction with the Control By CIS Division and Premise/SP Search flags

The user's CIS Divisions are used in some circumstances to restrict case type and customer contact type by CIS Division.

The user's Primary CIS Division is used to default the CIS Division on the Control Central search that is provided with the system.

Restricting CIS Division

CIS Division appears through much of the system. The primary objects that reference a CIS Division are account, service agreement, and premise. It is also referenced on secondary objects such as orders and on pages that interact with the primary objects such as start/stop and payment arrangement.

A flag on CIS Division, Restrict to Account CIS Division, can be used to restrict service agreements and premise to the account's CIS Division. It is important to note that it is the account's CIS Division that matters. When this flag is set, many of the system pages default CIS Division and restrict the user for changing the value. The following rules are enforced in addition to the visible changes:

- Service agreement's CIS Division must match its account's CIS Division.
- Account CIS Division cannot be changed if the account has non-canceled service agreements.

Copyright © 2026, Oracle and/or its affiliates.

- Account Protect CIS Division switch is turned on and is protected. This is done because the switch becomes irrelevant since all service agreements match the account's CIS Division and the account CIS Division would not change.
- When starting service, only premises that match the account CIS Division or do not have a CIS Division maybe chosen. This rule is applied when choosing a premise via start/stop or order.
- While most SA/SPs are created by the system when service is started, it is possible to manually maintain SA/SPs. When searching for a service point from Service Agreement - SA/SP, the search is limited to SPs whose premise matches the account CIS Division or does not have a CIS Division.
- As mentioned, this setting applies to the account CIS Division. When changing a premise's CIS Division, the system checks all accounts ever linked to the premise, to see if any of the accounts' CIS Division has set the Restrict to Account CIS Division option to **Restricted**.

> **Note**
>
> The Premise to Account association is checked through the relationship from Premise to Service Point to SA/SP to Service Agreement to Account.

- If the Account's CIS Division is set to restrict the CIS Division, then the Premise's CIS Division is restricted to remain the same as the Account's CIS Division. It can also be set to blank unless the premise type requires it.
- The characteristic premise on a service agreement is restricted to premises linked to the account if the service agreement requires an SP. For other service agreements the premise search is restricted to the account CIS Division.

> **Warning**
>
> Do not turn on this flag without reviewing your existing data to ensure that it adheres to the rules enforced when the flag is set. The above rules are enforced on not just current data, but historic data as well such as closed service agreements. It is intended to ensure a consistency over the life of the objects associated with the CIS Division. This flag is configurable for each CIS Division, but consider the implications of setting it on some, but not all CIS Divisions.

Using CIS Division for Operational Segmentation

CIS Division is found throughout many of the entities in the system and there are many configuration settings related to CIS Division. When these configuration settings are used in conjunction with each other, it is possible to operate distinct business entities or jurisdictions that are segmented by CIS division. The following summary describes the different settings and uses of CIS Division that you can configure if your organization operates in multiple jurisdictions and needs to segment data and business processes by CIS Division. Details about each of these can be found in the section of the document where the functionality is configured.

Associate CIS Divisions with Users

CIS Divisions can be associated with users and one can be denoted as the primary.

Control By CIS Division

Copyright © 2026, Oracle and/or its affiliates.

Setting the Control By CIS Division flag on the installation record drives many system changes. On the account it makes CIS Division required. The flag drives functionality changes on other pages such as person, premise, and order.

Premise/SP Access

Setting the Premise/SP Search flag on the installation record restricts access to premise and SPs not link to an account. The access is restricted to the user's CIS Division(s).

CIS Division

CIS Division provides the configuration of a work calendar, default customer classes for persons and businesses, credit rating and cash only beginning and threshold values, and To Do role assignment for each CIS Division. Additionally, bill route type, quote route type, and auto pay route type configuration can be overridden.

Restrict CIS Division

The Restrict to Account CIS Division flag is a setting on CIS Division. It is used to ensure that service agreements and premises match the account CIS Division and is a main component of segmenting data by CIS Division.

Customer Class

Customer class is used to default collection class on accounts.

Customer Class Control

Customer Class Control defines many business rules directly or through algorithms. These rules are specified per CIS Division. As noted above, Customer Class can be defaulted from CIS Division so there is a lot of flexibility in the configuration of these two entities. Some of the business rules are defaults such as access group and budget plan. Others such as payment distribution and billing days until due define the rule that apply to the account Customer Class/CIS Division combination.

SA Type

CIS Division is part of the SA Type key. SA Type defines the valid values for SAs of that type and also defines attributes such as debt class and write off debt class for of SAs of that type.

Collections

As noted above, Collection Class is defaulted on the account from customer class and SAs inherit their debt class from their SA Type. Collection Class Controls are defined for the unique combination of collection class and debt class. Collection class controls define the business rule and collection processes. Through configuration, collection rules can be defined for specific CIS Divisions.

Write Off

As noted above, collection class is defaulted on the account from customer class and SAs inherit their write off debt class from their SA Type. Write off controls are defined for the unique combination of collection class and write off debt class. Write off control define the business rule and write off processes. Through configuration, write off rules can be defined for specific CIS Divisions.

Premise Type

Premise Types can control if CIS Divisions are required, optional or not allowed for premise of that type. When starting service, if the premise has a premise type, only SA Types with the same CIS Division as the premise that are identified to start for SP Types for SPs at the

Copyright © 2026, Oracle and/or its affiliates.

premise will start. This allows more generic SP Types that start SA Types specific for the premise CIS Division.

Restricting Types by CIS Division

Adjustment type, budget plan, bill cycle, case type, and customer contact type allow a CIS Division to be specified if the type is used for a specific CIS Division. Types without a CIS Division are not restricted.

Background Processing By CIS Division

Batch billing and the collection, severance, and write off background processes can all be configured to select records by CIS Division. See Choosing Which Records Are Selected for more information

Setting Up CIS Divisions

This portal is used to display and maintain CIS Divisions.

Refer to Understanding CIS Divisions for more information.

You can access the portal from **Admin**, and then **General**, and then **CIS Division**.

The following zones may appear as part of the portal's **Main** page:

- **CIS Division List:** This zone lists all CIS division records. Broadcast a record to display the details of the selected record.
- **CIS Division:** This zone provides information about the selected CIS Division.

Setting Up Revenue Classes

Every service agreement references a service agreement (SA) type. Amongst other things, the SA type defines a service agreement's revenue class. The revenue class is used when the service agreement's rate books revenue to different GL distribution codes based on the service agreement's revenue class.

NOT_SUPPORTED

See Designing Calculation Groups and Rules for more information about how revenue class is used to determine the GL revenue accounts referenced on a bill. See Revenue Segmentation for more information about how revenue class affects the number of SA types you will need.

To set up revenue classes, choose **Admin**, and then **Financial**, and then **Revenue Class**.

Description of Page

Enter an easily recognizable **Revenue Class ID** and **Description** for every revenue class.

Setting Up Distribution Codes

Copyright © 2026, Oracle and/or its affiliates.

Distribution codes simplify the process of generating accounting entries by defining valid combinations of chart of account field values.

> **NOT_SUPPORTED**
>
> Refer to The Source Of GL Accounts On Financial Transactions for more information about the accounting entries associated will bills, payments and adjustments.

To set up distribution codes, open **Admin**, and then **Financial**, and then **Distribution Code**, and then **Add.**

Description of Page

Enter a unique **Distribution Code** and **Description** for the distribution code.

If this distribution code is a holding account used for payables cash accounting, check the **Use For Non-Accrual Accounting** switch and select the accounting method from the **Accounting Method** list. Select the priority level for the distribution code from the **Accounting Priority** list and enter the actual payable **Accounting Code.** The system will transfer monies from the holding account to the distribution code when the cash event occurs. Transfers will occur based on priority and debt age. For more information, refer to Payables Cash Accounting and Deferred Accrual Accounting.

Define the **GL Account Algorithm** used by the system when it interfaces financial transactions that reference this distribution code to your general ledger (refer to GLDL - Create General Ledger Download for more information about the download process). The logic embedded in this algorithm constructs the actual GL account number. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs your general ledger account number.

The **Write Off Controls** control how the system writes off debt associated with the distribution code. Refer to The Ramifications of Write Offs in the General Ledger for an explanation of how these fields are used at write-off time.

- Define the **Division** and **SA Type** of the service agreement to which bad debt associated with this distribution code should be transferred at write-off time. Note: only SA Types with a special role of Write Off may be selected.
- When the system transfers debt to the write-off service agreement defined above, the distribution code defined on this **Division / SA Type** will be debited unless you turn on the **Override Switch**. When this switch is turned on, the system overrides the distribution code of the transfer to side of the adjustment with the distribution code associated with the debt being written off. You'd typically turn this switch on for liability distribution codes because you want to debit the original liability account when the debt is written off. Note: if this switch is on the system also overrides the characteristic type / value with the respective value associated with the debt that is being written off.

Use the **GL Account Details** scroll to define how the system constructs the GL account associated with the distribution code when it interfaces the financial transaction to your general ledger. For each distribution code, enter the following information:

- Enter the **Effective Date** of the following information.
- Define whether, on the **Effective Date**, the following information is Active or Inactive. The system will only use effective-dated information that is Active.

Copyright © 2026, Oracle and/or its affiliates.

- Enter the **GL Account** that the general ledger uses to process financial transactions tagged with this distribution code.
- Enter the **Statistics** that should be passed to the general ledger during the GL interface for this **GL Account**. For example, if this **Distribution Code** is used to record electric, residential revenue, the **Statistics** would be kWh if you record the number of kWh in you general ledger along with the dollar value of the revenue.
- If you have configured your installation options to indicate that fund accounting is practiced, define the **Fund** associated with this distribution code. If your installation options indicate that fund accounting is not practiced, the field is not visible.
- Use the characteristics grid to define characteristic values for the **Distribution Code**. This grid supports all standard characteristic types (such as, Adhoc, Predefined, File Location, and Foreign Key) from the **Distribution Code GL Account Characteristics** table. To modify a characteristic, simply move to a field and change its value. The following fields display:
  - **Characteristic Type**. Indicate the type of characteristic.
  - **Characteristic Value**. Indicate the value of the characteristic.

> **Note**
>
> You can only choose characteristic types defined as permissible on the distribution code record. Refer to Setting Up Characteristic Types & Their Values for more information.

Setting Up Billable Charge Templates

A user creates a billable charge whenever a customer should be levied an ad hoc charge. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line.

> **Note**
>
> **Interfacing billable charges from an external system.** In addition to being entered manually, billable charges can also be interfaced from an external system. You would interface billable charges if your organization provides "pass through" billing services for a service provider. Refer to Uploading Billable Charges for more information.

A billable charge must reference a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each unbilled billable charge.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.

Copyright © 2026, Oracle and/or its affiliates.

- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.

> **Note**
>
> **Rates can be applied to billable charges**. Billable charges can be connected to a service agreement that also specifies a rate. The rate will be applied and lines added to the bill segment after the billable charge lines are added. For example, a rate can insert flat charges or be applied to service quantities associated with the billable charge.
>
> **Taxes on top of billable charges.** Rates cannot be applied to billable charge lines. If you need to perform a calculation such as applying taxes on top of the existing lines, add a service quantity (SQ) that contains the taxable amount with an SQ identifier that describes it as a taxable amount. A calculation rule can apply the tax to this SQ.

> **NOT_SUPPORTED**
>
> Refer to How To Create A One-Time Invoice for instructions describing how to create a bill for a billable charge outside of the normal bill creation process.

Billable charge templates exist to minimize the effort required to create a billable charge for a customer. A billable charge template contains the default bill lines, amounts and distribution codes used to levy a one-off charge.

The information on the template may be overridden by a user when the billable charge is created. For example, you can create a billable charge template to levy tree-trimming charges. This template would contain the bill lines, amounts and distribution codes associated with a tree trimming activities. Then, when you trim a tree for a customer, a user can create a billable charge using the template and override the amount to reflect the actual amount (if it differs from the norm).

> **Note**
>
> **Templates aren't required**. A billable charge can be created without a template for a truly unexpected charge.

After setting up the billable charge templates, you must indicate the SA types that can use each template. Obviously, only billable charge SA types (as defined on the SA type's special role) will reference billable charge templates.

- Billable Charge Template - Main
- Billable Charge Template - Line Characteristics
- Billable Charge Template - SQ Details

Copyright © 2026, Oracle and/or its affiliates.

Billable Charge Template - Main

Open **Admin**, and then **Billing**, and then **Billable Charge Template**, and then **Add** to define your billable charge templates.

Description of Page

Enter a unique **Billable Charge Template** ID and **Description** for the billable charge template.

Use **Description on Bill** to define the verbiage that should print on the customer's bill above the billable charge's line item details.

Use **Currency Code** to define the currency in which the billable charge's amounts are expressed.

Use the grid to define the line item details associated with the billable charge (note, the **Total Line Amount** field is automatically calculated. It is the sum of the **Charge Amount** on each of the Line Sequence items). The following fields are required for each entry in the grid.

Sequence Line sequence controls the order in which the line items appear on the bill segment.

Description on Bill Specify the verbiage to print on the bill for the line item.

Charge Amount Specify the default amount to charge for the line item.

Show on Bill Turn this switch on if the line item should appear on the customer's printed bill. It would be very unusual for this switch to be off.

Appears in Summary Turn this switch on when the amount associated with this line also appears in a summary line.

Memo Only, No GL Turn this switch on when the amount associated with this line does not affect the GL (or the total amount owed by the customer).

Distribution Code Specify the default distribution code associated with this line item.

If you use the drill down button on the left most column in the grid, you will be taken to the Line Characteristics tab with the selected line displayed.

NOT_SUPPORTED

For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about billing billable charges, refer to How To Create A One-Time Invoice .

Billable Charge Template - Line Characteristics

Open **Admin**, and then **Billing**, and then **Billable Charge Template**, and then **Search** and navigate to the **Line Characteristics** page to define your billable charge templates line characteristics.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

The **Line Sequence** scroll defines the billable charge template line to which you wish to assign characteristic values.

The characteristics grid supports all standard characteristic types (such as, Adhoc, Predefined, File Location, and Foreign Key) from the Billable Charge Template Line Characteristics table.

To modify billable charge template line characteristics, simply move to a field and change its value. To add characteristics, press + to insert a row and then fill in the information for each field. The following fields display:

Characteristic Type The type of characteristic.

Characteristic Value The value of the characteristic.

Billable Charge Template - SQ Details

Open **Admin**, and then **Billing**, and then **Billable Charge Template**, and then **Search** and navigate to the **SQ Details** page to define your billable charge templates service quantities.

Description of Page

To modify a template's service quantity, simply move to a field and change its value. To add a new service quantity to the billable charge template, press the + button to insert a row and fill in the information for each field. The following fields display:

Sequence Specify the sequence number of the SQ.

UOM Select the unit of measure of this SQ. One or more of UOM, TOU, or SQ identifier must be selected.

TOU Select the time of use period.

SQ Identifier Select the SQ identifier.

Service Quantity Specify the number of units of this service quantity.

Designing and Defining Bill Segment Types

Every service agreement references a service agreement (SA) type. Amongst other things, the SA type references a bill segment type. The bill segment type controls how bill segments and their related financial transactions are created.

> ⚠ **Warning**
>
> We strongly recommend understanding the concepts described in The Big Picture of Billing before setting up your bill segment types.

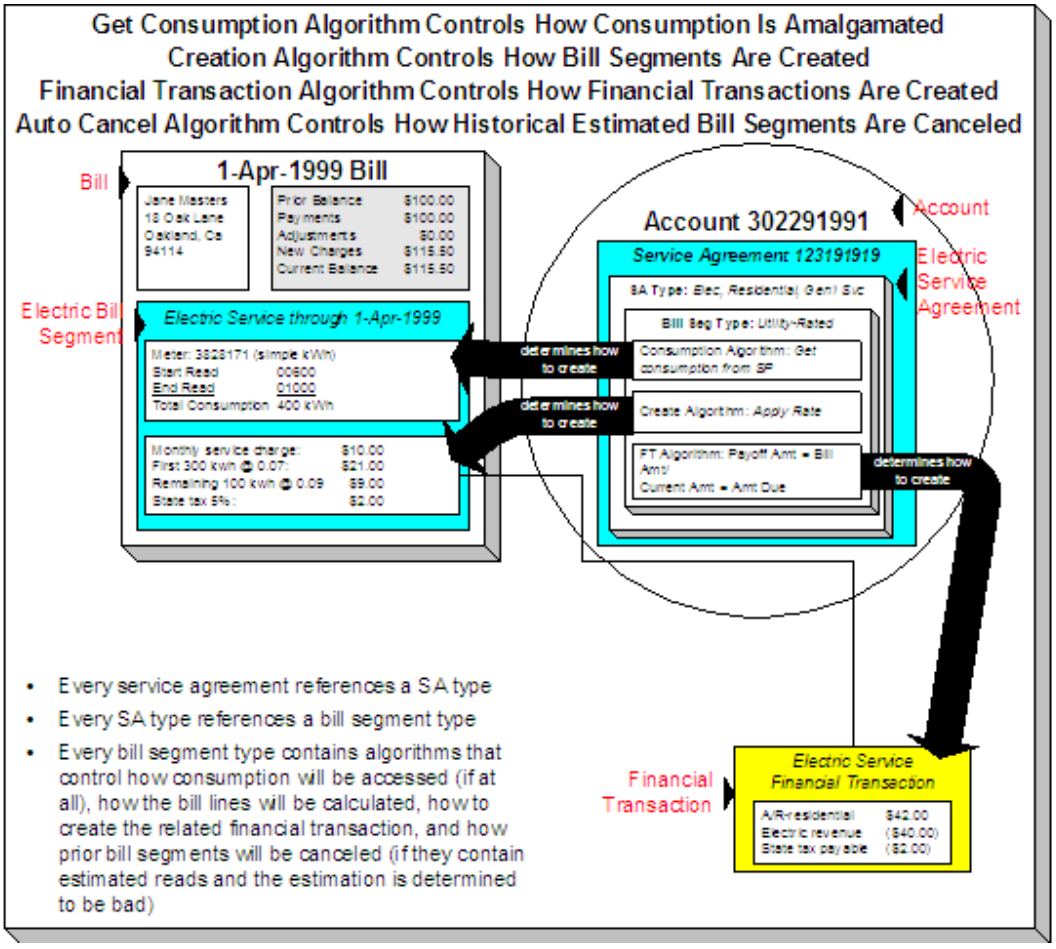
The topics in this section describe how to design and set up bill segment types.

- What Do Bill Segment Types Do?
- Designing Your Bill Segment Types
- Setting Up Bill Segment Types

Copyright © 2026, Oracle and/or its affiliates.

What Do Bill Segment Types Do?

Bill segment types control how bill segments and their related financial transactions are created. The following illustration will help you understand how the system uses bill segment types during the bill segment creation process:



Designing Your Bill Segment Types

Example Service Agreement Types

The following table contains a subset of the SA types listed under Defining Service Agreement Types and Designing Your SA Types And Start Options For Sub SAs and Designing SA Types For Service Provider Financial Settlements. However, if you are reading this document from top to bottom, you probably don't know what your SA types are (they are only designed much later) and will have to forestall this task until that time.

We're going to cheat and assume you know what your SA types are and fill in the algorithms necessary to create bill segments for each SA type. After this table is complete, we will look for unique combinations of the 4 algorithms and create a bill segment type for each one.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Before you can fill in the columns for your own SA types, you should be comfortable with the descriptions of the algorithms described under Setting Up Bill Segment Types.

Div. SA Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm
CA/G-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/G-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/G-IND	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/CABLE	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-COY	Apply Rate To Usage Request	Payoff = 0 / Current = 0	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-RESU	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/E-COMU	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
CA/WO-STD	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/WO-LIA	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/CHARITY	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/PA-REGU	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/MERCH-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/DEP-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
CA/ONETIME	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
CA/OVR UNDR	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable
CA/E-SUB ENR	Apply Rate	Payoff = Bill Amount / Current = Amount Due		

Copyright © 2026, Oracle and/or its affiliates.

CA/E-SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
CA/E-FIN SETTL	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable

Algorithm Combinations

Now, we'll extract unique combinations of the four algorithms and create a bill segment type for each.

Bill Segment Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm
BD RATED	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.
BILLCHRG	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed
COMPUSAG	Apply Rate To Usage Request	Payoff = 0 / Current = Bill Amount	Get Consumption Using Usage Request	N/A - no consumption is needed
RECUR	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
RECUR AS	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
RECURATB	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed
SUB RATE	Apply Rate	Payoff = Bill Amount / Current = Amount Due	Get Consumption From Master Bill Segment	N/A - rated sub service agreements are cancelled when their master is canceled.
SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed

Sample Service Agreement Types and Bill Segment Types

Just to make sure everything has been designed appropriately, we will return to our SA type samples and specify their respective bill segment types:

Div. SA Type	Calculation Algorithm	FT Algorithm	Consumption Algorithm	Auto Cancel Algorithm	Bill Segment Type
CA/G-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED

Copyright © 2026, Oracle and/or its affiliates.

CA/G-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/E-RES	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/E-COM	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/G-IND	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/CABLE	Apply Rate To Usage Request	Payoff = Bill Amount / Current = Amount Due	Get Consumption Using Usage Request	NA - can't estimate consumption.	BD RATED
CA/E-COY	Apply Rate To Usage Request	Payoff = 0 / Current = Bill Amount	Get Consumption Using Usage Request	N/A - no consumption is needed.	COMPUSAG
CA/WO-STD	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/WO-LIA	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/CHARITY	Recurring Charge	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR
CA/PA-REGU	Recurring Charge With Auto Stop	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR-AS
CA/MERCH-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECUR-AS
CA/DEP-I	Recurring Charge For Amount To Bill	Payoff = 0 / Current = Bill Amount	N/A - no consumption is needed	N/A - no consumption is needed	RECURATB
CA/ONETIME	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed	BILLCHRG
CA/OVR UNDR	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	
CA/E-SUB ENR	Apply Rate	Payoff = Bill Amount / Current = Amount Due	Get Consumption From Master Bill Segment	N/A - rated sub service agreements are cancelled when their master is canceled.	SUB RATE

Copyright © 2026, Oracle and/or its affiliates.

CA/E-SUB BC	Billable Charge	Payoff = Bill Amount / Current = Amount Due	N/A - no consumption is needed	N/A - no consumption is needed	SUB BC
CA/E-FIN SETTL	N/A - non billable	N/A - non billable	N/A - non billable	N/A - non billable	

And now you're ready to set up your bill segment types.

Setting Up Bill Segment Types

To set up bill segment types, open **Admin**, and then **Billing**, and then **Bill Segment Type**, and then **Add.**

Description of Page

Enter an easily recognizable **Bill Segment Type** and **Description** for every type of bill segment.

For each bill segment type, define the **Create Algorithm**. The logic embedded in this algorithm creates the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that creates a bill segment in the appropriate manner.

> **Warning**
>
> The **BS Create Algorithm** is a very important field as it controls how the system creates bill segments. There are some restrictions in respect of the values of certain fields on the SA type and the bill segment algorithm used on a SA type. Refer to Require Total Amount Switch versus Bill Segment Creation Algorithm, Allow Recurring Charge Switch versus Bill Segment Creation Algorithm, and Rate Required Switch versus Bill Segment Creation Algorithm **for more information**.

For each bill segment type, define the **Financial Algorithm**. The logic embedded in this algorithm constructs the financial transaction associated with the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs the bill segment financial transaction in the appropriate manner.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount .

If the bill segment requires consumption (e.g., meter reads) to be retrieved, define the **Get Consumption Algorithm**. The logic embedded in this algorithm retrieves the consumption that is billed on the bill segment. Refer to Designing Your Bill Segment Types for examples.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that retrieves consumption in the appropriate manner.

The **Auto Cancel Algorithm** is used by the system when it detects that a prior bill segment contains an estimated read that meets certain conditions when a non-estimated read is used on the current bill.

Examples include:

- The prior bill segment contains a bad estimated read (by "bad" we mean that the current bill has a non-estimated reading that is less than the estimated end read on the prior bill segment).
- The prior bill segment contains an estimated read (there may be situations where you want to cancel an estimated bill segment when a non-estimated read is also higher than the end read on the prior bill segment. For example, when tiered rates may cause a customer to be penalized by consumption being charged at a higher rate).

For detecting a bad estimated read on the prior bill segment, if you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that cancels bill segments that contain poorly estimated consumption.

For detecting an estimated read on the prior bill segment, if you haven’t done so already, you must set up this algorithm in the system and configure a feature configuration option. To do this:

- Create a new algorithm for the Auto Cancel Algorithm.
- On this algorithm, reference an Algorithm Type that cancels bill segments that contain estimated consumption (refer to Setting Up Algorithms).
- Configure the **Always Call Auto Cancel** option type on the Financial Transactions Option feature configuration and set the option type value to 'Y'. For more information about Feature Configurations, see Defining Feature Configurations.

Note that the **Auto Cancel Algorithm** is not used in Oracle Utilities Customer To Meter.The following features replace the algorithm:

- Usage Transaction Correction Processor
  - This is created during IMD processing when an anomaly is detected in the reads used in usage calculation. It sets the stage for sending the corrected read notification

Copyright © 2026, Oracle and/or its affiliates.

  - The Usage Transaction Correction Processor monitor job (D1-UTCRN) must be scheduled after all IMD processing batch jobs.
- Corrected Reads Notification
  - The application may be configured to automatically process all corrected read notifications by submitting the Corrected Reads Notification monitor job (C1-TCRNB). Refer to the configuration guideline below for automating this process.
  - If manual rebill is preferred, billing user may navigate to the off cycle generator portal to search for a customer's corrected read notification, review and rebill them one at a time starting from the oldest bill segment.
- Final Measurement Replacement — It is recommended that this VEE rule be configured for IMD processing. It is used to reject small changes in reads, which in turn prevents the creation and sending corrected reads notification and subsequent re-bill that would have resulted to small net differences in charges.

The **Bill Segment Information Algorithm** is used by the system to format the bill segment information that appears throughout the system. If the information you'd like displayed differs for bill segment types, you must set up this algorithm in the system. To do this:

Designing and Defining Deposit Classes

If you bill for deposits, you must set up one or more deposit classes. If your company does not bill for deposits, you can skip this section.

> **NOT_SUPPORTED**
>
> We strongly recommend familiarizing yourself with the concepts described in The Big Picture Of Deposits before tackling the information in this section.

The topics in this section describe how to design and set up deposit classes.

- What Do Deposit Classes Do?
- Designing Your Deposit Classes
- Setting Up Deposit Classes
- Setting Up Non-Cash Deposit Types

What Do Deposit Classes Do?

A deposit class contains the business rules that govern:

- How and when deposit interest is calculated.
- How the recommended deposit amount is calculated.
- When a deposit will be automatically refunded to a customer.
- When the system will recommend a new or additional deposit.

When you link a deposit class to a SA type, you are indicating that the SA type's service agreements are governed by the deposit class' business rules.

Copyright © 2026, Oracle and/or its affiliates.

In addition to linking a deposit class to the SA types used to bill for a deposit, you must also define a deposit class on SA types whose debt is covered by a deposit. Consider the following examples:

- Assume your company sells electricity, gas, and water; but deposits are only held only for electric service. In this situation, you'd need one deposit class - Electric - and you'd associate it with both the electric deposit SA type and the electric usage SA type(s) (the gas and water SA types would NOT reference a deposit class).
- If your company can apply a deposit to any type of debt, then you'd have just one deposit class - General Deposit. You'd link this deposit class to the deposit SA type, and to the other SA types whose debt is covered by the deposit.

> **Note**
>
> **Non-cash deposits.** You can also use the system to manage non-cash deposits (e.g., letters of credit, surety bonds, 3 rd party deposits). Non-cash deposits are held in respect of an account (and an account may have an unlimited number of non-cash deposits). Each non-cash deposit must reference a deposit class. Why? Because the system amalgamates cash and non-cash deposits when it determines if an account is holding an adequate deposit. Refer to 3rd Party Deposits for more information.

Designing Your Deposit Classes

A deposit class contains the business rules that govern:

- How and when deposit interest is calculated.
- How the recommended deposit amount is calculated.
- When a deposit will be automatically refunded to a customer.
- When the system will recommend a new or additional deposit.

You will need multiple deposit classes if any of the above rules / conditions differ for different types of customers. For example, if residential customers use a different recommended deposit algorithm as compared to commercial customers, you'd need one deposit class for residential and another for commercial.

You will need additional deposit classes if your customers can have multiple deposits where each deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one deposit class for regulated debt and another for unregulated debt.

We'll design deposit classes to satisfy the needs of a theoretical company to help you understand how to design your deposit classes. The following points describe the deposit requirements of our theoretical company:

- The recommended deposit amount is 2 times the average bill (averaged over the last 12 months). This is true regardless of the type of customer or debt.
- The system should automatically refund a deposit to a customer after:
  - The deposit has been held for at least 6 months; and

Copyright © 2026, Oracle and/or its affiliates.

  - The account's credit rating is greater than the credit rating threshold defined on the installation record or the account's CIS Division (i.e., the credit rating is no longer considered bad)

- This is true regardless of the type of customer or debt.
- Interest is calculated every 6 months. The interest rate is defined using a bill factor (refer to Setting Up Bill Factors for more information). This is true regardless of the type of customer or debt.
- When it's time to refund a refund a deposit, all outstanding debt will be paid off first. If any moneys remain, a check should be sent to the customer for the remainder. This is true regardless of the type of customer or debt.
- A customer could have both regulated and unregulated debt under a single account. When this happens, separate deposits will be held for each type of debt (where the regulated deposit can only be used to satisfy regulated debt and the unregulated deposit can only be used to satisfy unregulated debt).

You'd need the following deposit classes to satisfy the above requirement:

Deposit Class	Recommended Amount Rule	Auto Refund Condition	Interest Rules	Deposit Refund Method
Regulated	2 x Average Bill	Held for 6 months and credit rating is good	Simple interest every 6 months	Apply to outstanding debt first, refund remainder with a check
Unregulated	2 x Average Bill	Held for 6 months and credit rating is good	Simple interest every 6 months	Apply to outstanding debt first, refund remainder with a check

You may wonder why two deposit classes are needed when the rules are the same for both? Well, besides defining the applicable business rules for a deposit service agreement, a deposit class is defined on the SA types whose debt is covered by the deposit class' deposit. So, if you have two different types of debt where each type of debt can have its own deposit, you'd need at two deposit classes. Each deposit class would be associated with the service agreements that are being secured by the deposit.

Refer to Setting Up Deposit Classes for a description of the various algorithms defined in respect of a deposit class.

Setting Up Deposit Classes

In the previous section, Designing Your Deposit Classes, we presented a case study that illustrated a mythical organization's deposit classes. In this section, we explain how to maintain your Deposit Classes.

- Deposit Class - Main
- Deposit Class - Good Customer
- Deposit Class - Recommendation
- Deposit Class - Refund Method
- Deposit Class - Refund Criteria

Copyright © 2026, Oracle and/or its affiliates.

- Deposit Class - Refund Interest
- Deposit Class - Review Method

Deposit Class - Main

To set up deposit classes, select **Admin**, and then **Credit & Collection**, and then **Deposit Class**, and then **Add.**

Description of Page

Enter an easily recognizable **Deposit Class** and **Description**.

Use **Refund Description on Bill** to define the information that appears on the bill segment produced when it's time to refund the customer's deposit.

The remaining information on this page is used by the various deposit-oriented processes.

Refer to Deposit Class - Good Customer for information about the **Good Customer Algorithm**.

Refer to Deposit Class - Recommendation for information about the **Recommendation Algorithm** and **Review Tolerance Percentage**.

Refer to Deposit Class - Refund Method for information about the **Refund Method Algorithm**.

Refer to Deposit Class - Refund Criteria for information about the **Refund Criteria Algorithm**.

Refer to Deposit Class - Refund Interest for information about the **Interest Refund Algorithm** and **Months Between Interest Refund**.

Refer to Deposit Class - Review Method for information about the **Review Method Algorithm**.

Deposit Class - Good Customer

On Deposit Class - Main you must define the **Good Customer Algorithm** used by the system when it determines if a customer is considered good (the system recommends new / additional deposits for bad customers). Refer to Deposit Review for a description of the background process that reviews customers for adequate deposits.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if a customer is considered good.

Deposit Class - Recommendation

On Deposit Class - Main you must define the **Recommendation Algorithm** used by the system when it calculates a customer's recommended deposit amount.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that recommends deposits.

Copyright © 2026, Oracle and/or its affiliates.

The system uses the **Review Tolerance Percentage** to prevent the recommendation of small deposits by the Deposit Review background process. For example, if this field contains 10(%), the system would only recommend an additional deposit if the recommended amount was more than 10% of the existing deposit.

Deposit Class - Refund Method

On Deposit Class - Main you must define the **Refund Method Algorithm** used by the system when it refunds a deposit to the customer.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that refunds a deposit to a customer.

Deposit Class - Refund Criteria

On Deposit Class - Main you must define the **Refund Criteria Algorithm** used by the system when it determines if it should automatically refund a deposit to a customer. Refer to Deposit Review for a description of the background process that reviews deposits for refunds.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if a customer qualifies for a deposit refund.

Deposit Class - Refund Interest

On Deposit Class - Main you must define the **Interest Refund Algorithm** to define how the system calculates interest and how it refunds the interest to the customer.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Defining Algorithms).
- On this algorithm, reference an Algorithm Type that calculates interest on a deposit.

Interest will be automatically calculated every X months where X is defined in **Months Between Interest Refund**. Refer to Deposit Interest for a description of the background process that calculates interest on deposits. Also note that interest is calculated when a deposit service agreement is stopped.

Deposit Class - Review Method

On Deposit Class - Main you must define the **Review Method Algorithm** used by the system to determine what action to take if the system recommends a deposit (or additional deposit) amount for an account. Refer to Review Deposits for a description of the background process that reviews deposits for refunds. The algorithm supplied with the base product highlights new deposits and deposit amounts on the Deposit Review page.

If you haven't done so already, you must set up this algorithm in the system. To do this:

Copyright © 2026, Oracle and/or its affiliates.

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the review method the system uses if it recommends a deposit or additional deposit be applied to an account.

Setting Up Non-Cash Deposit Types

Non-cash deposit types are used to indicate the type of monetary instrument used for non-cash deposits. Refer to Non-Cash Deposits for more information.

To define your non-cash deposit types, select **Admin**, and then **Credit & Collection**, and then **Non-Cash Deposit Type.**

Description of Page

To modify a non-cash deposit type, move to a field and change its value.

To add a new non-cash deposit type, insert a row, then fill in the information for each field. The following fields display:

Non-Cash Deposit Type The unique identifier of the non-cash deposit type.

Description The description of the non-cash deposit type.

Review Before Expiration This switch indicates if the system will create a To Do entry when non-cash deposits of this type are close to expiration.

Third Party Deposit This switch indicates if the system requires a reference to a 3rd party's deposit service agreement for this type of non-cash deposit. Refer to 3rd Party Deposits for more information.

Setting Up Payment Segment Types

Every service agreement references a service agreement (SA) type. Amongst other things, the SA type references a payment segment type. The payment segment type controls how payment segments and their related financial transactions are created. To set up payment segment types, open **Admin**, and then **Payment Segment Type.**

Description of Page

Enter an easily recognizable **Payment Segment Type** and **Description** for every type of payment segment.

> **NOT_SUPPORTED**
>
> For more information about the source of the distribution codes on financial transactions, see The Source Of GL Accounts On Financial Transactions .

For each payment segment type, define the **Payment Segment Fin Algorithm**. The logic embedded in this algorithm constructs the actual financial transaction associated with the payment segment. Refer to Examples of Common Payment Segment Types for examples of how algorithms are used on common payment segment types.

If you haven't done so already, you must set up this algorithm in the system. To do this:

Copyright © 2026, Oracle and/or its affiliates.

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that constructs the payment segment financial transaction in the appropriate manner.

> **NOT_SUPPORTED**
>
> For more information about current and payoff amount, see Current Amount versus Payoff Amount .

- Examples of Common Payment Segment Types

Examples of Common Payment Segment Types

The following table shows several classic payment segment types used by many organizations:

Payment Segment Type	Payment Segment Financial Transaction Algorithm
Normal payment (if you practice accrual accounting). Refer to Accrual versus Cash Accounting for more information.	Payoff = Pay Amount / Current = Pay Amount (no cash accounting)
Normal payment (if you practice cash accounting). Refer to Accrual versus Cash Accounting for more information.	Payoff = Pay Amount / Current = Pay Amount (plus Cash Accounting)
Charity payment	Payoff =0 / Current = Pay Amount (the GL is affected)
Non-CIS Payments (When the FT is created, the distribution code and GL account to credit is retrieved from the pay). Refer to Non CIS Payments for more information	Payoff = Pay Amount / Current = Pay Amount (no cash accounting)

Designing And Defining Adjustment Types

A service agreement's debt may be changed with an adjustment. Every adjustment must reference an adjustment type. The adjustment type contains a great deal of information that is defaulted onto the adjustment, including whether the adjustment amount is calculated. It also controls many business processes associated with the adjustment. The topics in this section describe how to design and set up adjustment types.

- What Do Adjustment Types Do?
- Setting Up Adjustment Types
- Setting Up Adjustment Type Profiles
- The Big Picture Of Adjustment Approval
- Setting Up Approval Profiles

Copyright © 2026, Oracle and/or its affiliates.

What Do Adjustment Types Do?

An adjustment type contains the business rules that govern how its adjustments are managed by the system. Please refer to The Big Picture Of Adjustments for a complete description of how adjustment types impact the lifecycle of adjustments.

Setting Up Adjustment Types

The topics in this section describe how to set up adjustment types.

> **Note**
>
> **When a new adjustment type is added.** When you introduce a new adjustment type, you must update one or more adjustment profiles with the new adjustment type. Why? Because adjustment profiles define the adjustment types that may be levied on service agreements (adjustment profiles are defined on SA types). If you don't put the adjustment type on an adjustment profile, the adjustment type can't be used on any adjustment.

- Adjustment Type - Main
- Adjustment Type - Adjustment Characteristics
- Adjustment Type - Algorithms
- Setting Up Calculated Adjustment Types
- Examples of Common Adjustment Types

Adjustment Type - Main

To set up adjustment types, open **Admin**, and then **Financial**, and then **Adjustment Type**, and then **Add.**

Description of Page

Enter a unique **Adjustment Type** ID and **Description** for the adjustment type.

If your implementation requires restricting the list of adjustments types that users are able to select by CIS Division, specify a **CIS Division** on the adjustment type.

The **Adjustment Amount Type** indicates whether the adjustment about is calculated or not. Select Calculated Amount when you want to use a rate to perform calculations to generate the adjustment amount otherwise select Non-Calculated Amount. Refer to Setting Up Calculated Adjustment Types for more information about calculated adjustments.

Enter the **Distribution Code** that references the GL account associated with the adjustment. For example, if this adjustment type is used to levy a charge for a bad check, the distribution code would reference the revenue account to which you associate such revenue. Note, the offsetting distribution code is kept on the SA type.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Distribution Code for Calculated Adjustments.** Depending on the algorithm used for the calculated adjustment , the distribution code may come from the adjustment type or the calculation lines of the algorithm. If the adjustment's calculation algorithm gets the distribution code from the calculation lines, you do not need to specify a distribution code on the adjustment type.

> **NOT_SUPPORTED**
>
> For more information about the source of the distribution codes on financial transactions, see The Source Of GL Accounts On Financial Transactions .

Enter the **Currency Code** for adjustments of this type.

Turn on **Sync. Current Amount** if adjustments of this type exist to force a service agreement's current balance to equal its payoff balance. These types of adjustments are issued before a service agreement's funds are transferred to a write-off service agreement. If this switch is on, choose an **Adjustment Fin Algorithm** that does not impact payoff balance or the GL, but does affect the SA's current balance (refer to ADJT-CA for an example of such an algorithm).

Enter a **Default Amount** if an amount should be defaulted onto adjustments of this type.

> **NOT_SUPPORTED**
>
> For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount .

If the AP Adjustment should be recorded in respect of the customer's 1099 amounts, indicate the **A/P 1099 Flag**. This would typically be used on the adjustment used to credit the deposit service agreement with accrued interest. The values of this field are Interest and Miscellaneous. This type of adjustment would also have an **A/P Request Type Code** selected, as 1099 reporting is handled in A/P.

Turn on **Print By Default** if information about adjustments of this type should print on the account's next bill.

Choose an **A/P Request Type Code** if this adjustment is interfaced to accounts payable (i.e., it's used to send a refund check to a customer). Refer to A/P Check Request for more information.

The **Adjustment Freeze Option** defines when adjustments can be frozen and therefore when a service agreement's balance and the general ledger are affected by an adjustment. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion to understand the significance of this option. Also note, if the installation option's Bill Segment Freeze Option is Freeze At Will, this field is defaulted to Freeze At Will and cannot be changed.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Adjustment types for adjustments created during bill completion (e.g., by a bill completion algorithm) must have their adjustment freeze option set to Freeze At Will. Otherwise (i.e., if the option is Freeze At Bill Completion) they will not be frozen until a subsequent bill is completed.

If adjustments of this type require approval, define an **Approval Profile**. For more information, refer to The Big Picture of Adjustment Approvals.

Enter the verbiage to appear on the printed bill in **Description on Bill**.

Use the characteristics collection to define a **Characteristic Type** and **Characteristic Value** common to all adjustments of this type. These can be used for reporting purposes or in your algorithms.

Adjustment Type - Adjustment Characteristics

To define characteristics that can be defined for adjustments of a given type, open **Admin**, and then **Financial**, and then **Adjustment Type**, and then **Search** and navigate to the **Adjustment Characteristics** page.

Description of Page

Use the **Adjustment Characteristics** collection to define characteristics that can be defined for adjustments of a given type. Turn on the **Required** switch if the **Characteristic Type** must be defined on adjustments of a given type. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on. Use **Sequence** to control the order in which characteristics are defaulted.

Adjustment Type - Algorithms

To define algorithms for adjustments, open **Admin**, and then **Financial**, and then **Adjustment Type**, and then **Search** and navigate to the **Algorithms** page.

Description of Page

The grid contains **Algorithms** that control important adjustment functions. If you haven't already done so, you must set up the appropriate algorithms in your system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description

Copyright © 2026, Oracle and/or its affiliates.

Adjustment Cancellation	Optional	When an adjustment is canceled an algorithm of this type may be called to do additional work. Refer to The Lifecycle Of An Adjustment for more information about canceling an adjustment.
Adjustment Freeze	Optional	When an adjustment is frozen an algorithm of this type may be called to do additional work. Refer to The Lifecycle Of An Adjustment for more information about freezing an adjustment.
Adjustment Information	Optional	We use the term "Adjustment information" to describe the basic information that appears throughout the system to describe an adjustment. The data that appears in "Adjustment information" is constructed using this algorithm. Plug an algorithm into this spot to override the "Adjustment information" algorithm on installation options or the system default "Adjustment information" if no such algorithm is defined on installation options.
Adjustment FT Creation	Required	Algorithms of this type are used to construct the actual financial transaction associated with the adjustment. The financial transaction controls the adjustment's affect on the service agreement's payoff and current balances. It also constructs the information that is eventually interfaced to your general ledger. Refer to Examples of Common Adjustment Types for examples of how algorithms are used on common adjustment types.
Default Adjustment Amount	Optional	Algorithms of this type are used to default the adjustment amount. Refer to Default the Adjustment Amount for more information.

Copyright © 2026, Oracle and/or its affiliates.

Determine SA	Optional	Algorithms of this type are used to find a service agreement for which the adjustment can be posted. This plug-in is used particularly during adjustment upload when a staging record does not identify the SA ID. Refer to Interfacing Adjustments From External Sources for more information.
Resolve Suspense	Optional	Algorithms of this type are used to automatically resolve adjustments that are in suspense. Refer to Suspense Adjustments for more information
Generate Adjustment	Optional	Algorithms of this type are used to calculate the adjustment amount if an adjustment type indicates that the adjustment amount is calculated. Refer to Setting Up Calculated Adjustment Types for more information.
Validate Adjustment	Optional	Algorithms of this type are used to validate information for the adjustment after it is generated.

Setting Up Calculated Adjustment Types

You can use an algorithm to calculate an adjustment amount for example if you need to calculate tax on a base amount or calculate a non-sufficient funds charge based on the customer's credit rating. Because the base package algorithm calculates adjustment amounts by calling the rate application, calculated adjustments are sometimes referred to as ratable adjustments.

> **Note**
>
> **Ratable Adjustments Appear Deceptively Simple.** But, they are not. Calculated adjustments that use the base package algorithm have all the power and flexibility (and complexity) of the rate application. Anything that you can do with a rate can be applied to a calculated adjustment. For examples that illustrate the flexibility of the rate application (and therefore calculated adjustments), refer to Rate Examples.

Adjustment types that indicate they are calculated have a generate adjustment algorithm. The base package algorithm defines the rate schedule used to calculate the adjustment as well as any UOM, TOU or SQI parameters.

To set up calculated adjustment types using the base package generate adjustment algorithm type:

Copyright © 2026, Oracle and/or its affiliates.

- Define the rate that performs the calculations, including the rate schedule, calculation groups, and calculation rules. Refer to Rates for information.

> **Note**
>
> If you create your own Generate Adjustment algorithm type, you may not need to set up a rate that performs the calculations. It depends on the needs of your algorithm type.

- Create a Generate Adjustment algorithm (refer to Setting Up Algorithms) that references the base package algorithm type that generates calculated adjustments (see the table above).
- If you want the generation algorithm's calculation lines to provide the distribution codes when the adjustment is posted to the GL, create an Adjustment Financial Transaction algorithm (refer to Setting Up Algorithms) that references an algorithm type that creates the adjustment's financial transactions using the calculation lines. A parameter of the adjustment financial transaction algorithm determines whether the distribution codes are taken from the adjustment type (AT) or calculation lines (CL). The system comes supplied with several sample algorithm types that create adjustment financial transactions.
- Create an adjustment type where the **Adjustment Amount Type** is Calculated Amount, the Generate Adjustment event references the generation algorithm created above, and the Adj. Financial Transaction event references the adjustment financial transaction algorithm created above.

Examples of Common Adjustment Types

The following table shows several classic adjustment types used by many organizations:

Adjustment Type	Typical Distribution Code	Default Amount	Financial Transaction Algorithm	Print On Bill	A/P Adjust-ment	1099
NSF	NSF revenue	Your NSF fee	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
LPC	Late payment charge revenue	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No

Copyright © 2026, Oracle and/or its affiliates.

CONNECT	Connection charge revenue	Your connection charge	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
CUSTREL	Customer relationship expense	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
ADDCHARG	Misc Revenue	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
XFER	Balance transfer clearing	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	No	No
WO SYNC	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	No	No	No
REFUNDAP	A/P clearing	N/A	Payoff = Current = Adj. Amt Refer to ADJT-NM for an example of such an algorithm type	Yes	Yes	No

Copyright © 2026, Oracle and/or its affiliates.

DPA FIX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
CHARITFX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG ON	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG OFF	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
BUDG FIX	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No
DEPOSREF	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	Yes	No	No

Copyright © 2026, Oracle and/or its affiliates.

DEPOSINT	Interest expense	N/A	Payoff = Current = Adj. Amt. Refer to ADJT-NM for an example of such an algorithm type. Or Payoff Amt = Adj Amt / Current Amt = 0. Refer to ADJT-TA for an example of such an algorithm type. Use the first method if you want to have the interest reflected as a credit balance on the customer's bill. Use the second method if you roll the interest amount into the customer's existing deposit on hand.	Yes	No	Yes
DEPFIXCR	N/A	N/A	Payoff = 0 / Current = Adj. Amt Refer to ADJT-CA for an example of such an algorithm type	No	No	No

Setting Up Adjustment Type Profiles

Adjustment type profiles categorize your adjustment types into logical groups. When you link a profile to a SA type, you limit the type of adjustments to be linked to the SA type's service agreements. The creation of adjustment profiles and their linkage to SA types prevents

Copyright © 2026, Oracle and/or its affiliates.

inappropriate adjustments from being linked to your service agreements. More than one adjustment type profile may be linked to a SA type.

For example, you can create an adjustment type profile called Miscellaneous Fees and link to it the miscellaneous fee adjustment types. Then, you would link this profile to those SA types that are allowed to levy such fees.

> **Note**
>
> **Bottom line.** An adjustment can only be linked to a service agreement if its adjustment type is part of an adjustment type profile that is valid for the service agreement's SA type. If an adjustment type is not linked to a profile, it could never be levied.

To set up adjustment type profiles, open **Admin**, and then **Financial**, and then **Adjustment Type Profile**, and then **Add.**

Description of Page

Enter a unique **Adjustment Type Profile** and **Description** for the adjustment type profile.

Indicate the **Adjustment Types** that are part of the profile.

- Examples Of Common Adjustment Profiles

Examples Of Common Adjustment Profiles

The following table shows several classic adjustment profiles used by many organizations (we've displayed some attributes from the adjustment type in the following table to help make it more understandable):

Adjustment Profile	Adjustment Type	Typical Distribution Code	Default Amount	Financial Transaction Algorithm
FEES	NSF	NSF revenue	Your NSF fee	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	LPC	Late payment charge revenue	Your LPC	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	CONNECT	Connection charge revenue	Your connection charge	Payoff Amt = Adj Amt / Current Amt = Adj Amt
MISCEXP	CUSTREL	Customer relationship expense	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
XFER	TRANSBAL	Balance transfer clearing	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
REFUND	REFUND	A/P clearing	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt

Copyright © 2026, Oracle and/or its affiliates.

DPA	ADJCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	SYNCCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
CHARITY	CHAR FIX	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
BUDGET	BUDG ON	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	BUDG OFF	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	BUDG FIX	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	FIX PAY	Customer relationship expense	N/A	Payoff Amt = Adj Amt / Current Amt = 0
DEPOSIT	DEPOSBILL	N/A	Your standard deposit amount	Payoff Amt = 0 / Current Amt = Adj Amt
	DEPOSINT	Interest expense	N/A	Payoff Amt = Adj Amt / Current Amt = Adj Amt
	SYNCCURR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt
	ADJCUR	N/A	N/A	Payoff Amt = 0 / Current Amt = Adj Amt

The Big Picture Of Adjustment Approval

Some implementations require adjustments to be approved by one or more managers before they impact a customer's debt and the general ledger. For example, an adjustment used to rebate a credit balance may require managerial approval before the rebate is sent to the customer. The topics in this section describe how to set up the system to support the approval of adjustments.

- Approval Is Controlled By Approval Profiles
- Approval Profiles Can Be Linked To Multiple Adjustment Types
- Adjustments Created In Batch Are Not Approved
- Approval Inserts A Step Into An Adjustment's Lifecycle
- Approval Requests Manage And Audit The Approval Process
- To Do Entries Are Created To Notify Approvers
- Monitoring and Escalating Approval Requests

Copyright © 2026, Oracle and/or its affiliates.

- Rejecting Deletes The Adjustment
- Designing Your Approval Profiles
- Implementing Other Approval Paradigms

Approval Is Controlled By Approval Profiles

An approval profile contains the rules that define if and how an adjustment is approved. If an adjustment type does not reference an approval profile, the related adjustments do not require third-party approval before they impact a customer's debt. If an adjustment type references an approval profile, the approval profile's approval hierarchy defines if the adjustment requires approval and who the authorized approvers are. For example, an approval profile can be configured with the following approval hierarchy:

- Adjustments < $0 require approval by the "credit approvers role"
- Adjustments >= $0 and <= $10 do not require approval
- Adjustments > $10 and <= $100 require the approval of a user that belongs to the "level 1 approvers role"
- Adjustments > $100 require two levels of approval: first a user that belongs to the "level 1 approvers role" must approve the adjustment; afterwards, the adjustment must be approved by a user that belongs to the "level 2 approvers role"

> **Note**
>
> **Transfer adjustments.** The term "transfer adjustment" refers to two adjustments that are used to transfer moneys between two service agreements. The adjustment with the positive amount is considered to be the debit adjustment; the other adjustment is considered the credit adjustment. When a transfer adjustment requires approval, only one of the adjustments needs to be approved. You control whether the debit side or the credit side of a transfer adjustment is used to control the approval process when you set up the approval profile.

Approval Profiles Can Be Linked To Multiple Adjustment Types

Approval hierarchies are frequently the same for many adjustment types. The system allows an approval profile to be linked to multiple adjustment types to simplify the definition and maintenance of the rules over time.

Adjustments Created In Batch Are Not Approved

The system assumes that no approval is necessary for adjustments created by batch processes even those whose adjustment type references an approval profile.

Approval Inserts A Step Into An Adjustment's Lifecycle

Copyright © 2026, Oracle and/or its affiliates.

The Lifecycle Of An Adjustment explains how an adjustment is transitioned from the Freezable state to the Frozen state when it should impact the general ledger and the customer's balance. If an adjustment's adjustment type references an approval profile, the user cannot freeze the adjustment directly. Rather, the user must submit the adjustment for approval when it's ready and only when the last applicable approver approves the adjustment will it become Frozen.

> **Note**
>
> **Freeze during bill completion.** You can configure the system to only freeze certain types of adjustments when the next bill is completed for the adjustment's account. When the last approver approves such adjustments, they remain in the Freezable. When the next bill is completed for the account, these adjustment become Frozen. Such adjustments that have not been approved at the time of bill completion will remain in the Freezable state. Refer to Preventing SA Balances And The GL From Being Impacted Until Bill Completion for more information.

Approval Requests Manage And Audit The Approval Process

Users submit an adjustment for approval using a dedicated button on the Adjustment page. When an adjustment is submitted for approval, the system creates an "approval request". The approval request determines if the adjustment requires approval and, if so, the list of approvers. If the adjustment does not require approval, the approval request is updated to indicate such and the adjustment is Frozen immediately (if freezing is allowed prior to bill completion). If the adjustment requires approval, the approval request's state becomes Approval In Progress and the approver(s) are notified.

> **Note**
>
> **Approval submission logic is customizable.** The previous paragraph describes how the base-package works when an adjustment is submitted for approval. This logic resides in an algorithm that's plugged in on the C1-AdjustmentApprovalProfile business object in the Determine Approval Requirements system event. Your implementation can change this logic by developing a new algorithm and plugging it into this business object. If your logic is meant to supersede the base-package algorithm, remember to inactivate the base-package algorithm by adding an appropriate inactivation option to this business object.

To Do Entries Are Created To Notify Approvers

When an approval request detects an adjustment requires approval, it notifies the first approver by creating a To Do entry. The To Do entry is created using the To Do type and To Do role defined on the approval profile. All users who belong to the approving To Do role can see the entry. When a user drills down on an adjustment approval To Do entry, the Adjustment - Approval portal is opened. This portal contains summary information about the adjustment and the approval history of the adjustment. This portal is also where the user approves or rejects the adjustment.

Copyright © 2026, Oracle and/or its affiliates.

When the first user in the To Do role approves an adjustment, the To Do entry is Completed and the approval request's audit log is updated. If there are no higher levels of approval required, the adjustment is Frozen (if freezing is allowed prior to bill completion) and the approval request is moved to the Approved state. If there are higher levels of approval required, a new To Do entry is created to the next To Do role in the approval hierarchy.

> **Note**
>
> **To Do entries can create email.** A To Do entry can be configured to create an email message for every user in the To Do role to inform the user(s) of new adjustments requiring their attention. Refer to To Do Entries May Be Routed Out Of The System for the details.

Monitoring and Escalating Approval Requests

The base-package is supplied with an algorithm that your implementation can use to monitor approval requests that have been waiting too long for approval. This algorithm can complete the current To Do entry and create a new one for a different role when the timeout threshold defined on the algorithm's parameters is exceeded. If you've configured the system to send email for approval, this algorithm can also send x reminder emails (where x is defined on the algorithm's parameters) before the approval request is escalated to the new To Do role. Refer to C1-APR-TMOUT for more information about this algorithm. If you plan to enable this functionality, plug-in your configured algorithm on the Approval In Progress state on the C1-AdjustmentApprovalRequest business object.

Rejecting Deletes The Adjustment

When an adjustment is being approved, anyone with access to the adjustment can reject it by using the Adjustment - Approval portal. Users other than the current approver are allowed to reject an adjustment to allow an "in process" an adjustment to be withdrawn.

When an adjustment is rejected, the following takes place:

- The user is prompted for a reject reason.
- The approval request's audit log is updated with the reject reason and the approval request is moved to the Rejected state.
- The adjustment is deleted.

Designing Your Approval Profiles

The following points describe a recommended design process:

- Create logical groups of adjustment types where each group has the same monetary hierarchy and approvers. An approval profile will be required for each of these groups.
- The number of To Do types (if any) that need to be created is dependent on how the adjustment approval To Do entries should be organized on To Do lists. For example, if all approval request To Do entries can appear in the same To Do list, you can use the base-package adjustment approval To Do type. However, if your organization prefers each

Copyright © 2026, Oracle and/or its affiliates.

approval profile's To Do entries to appear in a distinct To Do list, a separate To Do type will be needed for each list. Note that the base-package is supplied with a To Do type called C1-ADAPP that should be used as the basis for any new approval request To Do type.

- The number of To Do roles is dependent on who approves your adjustments. At a minimum, you will require a separate To Do role for each level in your approval profiles. Remember that every user in a To Do role will see its entries (and receive email if you've configured the system to do such).
- Refer to Monitoring and Escalating Approval Requests for how to configure the system to escalate approval requests that have been waiting too long.
- If your implementation requires email notification when an adjustment requires approval, the following setup is required:
  - Set up an outbound message type, external system, and message sender. Refer to To Do Entries May Be Routed Out Of The System for the details.
  - Every To Do type referenced on your approval profiles should be configured as follows:
- Define the F1-TDEER batch process as the To Do type's routing process
- Set up an algorithm that references the C1-ADJAREQEM algorithm type and plug it in the External Routing system event.

Implementing Other Approval Paradigms

The above sections describe how the base-package adjustment approval process works. Because adjustment approval has been implemented using the C1-AdjustmentApprovalProfile and the C1-AdjustmentApprovalRequest business objects, your implementation can add additional business rules and change the approval user interface as required. Alternatively, if your implementation has a radically different approval process, you can create a different business objects with their own business rules. To learn how to do this, please enroll in the Configuration Tools training class.

Setting Up Approval Profiles

Approval profiles contain the rules that control how adjustments are approved. To set up an approval profile, open **Admin**, and then **Financial**, and then **Approval Profile**, and then **Add**.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Adjustment Approval for a detailed description of how approval profiles govern the adjustment approval process.

The topics in this section describe the base-package zones that appear on the Approval Profile portal.

- Approval Profile List
- Approval Profile
- Approval Profile's Adjustment Types

Copyright © 2026, Oracle and/or its affiliates.

Approval Profile List

The Approval Profile List zone lists every approval profile. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent approval profile.
- Click the Add link in the zone's title bar to add a new approval profile.

Approval Profile

The Approval Profile zone contains display-only information about an approval profile. This zone appears when an approval profile has been broadcast from the Approval Profile List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the approval profile.
- Click the **Delete** button to start a business process that deletes the approval profile.
- Click the **Duplicate** button to start a business process that duplicates the approval profile.

Please see the zone's help text for information about this zone's fields.

Approval Profile's Adjustment Types

The Approval Profile's Adjustment Types zone lists every adjustment type that is governed by this approval profile. This zone appears when there is at least one adjustment type governed by the approval profile displayed in the Approval Profile zone:

To add an adjustment type to this list:

- Navigate to the Adjustment Type page and display the desired adjustment type.
- Specify the governing approval profile and update the adjustment type.

To remove an adjustment type from this list:

- Navigate to the Adjustment Type page and display the desired adjustment type.
- Change or remove its approval profile and update the adjustment type.

Designing and Defining Budget Plans

If you allow your customers to pay a budget amount each month (as opposed to their actual bill amount), you must set up one or more budget plans. If your company does not offer budget billing options, you can skip this section.

The topics in this section describe how to design and set up budget plans.

- The Financial Impact Of Budget Plans
- What Do Budget Plans Do?

Copyright © 2026, Oracle and/or its affiliates.

- Designing Your Budget Plans
- Setting Up Budget Plans

The Financial Impact Of Budget Plans

The only difference between a customer who participates in budget billing and one who doesn't is that budget billing customer have bill segments where payoff amount differs from current amount. Why? Because the payoff amount is the actual amount of the bill. The current amount is the amount the customer is expected to pay (i.e., their budget amount).

Let's run through an example of a customer on a budget to illustrate a service agreement where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-19	Bill: $125, Budget $150	125	150
15-Jan-19	Payment: $150	-25	0
2-Feb-19	Bill: $175, Budget $150	150	150
14-Feb-19	Payment: $150	0	0
3-Mar-19	Bill: $200, Budget $150	200	150
15-Mar-19	Payment: $150	50	0

NOT_SUPPORTED

For more information about current and payoff amounts, refer to Current Amount versus Payoff Amount.

What Do Budget Plans Do?

A budget plan contains the business rules that govern:

- How the recommended budget amount is calculated.
- When and how a customer on an ongoing budget plan will have their budget amount periodically trued up.
- The conditions under which the system will highlight an existing budget amount as being anomalous with the customer's current use patterns.

You may have different budget plans for different customer segments. For example, customers with large bills may have their budget amount recalculated every month, whereas small customers may have their budget amount only recalculated annually. You define which budget plans govern a customer's bills via a **budget plan on the customers' accounts**. An account's initial budget plan is defaulted from its customer class. You may override an account's budget plan at will.

Copyright © 2026, Oracle and/or its affiliates.

Designing Your Budget Plans

NOT_SUPPORTED

Refer to Budget Billing for background information about budget billing.

A budget plan contains the business rules that govern:

- How the recommended budget amount is calculated.
- When and how a customer on an ongoing budget plan will have their budget amount periodically trued up.
- The conditions under which the system will highlight an existing budget amount as being anomalous with the customer's current use patterns.

You will need multiple budget plans if any of the above rules / conditions differ for different types of customers. For example, if residential customers use a different recommended budget algorithm as compared to commercial customers, you'd need one budget plan for residential and another for commercial.

We'll design budget plans to satisfy the needs of a theoretical company to help you understand how to design your budget plans. The following points describe the budget requirements of our theoretical company:

- The recommended budget amount is the last year's real bill amounts plus any existing debit/credit balance divided by 12. This is true regardless of the type of customer.
- The frequency of budget true up is monthly for commercial customers and annually for residential customers.
- The system should highlight when a residential customer's budget is more than 30% out of whack with what their budget amount would be if it was recalculated.
- The system should highlight when a commercial customer's budget is more than 20% out of whack with what their budget amount would be if it was recalculated.

You'd need the following budget plans to satisfy the above requirement:

Budget plan	Recommended Amount Algorithm	True Up Algorithm	Monitor Algorithm
Residential	Average Bill	True up every 12 months	Highlight when more than 30% out
Commercial	Average Bill	True up every month	Highlight when more than 20% out

Refer to the Page Controls under Setting Up Budget Plans for a description of the various algorithms defined in respect of a budget plan.

Setting Up Budget Plans

Copyright © 2026, Oracle and/or its affiliates.

In the previous section, Designing Your Budget Plans, we presented a case study that illustrated a mythical organization's budget plans. In this section, we explain how to maintain your Budget Plans.

- Budget Plan - Main
- Budget Plan - Calculation Algorithm
- Budget Plan - Monitor Algorithm
- Budget Plan - True Up Algorithm

Budget Plan - Main

To set up budget plans, select **Admin**, and then **Credit & Collection**, and then **Budget Plan**, and then **Add.**

Description of Page

Enter an easily recognizable **Budget Plan** and **Description**.

If your implementation requires restricting budget plans to accounts by CIS Division, specify a **CIS Division** on the budget plan.

The remaining information on this page is used by the various budget-oriented processes.

Refer to Budget Plan - Calculation Algorithm for information about the **Calculation Algorithm**.

Refer to Budget Plan - Monitor Algorithm for information about the **Monitor Algorithm**.

Refer to Budget Plan - True Up Algorithm for information about the **True Up Algorithm** and **Months for True Up**.

Budget Plan - Calculation Algorithm

On Budget Plan - Main you must define the **Calculation Algorithm** used by the system when it calculates a customer's recommended budget amount.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that calculates recommended budget amounts.

Budget Plan - Monitor Algorithm

On Budget Plan - Main you must define the **Monitor Algorithm** used by the Budget Monitor background process when it determines if a customer's budget plan is out-of-sync with their consumption patterns.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **What happens?** If the algorithm determines that a customer's budget plan is out-of-sync with its current recommended amount, an entry is added to the Budget Review page.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that highlights if a customer's current budget amount is out-of-sync with their consumption patterns.

Budget Plan - True Up Algorithm

On Budget Plan - Main you must define the **True Up Algorithm** used by the Budget True Up background process when it periodically trues up a customer's budget.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that trues up budget amounts.

The system will automatically true up a customer's budget amount every X months (X is defined in **Months for True Up**).

Tender Management

When a payment is received, a tender is created to record what was remitted (e.g., cash, check, credit card). The topics in this section describe control tables that must be set up in order to remit tenders.

> **NOT_SUPPORTED**
>
> We strongly recommend Tender Management and Workstation Cashiering before setting up the control tables described in this section.

- Setting Up Tender Types
- Setting Up Tender Sources

Setting Up Tender Types

Tender types are used to indicate the method in which the tender was made. A unique **Tender Type** must exist for every type of tender that can be remitted. For example, if you allow cash, checks, direct debits from a checking account, and direct debits from a credit card to be tendered, you'd need the following tender types:

Copyright © 2026, Oracle and/or its affiliates.

Tender Type	Description	Like Cash	Generate Auto Pay	Require External Source ID	Require Expiration Date	External Type
CASH	Cash	Yes	No	N/A	N/A	N/A
CHEC	Check	No	No	N/A	N/A	N/A
OVUN	Cash drawer - over/under	No	No	N/A	N/A	N/A
DDCH	Direct debit - checking	No	Yes	Yes	No	Checking withdrawal
CRED	Direct debit - credit card	No	Yes	No	Yes	Credit card withdrawal

Go to **Admin**, and then **Financial**, and then **Tender Type** to define your tender types.

Description of Page

Enter a unique **Tender Type** and **Description** for the tender type.

Turn on the **Like Cash** switch if this tender type is cash or the equivalent of cash. This indicator controls if the system generates a warning if a cash-only account remits a tender other than cash. It is also used to generate a warning for online cashiers to turn in their tenders when the cash-like amount exceeds the maximum amount balance defined for the tender source.

Turn on **Generate Auto Pay** if this type of tender causes an automatic payment request to be routed to a financial institution. For example, this switch will be on if this tender type is used for direct debits from a customer's checking account (because every tender of this type will have an automatic payment request created when the tender is created).

The following fields are only used for tender types associated with automatic payments:

External Type This field is used by the background process that creates the information that is interfaced to the automatic payment source. Specifically, it controls the record type associated with the different types of automatic payments that are routed to the automated clearinghouse (ACH).

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is EXT_TYPE_FLG.

Require Ext. Src. ID This switch indicates if an Auto-Pay Source that references this type of tender must contain an External Source ID. The External Source ID is the unique identifier of the financial institution to which the automatic payment will be routed.

This switch is typically turned on for tender types associated with checking / saving direct debits. It is turned off for tender types associated with credit card debits (you don't need an external source for a credit card debit, you just need the credit card number).

Expiration Date Required Turn this switch on if an Auto-Pay Option that references an auto-pay source that references this type of tender must also contain an expiration date (e.g., automatic debit / credit cards).

Turn this switch off for tender types associated with checking / saving direct debits.

Copyright © 2026, Oracle and/or its affiliates.

Tender Authorization Indicates that tenders of a particular type require authorization prior to being created.

Business Object If Tender Authorization has a value of Required, a Business Object must be specified for the tender type. The primary function of this Business Object is to manage the authorization of payment tenders.

> **NOT_SUPPORTED**
>
> For more information on authorizing credit card payments, refer to the Tender Type - Credit Card with Authorization business object.

Turn on **Allow Cash Back** if the system should automatically calculate a cash back amount when a tender is remitted for this tender type and the amount tender exceeds the amount being paid.

Setting Up Tender Sources

A unique **Tender Source** must exist for every potential source of funds. For example,

- Every cashiering station will have a unique tender source.

> **Note**
>
> If your organization accepts alternate currency payments online, then a tender source must exist for each currency code accepted at the cashiering station.

- Every lock box will have a unique tender source.
- Your remittance processor will have a unique tender source.
- If you allow customers to pay bills automatically (e.g., via EFT), you'll need a tender source for each institution to which you route automatic payment requests. For example, if you route automatic payment requests to the automated clearinghouse (ACH), you'll need a tender source for the ACH.

For example, if you have 3 lock boxes, 2 cash drawers at an area office A, 2 cash drawers at area office B, and a single remittance processor, you'd need the following tender sources:

Tender Source	Type	External Source ID (Lockbox ID)	Default Starting Balance	Currency Code	Suspense Service Agreement
CASH-A01	Cashiering	N/A	150.00	USD	N/A
CASH-A02	Cashiering	N/A	150.00	USD	N/A
CASH-B01	Cashiering	N/A	150.00	USD	N/A
CASH-B02	Cashiering	N/A	150.00	USD	N/A
LB-INDUS	Lockbox	112910-A	N/A	USD	9291019281
LB-COMM	Lockbox	938219-C	N/A	USD	4739837372
LB-RESID	Lockbox	372829-B	N/A	USD	1912910192

Copyright © 2026, Oracle and/or its affiliates.

REMIT	Lockbox	N/A	N/A	USD	1920038437
ACH	Auto Pay	N/A	N/A	USD	N/A

To set up a tender source, select **Admin**, and then **Financial**, and then **Tender Source**, and then **Add.**

Description of Page

Enter an easily recognizable **Tender Source** and **Description** for the tender source.

Define the **Tender Source Type**. Valid values are: Ad Hoc, Auto Pay, Online Cashiering and Lockbox. The system uses this information to prevent tender controls from different sources from being included under the same deposit control. In other words, you can't mix ad hoc, automatic payment, cashiering and lockbox tenders under the same deposit control.

> **NOT_SUPPORTED**
>
> For more information, refer to Maintaining Deposit Controls.

If the source is an external system (e.g., a lockbox or an automatic payment destination), use **External Source ID** to define the unique identifier of the source. The background process that interfaces tenders from this source uses this information to create the appropriate tender control when it interfaces payments from external sources.

If this source is a cash drawer, define the **Default Starting Balance**. This balance is defaulted onto new tender controls and may be overridden.

> **Note**
>
> The tender type of the **Start Balance** is defined on the installation record.

If this source is a cash drawer, define the **Max Amount Balance**. When the amount of cash-like tenders in a cash drawer exceeds this balance, a warning is issued to remind the cashier to turn in some of the funds to a tender control.

Define the **Currency Code** of tenders linked to this source. All tenders in a source must be of the same currency.

If this tender source is associated with payments that are interfaced from an external source (e.g., a lockbox or a remittance processor), use Suspense **Service Agreement** to define the service agreement whose account will hold uploaded payments with an invalid account. Refer to Payment Upload Error Segmentation for more information about suspense service agreements. Also note, because the payment upload process simply books payments that reference invalid accounts to the account associated with this service agreement, this account should belong to a customer class with the appropriate payment distribution algorithms. This may entail creating a new customer class that will only be used on these "suspense accounts". This customer class would need the following algorithms:

- We'd recommend using a simple payment distribution algorithm like PYDIST-PPRTY (distribute payment based on SA type's payment priority and the age of the debt).

Copyright © 2026, Oracle and/or its affiliates.

- We'd recommend using an overpayment distribution algorithm like OVRPY-PPRTY (distribute overpayment to highest priority SA type).

Define the **Bank** and **Bank Account** into which the tender source's moneys will be deposited. The bank account defines the distribution code used to build the GL details for the payment. Refer to The Source of GL Accounts on Financial Transactions for more information. Note that the bank code and bank account can later be overwritten when entering Tender Deposits on Deposit Control.

Printer IP Address and **Printer Port** are for implementation use, if needed. The base samples for receipt printing do not use these settings, as they are designed for local printers. For more information on receipt printing, refer to:

- The Customer Cloud Service Receipt Printing section of the Cloud Services Implementation Guide. Or,
- The Receipt Printing section of the Optional Products Installation Guide, for on-premise implementations.

If this tender source is associated with payments that are interfaced from an external source, for example tender sources associated with Auto Pay and Lockbox **Tender Source Types**, the information is also used as follows:

- The payment upload process uses this information to populate the bank and bank account when it creates deposit control records for the tender controls it creates during the interface. Refer to Managing Payments Interfaced From External Sources for more information.
- The automatic payment interface uses this information to populate the bank and bank account when it creates deposit control records for the tender controls it creates during the interface.

> **Note**
>
> If your organization accepts alternate currency payments online, then a tender source must exist for each currency code accepted at the cashiering station. When a user adds a tender control the system attempts to default a tender source based on the currency of the deposit control and the tender source(s) define on the user's record.

> **NOT_SUPPORTED**
>
> Refer to Alternate Currency Payments for more information.

Automatic Payment Options

If your customers can pay their bills automatically (via direct debit or credit card debits), you'll need to set up the various control tables described in this section.

Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> Besides the tables described in this section, additional values must also be added to control tables defined under Tender Management. Specifically, refer to Setting Up Tender Types and Setting Up Tender Sources.

> **NOT_SUPPORTED**
>
> Refer to Automatic Payments for more information about how automatic payments are handled in the system.

- Setting Up Auto Pay Route Types
- Setting Up Auto Pay Source Codes
- SEPA Payments
- Web Debit Account Validation

Setting Up Auto Pay Route Types

This portal is used to display and maintain an Auto Pay Route Type.

Auto Pay Route Types are used to control when and how automatic payment requests are routed to a financial institution, and when the general ledger is impacted.

You can access the portal from **Admin**, and then **Financial**, and then **Auto Pay Route Type**.

The information captured on the auto pay route type depends on the business objects supported by your implementation. Refer to the embedded help for more information.

Setting Up Auto Pay Source Codes

A unique **Auto Pay Source** must exist for every bank / credit card company / bill payment service that your customer's use as the source of the funds when they sign up for automatic payment. For example,

- Every bank will have a unique auto pay source.
- Every credit card company will have a unique auto pay source.

To set up an auto pay source, select **Admin**, and then **Financial**, and then **Auto Pay Source Type**, and then **Add.**

Description of Page

Enter an easily recognizable **Auto Pay Source Code** and **Description** for the auto pay source.

The **Source Name** is the name of the financial institution.

When the system creates an automatic payment request, it also creates an associated payment tender. This tender (like all tenders) must have a tender type. This field defines the **Tender Type** associated with this auto pay source's tenders. The **Prenote Tender Type** is also

Copyright © 2026, Oracle and/or its affiliates.

specified on checking and savings tender types. It determines the transaction code to stamp on an auto pay extract records that are sent out for the purpose of debit prenotification. Refer to Setting Up Tender Types for more information.

The **External Source ID** is the unique identifier of the financial institution to which the automatic payment will be routed (e.g., the bank routing ID of the bank). This field is typically blank on automatic payments routed to credit card companies because the credit card company doesn't have an external source ID (whereas direct debits from banks must have a bank routing number). Whether this field is required is controlled by the **Tender Type**.

The **Auto Pay Route Type** controls when and how automatic payment requests get routed to a financial institution. It also controls when the general ledger is impacted by the automatic payments financial transaction. Refer to Setting Up Auto Pay Route Types for more information.

The **Work Calendar** defines the financial institution's workdays. This information is used to determine the date on which automatic payment requests will be sent to the financial institution. Refer to Setting Up External Workday Calendars for more information.

The **Validation Algorithm** defines how the system validates the customer's account ID at the financial institution. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that validates the customer's account ID at the financial institution.

Refer to Account - Auto Pay for more information.

SEPA Payments

The topics in this section provide background information about the Single Euro Payments Area (SEPA) payment functionality.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization is participating in SEPA payment transactions such as direct debit collection.

- What Is SEPA?
- SEPA Direct Debit
- SEPA Direct Debit Mandate
- Configuring the System for SEPA Direct Debits

What Is SEPA?

SEPA (Single Euro Payments Area) is a European Union (EU) integration initiative that is aimed at streamlining processes that are related to cross-border payments. SEPA consists of the EU member states plus a few additional countries. In SEPA, customers can make electronic Euro payments within and across these countries under the same rights and obligations. SEPA payment services are based on global ISO (International Organization for Standardization) standards.

Copyright © 2026, Oracle and/or its affiliates.

SEPA Direct Debit

The SEPA Direct Debit (SDD) scheme allows a creditor to collect money from the debtor, with prior approval from the debtor. The direct participants are the following:

- Creditor
- Creditor's bank
- Debtor
- Debtor's bank

The debtor and creditor must each hold an account with a payment service provider located within SEPA. The accounts may be in Euro or in any other SEPA currency. However, the transfer of funds between the debtor's bank and the creditor's bank always takes place in the Euro currency. The indirect participants are the following:

- Clearing and Settlement Mechanisms (CSMs) such as an automated clearinghouse
- Intermediary Banks that offer intermediary services to debtor banks and/or creditor banks

SEPA Direct Debit Mandate

The mandate is the consent and authorization that the debtor gives to the creditor, to allow the creditor to initiate direct debit collections. The creditor is responsible for storing the original mandates, together with any amendments relating to the mandate or information regarding its cancellation or lapse. The contents of this section describe how mandates are issued and canceled.

- Issuing a Mandate
- Canceling a Mandate
- The Lifecycle of a Direct Debit Mandate Task

Issuing a Mandate

The creditor initiates the issuance of a mandate by sending the mandate form (either paper or electronic) to the debtor with the creditor information filled in. The creditor information includes the creditor's unique identifier as a SEPA creditor. An ISO standard specifies the structure of the creditor identification, which includes country code, a check digit, the creditor's business code and a country-specific identifier for the creditor. The creditor ensures that the mandate form contains the mandatory legal wording and the mandatory set of information, as specified in the standards. The debtor must ensure that the required information is supplied and that the mandate is signed, either in writing or electronically. The mandatory debtor information includes the debtor's international bank account number (IBAN) and bank identifier code (BIC). The creditor assigns a unique reference for the mandate and provides that reference to the debtor before the first initiation of a collection. The debtor can then use both the unique mandate reference and the creditor identification to verify the bank transactions.

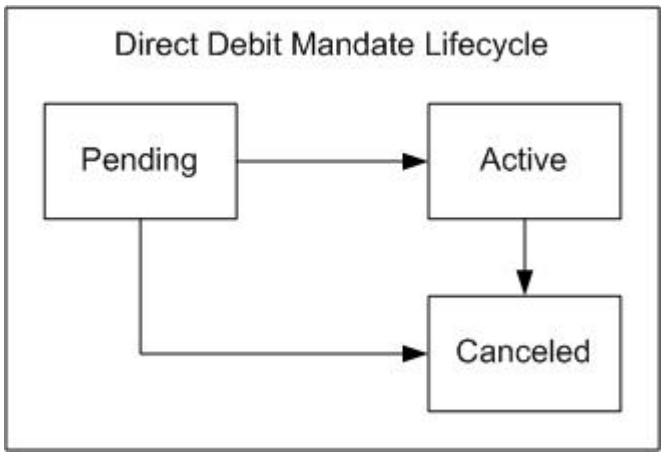
Canceling a Mandate

A mandate can be canceled by either the debtor or the creditor, without the involvement of either bank.

Copyright © 2026, Oracle and/or its affiliates.

The Lifecycle of a Direct Debit Mandate Task

The following diagram shows the possible lifecycle of the Direct Debit Mandate Task business object:



Pending State: The direct debit mandate starts out in a pending state.

Active State: The direct debit mandate is transitioned manually by a user to the active state. Once a mandate is created and activated in the system, the account's auto pay information must be updated to reflect information on the mandate. This process may be automated within the mandate's lifecycle and a sample algorithm exists in the demo database to achieve this. This algorithm is also included in the export bundle in the demo database.

Canceled State: The direct debit mandate is transitioned manually by a user to the canceled state.

Configuring the System for SEPA Direct Debits

The following sections describe the setup required if your organization intends to process SEPA direct debit transactions.

- Define Auto-Pay Route Type
- Define a Direct Debit Task Type
- Define SEPA Country Codes
- Define Mandate Cancel Reasons
- Define Navigation Options
- Setting Up Account Auto Pay

Define Auto-Pay Route Type

Configure an auto-pay route type for routing direct debit requests to a financial institution. This auto-pay route type must reference the SEPA Direct Debit Payment Extract (C1-SDDCE) batch process.

Refer to Setting Up Auto-Pay Route Types for more information.

- Define Auto-Pay Source Type

Copyright © 2026, Oracle and/or its affiliates.

Define Auto-Pay Source Type

Ensure that your auto-pay source codes reference the above auto-pay route type. You must also configure a validation algorithm on the auto-pay source type to ensure the International Bank Account Number (IBAN) defined on the Auto-pay Source of the Account has the correct format.

Refer to Setting Up Auto Pay Source Codes for more information.

Define a Direct Debit Task Type

Your implementation must configure a **Direct Debit Task Type.** This service task type captures general information that the SEPA extract process needs to create the output file. The base product provides a business object for direct debit task type, **C1-DirectDebitMandateTaskType.**

Note that the direct debit task type is maintained on the Service Task Type portal.

Define SEPA Country Codes

The list of countries and territories that are part of the Single Euro Payment Area (SEPA) must be defined on extendable lookup **C1-SEPACountryCodeLookup**. These country codes are used by the system to validate the creditor identification entered on a direct debit task type.

Note that an export bundle exists in the demo database that includes a sample list of SEPA country codes. This list must be verified by your implementation for accuracy.

Define Mandate Cancel Reasons

A cancel reason is required when canceling a direct debit mandate. Ensure that these status reasons have been configured for the direct debit task business object, **C1-DirectDebitMandateTask.**

Define Navigation Options

The base product includes navigation options for easy access to query and maintain an account's direct debit mandate. To access these portals, add a new menu item to the account context menu (**CI_CONTEXTACCOUNT**). This menu item should include two menu lines:

- A context menu line for navigation to the direct debit mandate that references a navigation option of **c1DirectDebitMandate.** The system uses the account in context and attempts to find the most recent active mandate for the account. If a mandate exists, navigation is to the service task maintenance portal. If an active mandate does not exist, navigation is to the service task query portal.
- A context menu line for adding a new direct debit mandate that references a navigation option of **c1AddDirectDebitMandate.**

Setting Up Account Auto Pay

The SEPA direct debit extract process retrieves banking details from the account's auto pay record. Once a mandate is created and activated in the system, the account's auto pay information must be updated to reflect information on the mandate. This process may be

Copyright © 2026, Oracle and/or its affiliates.

automated within the mandate's lifecycle and a sample algorithm exists in the demo database to achieve this. This algorithm is also included in the export bundle in demo.

Web Debit Account Validation

Overview

This section describes Web Debit Account Validation and the steps to enable this functionality.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization is required to validate bank accounts that are used for the first time in electronic payments and/or automatic payment enrollments. This functionality does not cover credit card payment types, which are verified through tender authorization.

Web Debit Account Validation allows your implementation to validate a first-use checking/savings account that is captured through any of the following:

- Agent-assisted One-Time Payments (via Add One Time Payment user interface)
- Agent-assisted Account Auto Pay Setup (Account - Auto Pay page)
- Self-Service Automatic Payment Setup
- Self-Service One Time Payment
- Self-Service Wallet Management
- Other user interface or back-end processes that add/update Account Auto Pay records using the Account Auto Pay row maintenance.

The account validation method is your implementation's choice, based on what regulation prescribes as acceptable account validation options. For instance, the National Automated Clearinghouse Association (NACHA) - that governs the ACH network in the United States - prescribes any of the following options:

- **ACH prenotification.** With this option, a zero-amount checking/savings debit pre-note transaction is sent through the auto pay extract, to verify the bank account information. If a negative response is not received from the bank within a set number of days, the account is considered valid.
- **ACH micro-transaction verification.** With this option, two very small transactions – a debit to deposit the money and a credit to remove the money - are sent to the account. The account holder needs to confirm the amounts on the transactions. The account will only be considered verified if the account holder describes the amounts correctly.
- **Commercially available third party validation service vendors.** Some financial institutions and third parties have access to a large number of accounts and may offer a product that leverages this information. These services can be accessed for verifying that account information is correct and/or verifying the account holder's identity. The responses can be real-time, same-day or next-day.

Web Debit Account Validation is a Plug-in

An algorithm plugged in on Auto Pay Route Type or on CIS Division - Auto Pay Route Type Override Controls (if applicable) is responsible for web debit account validation. Payment

Copyright © 2026, Oracle and/or its affiliates.

methods captured for automatic payment setup and one time payments specify the Auto Pay Source, which in turn, identifies the Auto Pay Route Type.

The base product provides a Web Debit Account Validation Algorithm for ACH prenotification. The algorithm creates a zero-amount payment event with a Prenote Tender Type (from the Auto Pay Source) and an Auto Pay Clearing Staging record containing the account number to verify and that is scheduled to extract the same day.

Refer to the Web Debit Account ACH Prenotification (C1WDAPRENTF) algorithm for details.

> **Note**
>
> The auto pay extracts (APAYACH and C1-APACH) stamp specific transaction codes for checking debit prenotification (28) and savings debit prenotification (38) on the Entry Detail record. The transaction code is retrieved from the prenotification Tender Type's External Type.

Web Debit Account Validation Applies to Self-Service Tasks that Manage Payment Options

The following self-service task business objects include an Account Validation status, where web debit account validation is performed:

- Self-Service Automatic Payment Setup Task - WX-AutoPayTask
- Self-Service One Time Payment Task - WX-OneTimePayTask
- Self-Service Payment Option Task - C1-PaymentOptionsTask

The service task's Account Validation status has an enter algorithm that performs the Web Debit Account Validation Algorithm configured in the Auto Pay Source's Auto Pay Route Type (or in the CIS Division's Override Auto Pay Controls, if applicable) when the service task type has Web Debit Account Validation enabled and the specified bank account number is not found in any of the account's existing auto pay options and in any of the account persons' self-service payment options. The account auto pay option or self-service payment option is not saved until the bank account is determined to be valid. If web debit account validation processing encounters an error, the service task goes to the Account Validation Error status that includes retry processing and wait timeout processing. If the bank account is deemed invalid, the service task goes to the Rejected status. Refer to the above mentioned business objects in the application, for more details.

> **Note**
>
> When checking for the existence of a bank account number in person self-service payment options, the system relies on the hash value (in HASH_EXT_ACCT_ID). Implementations must ensure that encryption configuration for this table field includes both the encrypted and hash fields and that legacy data has both fields populated. Refer to Encrypting Sensitive Data and Managing Encrypted Data for more details.

Validating Bank Accounts Captured on Add One-Time Payment UI

The Add One-Time Payment user interface allows the capture of first-time-use bank accounts. If a bank account has not been used on any of the existing Account and prior self-service one time payments, the system will create a One-Time Payment Task (C1-OneTimePaymentTask) that performs web debit account validation and adds the payment event (if the bank account is valid). Since the Add One-Time Payment UI can handle multiple tenders, a single One-Time

Copyright © 2026, Oracle and/or its affiliates.

Payment Task manages all auto pay tenders that require validation. The service task's Account Validation status has an enter algorithm that performs the Web Debit Account Validation Algorithm configured in the Auto Pay Source’s Auto Pay Route Type (or in the CIS Division’s Override Auto Pay Controls, if applicable).

Note that the Payment Event is kept in an Incomplete state while web debit account validation is in progress. The service task's lifecycle processing on the service task will either freeze or delete the Payment Event, depending on the result of bank account validation. If the payment is frozen, a Person Self-Service Payment Option is also created for the main person on the account, to make that payment option available for future self-service one-time payments.

Refer to the C1-OneTimePaymentTask business object in the application, for more details.

Validating Bank Accounts on Account Auto Pay Setup

The Account - Auto Pay page checks if a specified bank account number has not been used on prior account auto pay options or on person self-service payment options. If this is the case, the system creates a Bank Account Auto Pay Setup Task that performs the web debit account validation. The service task's Account Validation status has an enter algorithm that performs the Web Debit Account Validation Algorithm configured in the Auto Pay Source’s Auto Pay Route Type (or in the CIS Division’s Override Auto Pay Controls, if applicable).

Note that the Account Auto Pay record is not added/updated until the bank account is deemed valid. The auto pay details are kept in the service task while the validation is ongoing. If the customer happens to change the bank account number while validation is in progress, the service task will be updated with the new bank account number and will transition back to the Account Validation status, to validate the new bank account number.

If the bank account is deemed valid, the Account Auto Pay is added/updated. A Person Self-Service Payment Option is also created for the main person on the account, to make that payment option available for future self-service one-time payments.

If web debit account validation processing encounters an error, the service task goes to the Account Validation Error status that includes retry processing and wait timeout processing. If the bank account is deemed invalid, the service task goes to the Rejected status.

Refer to the C1-BankAcctApaySetupTask business object in the application, for more details.

An informational message will appear above the Account Auto Pay scroll if the account has one or more pending auto pay enrollments. The Alerts sidebar zone will also highlight the pending auto pay enrollment(s) if the Highlight Pending Auto Pay Enrollment (C1PNAPAYALRT) Control Central alert algorithm is enabled. Both the informational message on Account Auto Pay and the sidebar alert navigate to the service task. If there are multiple open service tasks, the navigation will be to the most recently updated service task (based on status update date/time).

> **Note**
>
> Since the Bank Account Auto Pay Setup Task is created by the Account Auto Pay row maintenance program, any other user interface (e.g., process flow) or back-end processing that calls the row maintenance program to add//update Account Auto Pays will perform this bank account validation, if needed.

Configuring the System for Web Debit Account Validation

The following are the steps to enable Web Debit Account Validation functionality:

Copyright © 2026, Oracle and/or its affiliates.

- Determine the account validation option(s) that your implementation will use.
  - If opting for ACH prenotification, you can use the Web Debit Account ACH Prenotification (C1WDAPRENTF) algorithm provided in the base product. To use this algorithm:
    * Configure tender types for checking debit prenotification and savings debit prenotification. Use the applicable External Type value on each - i.e. Checking Debit Prenotification (28) or Savings Debit Prenotification (38), The auto pay extract process uses the Prenote Tender Type's External Type as the transaction code on the Entry Detail record in the extract file.
    * Configure your checking and savings Auto Pay Sources with the applicable Prenote Tender Types.
  - If opting for ACH micro-transaction verification or third partly validation, create a Web Debit Account Validation algorithm for your chosen method.
- Plug in the Web Debit Account Validation on your applicable Auto Pay Route Types and CIS Division - Auto Pay Route Type Override Controls (if applicable).
- To enable bank account validation on Account Auto Pay setup:
  - Add a Service Task Type using the Bank Account Auto Pay Setup Task Type (C1-BankAcctApaySetupTaskType) business object and configure the settings accordingly.
  - On your Financial Transaction Options feature configuration, add the Bank Account Auto Pay Setup Service Task Type option and specify the service task type that you created in the previous step.
  - On the Bank Account Auto Pay Setup Task BO:
    * If using the base-supplied Web Debit Account ACH Prenotification algorithm (or a CM algorithm that allows for a wait period after which the account is considered valid), monitor logic on the Account Validation status should check if the wait time has elapsed - and if so, transition the service task accordingly. Note that the Check Account Validation Status monitor algorithms provided in the base product simply transition the service task to the next state (i.e. Process Auto Pay) when the wait time elapses. The base algorithm does not check for a negative response because the form of response is implementation-specific. If your implementation needs to check for a negative response, copy the base algorithm and add that processing. Refer to the Check Account Validation Status for Auto Pay Setup (C1APAYACVAST) algorithm for details.
    * To use the To Do creation algorithm plugged in on the Account Validation Error status, configure a new algorithm parameter instance to specify the Retry Frequency and Discard Cancel Terminate Reason (if applicable).
    * To use the retry processing on the Account Validation Error status, configure a new algorithm parameter instance to specify the Maximum Retry.
  - Configure the Highlight Pending Auto Pay Enrollment (C1PNAPAYALRT) Control Central alert algorithm on Installation Options.
- If your implementation will use the Add One-Time Payment UI (i.e., instead of the Payment Event - Add pop-up), configure the One-Time Payment Task feature option. Refer to One Time Payment Options for more details.
- If your implementation will use self-service automatic payment and self-service one time payment business objects:
  - On your Self-Service Automatic Payment Setup and One Time Payment Self-Service Task Types, set the Web Debit Account Validation flag to Yes. To use the wait timeout

Copyright © 2026, Oracle and/or its affiliates.

processing built into the Account Validation Error state, configure the Timeout Processing settings. Refer to the service task's embedded help for more details.

- On the Self-Service Automatic Payment Setup and One Time Payment service task business objects:
  - If using the base-supplied Web Debit Account ACH Prenotification algorithm (or a CM algorithm that allows for a wait period after which the account is considered valid), monitor logic on the Account Validation status should check if the wait time has elapsed - and if so, transition the service task accordingly. Note that the Check Account Validation Status monitor algorithms provided in the base product simply transition the service tasks to the next state (i.e. Process Auto Pay, Process Payment or Process Request) when the wait time elapses. The base algorithms do not check for a negative response because the form of response is implementation-specific. If your implementation needs to check for a negative response, copy the base algorithms and add that processing. Refer to the Check Account Validation Status for One Time Payment (C1OTPACVAST) and Check Account Validation Status for Auto Pay Setup (C1APAYACVAST) algorithms for details.
  - To use the To Do creation algorithm plugged in on the Account Validation Error status, configure a new algorithm parameter instance to specify the Retry Frequency and Discard Cancel Terminate Reason (if applicable).
  - To use the retry processing on the Account Validation Error status, configure a new algorithm parameter instance to specify the Maximum Retry.

One-Time Payment Options

The Add One-Time Payment popup provides the ability to 'quick add' a payment event using existing auto pay methods, new checking/savings auto pay methods, cash and non-cash/non-auto pay methods (like check).

This user interface provides the same capabilities as the Payment Event - Add Dialog, including processing for open-item and non-CIS payor accounts. In addition, it provides the following:

- Ability to specify auto pay tender details within the same popup. No need to navigate to the Payment Event page.
- Ability to select an existing auto pay method from either a currently enrolled Account Auto Pay option or a stored Person Self-Service Payment Option.
- Ability to specify new auto pay methods, with an option to save the payment method for future one-time payments
- Web debit account validation can be enforced for first-use checking and savings accounts. Refer to Web Debit Account Validation for more details.

Enabling the Add One-Time Payment User Interface

To enable the Add One-Time Payment popup, configure the following options in Financial Transaction Options feature configuration:

- **Enable Add One-Time Payment UI**. Set this to Y, to replace the Payment Event-Add pop-up with the Add One-Time Payment pop-up on all existing launch points of the old pop-up.
- **One-Time Payment UI Navigation**. Specify where the user should be navigated after successful one-time payment creation. This is optional. If not specified, the pop-up simply closes.

Copyright © 2026, Oracle and/or its affiliates.

- **One-Time Payment Service Task Type**. If your implementation enabled Web Debit Account Validation, add a One-Time Payment service task type using the base-provided C1-OneTimePaymentTaskType and C1-OneTimePaymentTask. Specify your One-Time Payment service task type on this feature option.
- **Override Script for Add One-Time Pmt UI**. If your implementation has additional requirements for the Add One-Time Payment UI, you can clone the base-owned BPA script (C1OTPayAddNv) and specify your CM script on this feature option. If configured, the system will launch this override script instead of the base-owned script.

> **NOT_SUPPORTED**
>
> Refer to Add One-Time Payment for more information.

Third Party Payment Processing Options

If you allow customers to make one time payments and/or pay bills automatically through a third party payment processing system, you will need to set up a number of control tables to enable these functions in the system.

> **Important**
>
> Aside from the tables mentioned in these sections, control tables defined under Automatic Payment Options must also be configured. Additional values must also be added to control tables defined under Tender Management. Specifically, refer to Setting Up Tender Types and Setting Up Tender Sources.

> **NOT_SUPPORTED**
>
> Refer to Third Party Payment Processing for the big picture of third party payment processing.

- Define Service Task Types
- Define Auto Pay Sources and Auto Pay Route Types
- Define Tender Cancel Reason
- Third Party Processing DateTime
- Third Party Payment Processing Master Configuration

Define Service Task Types

The web services used to process one time payment notifications, automatic payment setup and account verification create a corresponding service task record to perform the needed actions.

The following service task types need to be configured and defined on the Third Party Payment Processing Integration master configuration:

Copyright © 2026, Oracle and/or its affiliates.

- A **Third Party One Time Payment** service task type is used to create an immediate or scheduled third party one time payment. The base product supplies the following service task type and service task business objects: **C1-ThirdPartyOneTimePyTaskType** and **C1-ThirdPartyOneTimePyTask**. Refer to these business objects for details on schemas and processing logic.
- A **Third Party Account Verification** service task type is used to verify an account, which is typically done prior to capturing and processing a one time payment within the third party payment processing system. The base product supplies the following service task type and service task business objects: **C1-ThirdPartyAcctVerifTaskType** and **C1-ThirdPartyAcctVerifTask**. Refer to these business objects for details on schemas and processing logic.
- A **Third Party Auto Pay Setup** service task type is used to add or update an account's ongoing automatic payment option. The base product supplies the following service task type and service task business objects: **C1-ThirdPartyAutoPayTaskType** and **C1-ThirdPartyAutoPayTask**. Refer to these business objects for details on schemas and processing logic.

Refer to Defining Service Task Types for details on setting up service task types

Define Auto Pay Sources and Auto Pay Route Types

Configure specific auto pay sources for the payment types that your third party payment processing system accepts for one time payment processing and ongoing automatic payment processing. Base-supplied Payment Type values are: Checking Withdrawal, Savings Withdrawal and Credit Card Withdrawal. Your implementation can add payment types. In the Third Party Payment Processing Integration master configuration, each accepted payment type is associated with a Tender Type that is used to create the payment tender associated with a third party one time payment. For checking and savings payment types, the tender type and routing number are used to identify the auto pay source.

Auto pay sources used for ongoing automatic payment processing (i.e. ACH) should be associated with an auto pay route type that specifies C1–TAPEX as the Extract Batch. This ensures that the correct extract batch code is stamped on bill-related auto pay staging records during auto pay activation.

Define Tender Cancel Reason

Configure a Tender Cancel Reason that will indicate that a payment tender was canceled from the third party payment processing system. This cancel reason is configured in the Third Party Payment Processing Integration master configuration and used when a scheduled or processed one time payment is canceled.

Third Party Processing DateTime

The services used in third party payment processing allow for the possibility that the third payment processing system is processing in a different time zone as the utility's. The Third Party Payment Processing Integration master configuration includes the third party's Processing Cutoff Time and Processing Cutoff Time Zone, as well as the local equivalent of the processing cutoff time, based on the Time Zone defined on the Installation record.

The payment datetime that is sent by the third party payment processing system with one time payment notifications and auto pay confirmations is stored as-is as a characteristic on the one time payment's tender. This is done for the purposes of tender controls processing and

Copyright © 2026, Oracle and/or its affiliates.

payments reconciliation (*i.e.*, with reports generated from the third party payment processing system.

As a standard, inbound and outbound communications external systems must use the standard xsd datetime format (yyyy-MM-dd'T'hh:mm:ss[+ or -]hh:mm). The payment datetimes sent with third party one time payment notifications and auto pay confirmations should follow this format.

The Third Party One Time Payment Task also stores the local equivalent of the payment datetime, based on the Time Zone defined on the Installation record. This local datetime is used anywhere third party one time payments (and their payment dates) are displayed in the system (*e.g.*, in the Scheduled Payments Log in Control Central - Account Information.

Third Party Payment Processing Master Configuration

The integration with a third party payment processing system requires the setup of master configuration data that controls the processing of one time payments and ongoing automatic payments.

To set up the third party payment processing master configuration, navigate using **Admin**, and then **General**, and then **Master Configuration**. You are brought to an all-in-one portal with options to add/edit the Third Party Payment Processing Integration master configuration record.

Refer to the master configuration's embedded help for more details on how each section in this record is configured.

Payment Advices

The topics in this section provide background information about payment advice functionality.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization issues payment advices to the customer instead of initiating electronic funds transfer directly to the customer's bank.

- What Is A Payment Advice?
- Payment Advice Versus Direct Debit
- Setting Up The System To Enable Payment Advices

What Is A Payment Advice?

Payment advice is a money order that is established at the initiative of the utility. When a bill is completed, the utility sends the customer a document that indicates a payment amount and the customer's bank details. If the customer agrees to the information on the payment advice, they sign it and return it to the clearinghouse address that is indicated on the payment advice. The

Copyright © 2026, Oracle and/or its affiliates.

clearinghouse, in turn, sends the dated and signed payment advice to the customer's bank, which completes the payment.

Payment Advice Versus Direct Debit

The existing functionality that creates automatic payments is referred to as direct debit processing. Payment advice processing differs from direct debit processing in the way that automatic payments get initiated. With payment advice processing, the usual automatic payment records - i.e. payment event, payment, tender and auto pay clearinghouse staging - are not created. Instead, a payment advice is printed and sent to the customer. The customer sends the approved payment advice directly to the clearinghouse.

> **Note**
>
> The system does not provide sample processes for extracting and printing payment advice information. Your implementation team would have to create these.

Setting Up The System To Enable Payment Advices

You must set up a Financial Transaction Options Feature Configuration to define parameters that control payment advice functionality.

The following points describe the various **Option Types** that must be defined:

- Payment Advice Functionality Supported. This option controls whether the system allows for payment advice processing.
  - Enter Y if the system should allow for both direct debit and payment advice processing.
  - Enter N if the system should only allow for direct debit processing.
- Default Auto Pay Method. This option is used for defaulting the auto pay method on new account auto pay records.

Refer to Account - Auto Pay for more information on auto pay method.

> **Note**
>
> The system assumes direct debit processing if the above feature options are not defined.

Credit Card Payments

If your organization accepts credit card payments, you can configure the system to authorize customers' credit card charges in real-time, and perform an authorization reversal (also in real-time) when the credit card payment is canceled. When the authorization web service is not

Copyright © 2026, Oracle and/or its affiliates.

available, you can permit users to enter authorization codes manually so that they can continue processing payments.

- Configuring the System for Tender Authorization

Configuring the System for Tender Authorization

The following sections describe the setup required if your organization intends to use the base CyberSource integration tender authorization functionality.

- Define the Outbound Message Type
- Define the Message Sender
- Define the External System and Configure the Messages
- Define a User
- Set up the Tender Authorization Algorithm
- Define a Business Object
- Define Tender Types
- Tender Authorization - Feature Configuration

Define the Outbound Message Type

An outbound message type is required for the CyberSource authorization outbound message. This outbound message type must reference the base CyberSource - Credit Card Authorization business object.

An outbound message type is required for the CyberSource reversal outbound message. This outbound message type must reference the base CyberSource - Credit Card Reversal business object.

Define the Message Sender

A Message Sender is required to define how to send messages to CyberSource. Use the context of the Message Sender to define the web service interface.

Define the External System and Configure the Messages

Define an external system and configure the valid outbound message types and the method of communication for each (XAI,Batch or Real Time; Real Time is generally the appropriate choice for credit card authorization). You will also need to select the appropriate XSLs to format both the request and response to the outbound message types for CyberSource.

Define a User

Copyright © 2026, Oracle and/or its affiliates.

Add a user to hold details required for CyberSource communication. Security information (e.g. Merchant Id, Merchant Reference Code, CyberSource User Name and Password) needed to interface with CyberSource is stored as user characteristics.

Set up the Tender Authorization Algorithm

A Tender Type (BO) - Tender Authorization algorithm must be configured. This algorithm performs a tender authorization or a tender authorization reversal through CyberSource.

Define a Business Object

A business object (BO) must be created for the TENDER TYPE maintenance object. This BO must reference the tender authorization algorithm created.

Define Tender Types

Update the appropriate tender type(s) to denote that authorization is required. The new business object must be specified on the tender type(s).

Tender Authorization - Feature Configuration

If your implementation has a need to prevent users from overriding the automatic tender authorization, then you must set up the Allow Manual Tender Authorization Override option type on the Financial Transaction Options Feature Configuration. The Allow Manual Tender Authorization Override option must have a value of N in order to suppress the Authorization Override checkbox.

NOT_SUPPORTED

For more information on credit card payment authorization refer to the Tender Type - Credit Card with Authorization business object.

Non CIS Payments

Payment Templates can be configured for common types of non CIS payment allocations. These templates are used to default the payment distribution and allow non CIS payments to be directly allocated to specific distribution codes.

- Setting Up Payment Templates

Setting Up Payment Templates

Copyright © 2026, Oracle and/or its affiliates.

Payment templates contain the rules that control how non CIS payments are created. You can use a payment template to default the payment distribution for common types of non CIS payments. To set up a payment template, open **Admin**, and then **Financial**, and then **Payment Template**, and then **Add.** The topics in this section describe the base-package zones that appear on the Payment Template portal.

- Payment Template List
- Payment Template

Payment Template List

The Payment Template List zone lists every payment template. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent payment template.
- Click the Add link in the zone's title bar to add a new payment template.

Payment Template

The Payment Template zone contains display-only information about a payment template. This zone appears when a payment template has been broadcast from the Payment Template List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the payment template.
- Click the **Delete** button to start a business process that deletes the payment template.
- Click the **Duplicate** button to start a business process that duplicates the payment template.
- Click the **Activate** or **Deactivate** button to start a business process that updates the status of the payment template.

Please see the zone's help text for information about this zone's fields.

Alternate Currency Payments

The topics in this section provide background information about alternate currency payments.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization accepts payments tendered in a currency other than the customer's currency. If your organization does not accept alternate currency payments, you need only indicate such on the Installation Record; no other setup is required.

- What Is An Alternate Currency Payment?

Copyright © 2026, Oracle and/or its affiliates.

- Configuring the System for Alternate Currency Payments

What Is An Alternate Currency Payment?

The currency code on the customer's account defines the currency in which the account's financial transactions are expressed. If the customer remits a payment in a different currency, this is referred to as an alternate currency payment. The system enables conversion of the tendered amount to the account's currency and captures the alternate currency and amount, as well as the exchange rate used in the conversion on the payment tender. The payment tender is linked to a tender control that references the alternate currency.

Configuring the System for Alternate Currency Payments

- Allowing Alternate Currency Payments
- Payment Event Business Object
- Define the User's Tender Sources

Allowing Alternate Currency Payments

You must set the **Alternate Currency** flag on the installation Record to Allowed if your organization accepts alternate currency payments. This option controls whether the **Currency Converter** button is displayed when a payment is processed on the payment portal.

Payment Event Business Object

A business object (BO) must be created for the PAY EVENT maintenance object. You must specify this BO as the option value for the CIS Payment Event Add BO option type on the Financial Transaction Options Feature Configuration. This BO must have the Currency Conversion Script BO option defined. This BPA script is invoked when the user clicks on the **Currency Converter** button during CIS payment processing on the payment portal.

> **Note**
>
> **Currency conversion logic is customizable.** The base product includes a script for currency conversion called C1-ConvCurr that's plugged in on the C1-CISPaymentEvent business object. This script converts an alternate currency amount to the account's currency using a bill factor value. The bill factor to use is derived by concatenating the alternate currency code and the account's currency code. For example, if converting US Dollars (USD) to Barbados Dollars (BBD) the bill factor code to use would be USDBBD. Your implementation can change this logic by developing a new script and plugging it into the payment event business object.

Copyright © 2026, Oracle and/or its affiliates.

Define the User's Tender Sources

Define the tender source(s) for the location (e.g., the specific cash drawer) in which a user's payment tenders are stored during the day. A tender source should be specified for each currency that payments are accepted in. The tender source(s) on the user record are used by the system when a user adds a new tender control. The system attempts to default a tender source on a new tender control based on the deposit control's currency and the tender source(s) defined on the user's record.

Payment Event Distribution

The base-package, by default, creates a single payment for a payment event. If your business requires potentially many payments to be created when payment events are added, you'll need to set up the various control tables described in this section.

> **NOT_SUPPORTED**
>
> Refer to Distributing A Payment Event for more information about how payment event distribution is handled in the system.

- Making Payments Using Distribution Rules
- Setting Up The System To Use Distribution Rules

Making Payments Using Distribution Rules

As part of this method, one or more distribution details are provided at payment time along with the usual payment and tender information. Each distribution detail record references a distribution rule and a corresponding value. The distribution rule encapsulates the business rules that govern the distribution of the payment amount into payments using the specified value.

The type of value being captured on the distribution detail and the logic that uses it to create payments are defined on the distribution rule.

- Rule Value
- Determine Tender Account
- Creating Payment(s)
- Rule Value Can Capture Additional Information

Rule Value

Copyright © 2026, Oracle and/or its affiliates.

The primary use of the rule value is to identify the business entity whose balance is to be relieved by creating payment(s). In most cases where the payor account is the same as the payee account it may also used to identify the tender account associated with the payment(s).

Determine Tender Account

The very first step in processing a distribution detail is to identify the tender account (i.e. the payor) associated with the payment. To do that the system calls the Determine Tender Account algorithm defined on the distribution rule providing it with the rule value and other tender information.

Creating Payment(s)

The business logic that distributes a payment amount into one or more payments(s) targeted towards the entity identified by a rule value is held in designated Create Payment algorithms defined on the distribution rule.

Rule Value Can Capture Additional Information

A rule value can also be used to capture additional information provided at payment time, like address information, comments, etc. Obviously payment distribution details with this type of rule value should have a zero payment amount, as they are not real payments. These distribution details end up being linked to a payment event, but unlike other distribution details they do not contribute any payments. You can think of these details as payment event characteristics.

You don't have to set up a Create Payment algorithm for distribution rules intended solely to capture additional payment information.

Setting Up The System To Use Distribution Rules

- Setting Up Distribution Rules
- Feature Configuration

Setting Up Distribution Rules

Define a Distribution Rule for each payment event distribution method practiced by your business.

- Distribution Rule - Main
- Distribution Rule - Algorithms

Distribution Rule - Main

Copyright © 2026, Oracle and/or its affiliates.

To set up a distribution rule, navigate to **Admin**, and then **Financial**, and then **Distribution Rule**, and then **Add.**

Description of Page

Enter a unique **Distribution Rule** and **Description** for the distribution rule.

Provide a short and unique **Distribution Rule Label** to be used as rule's name throughout the system.

Characteristic Type defines the type of entity whose balance is relieved by the payment(s) this rule creates. For example, if this rule targets payments(s) towards a specific service agreement, you'd reference a characteristic that its value identifies a service agreement. We use the term "rule value" for the characteristic value.

Distribution Rule - Algorithms

Navigate to **Admin**, and then **Financial**, and then **Distribution Rule**, and then **Search** and navigate to the **Algorithms** page to set up the algorithms appropriate for your distribution rule.

Description of Page

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description
Create Payment	Optional	This algorithm is executed to distribute a payment distribution detail payment amount into one or more payments.
Determine Tender Account	Optional	This algorithm is executed to determine the tender account associated with the payment distribution detail. Only one such algorithm may be specified.

Feature Configuration

You must set up a Financial Transaction Options Feature Configuration to define parameters that control various payment event distribution options.

The following points describe the various **Option Types** that must be defined:

Copyright © 2026, Oracle and/or its affiliates.

- Always Enable Distribution Rule. This option controls whether the system should only use the distribution rule method to add payment events or rather allow both the default method and the distribution rule method to coexist.
  - Enter Y if the system should always use distribution rules. With this setting, navigation to the Payment Event page in add mode opens up the Payment Event Quick Add page (defaulting it to the single payment event dialog). This dialog is designed to create a payment event using distribution rules
  - Enter N if the system should allow both methods. With this setting, navigation to the Payment Event page in add mode opens up the standard Payment Event - Add Dialog that uses the default method to create a payment event. If you want to use the distribution rule method, navigate to the Payment Event Quick Add page from the menu.
- Default Distribution Rule. This option states your default distribution rule that appears throughout the system.

Cancel Reasons

As described in The Financial Big Picture , the various types of financial transactions can be canceled if their financial impact needs to be reversed from the system. Whenever a financial transaction is canceled, a cancel reason must be specified. This section describes the control tables that contain the cancel reason codes.

- Setting Up Bill (Segment) Cancellation Reasons
- Setting Up Payment Cancellation Reasons
- Setting Up Adjustment Cancellation Reasons

Setting Up Bill (Segment) Cancellation Reasons

Open **Admin**, and then **Billing**, and then **Bill Cancel Reason** to define your bill segment cancellation reason codes.

Description of Page

Enter an easily recognizable **Bill Cancel Reason** and **Description** for the bill cancellation reason.

Only use **System Default** on those reason codes that are placed on bill segments that are automatically canceled by the system. Valid values are: Turn off auto-cancel, Bad estimated read auto-cancel, MDM Corrected Read, and Mass Cancel. The reason code identified as Turn off auto-cancel is placed on bill segments that are automatically canceled when the final bill segment ends before the prior bill (and therefore we have to cancel the prior bill). The reason code identified as Bad estimated read auto-cancel is placed on bill segments that are automatically canceled by the system when it detects that it used an estimated read whose consumption is greater than the next actual read (and therefore we have to cancel the estimated bill segment). The reason code identified as Mass Cancel is placed on bill segments that are canceled as a result of the execution of the Mass Cancellation background process. The reason code identified as MDM Corrected Read is placed on bill segments that are automatically canceled by the system when processing a corrected read notification. Corrected read notifications are received from MDM when reads that were used in bill determinant

Copyright © 2026, Oracle and/or its affiliates.

calculation requests (usage requests) are replaced or modified. This notification may result in the rebill of frozen bill segments. Refer to Usage Requests for more information.

> **Note**
>
> **Required values.** You must have one reason code defined for each of the System Default values.

Setting Up Payment Cancellation Reasons

Open **Admin**, and then **Financial**, and then **Pay Cancel Reason** to define your payment cancellation reason codes.

Description of Page

Enter an easily recognizable **Cancel Reason** and **Description** for the payment cancellation reason.

Turn on the **NSF Charge** switch if the system should invoke the non-sufficient funds (NSF) algorithm when a tender is cancelled using this reason code. Refer to NSF Cancellations for more information.

The next several fields are used to change an account's credit rating or cash-only points if a tender is canceled using the respective reason code.

- Use **Affect Cash-Only Score By** to define how tenders canceled using this reason will affect the account's cash-only score. This should be a positive number. When a customer's cash-only points exceed the cash-only threshold amount defined on the installation record or the account's CIS Division, the account is flagged as cash only during payment processing and on Control Central.
- Use **Affect Credit Rating By** to define how tenders canceled using this reason will affect the account's credit rating. This should be a negative number. A customer's credit rating is equal to the start credit rating amount defined on the installation record or the account's CIS Division plus the sum of credit rating demerits that are currently in effect.
- Use **Months Affecting Credit Rating** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

> **NOT_SUPPORTED**
>
> For more information, refer to Account - Credit Rating.

> **Note**
>
> **The payor gets the credit rating / cash only hit.** When you cancel a tender you must specify a cancellation reason. If the cancellation reason indicates a credit rating / cash only demerit should be generated, the system levies the credit rating transaction on the PAYOR's account.

Copyright © 2026, Oracle and/or its affiliates.

The **System Default** is specified on those cancellation reasons that are placed on payment segments that are automatically cancelled by the system. Valid values are: Re-opened Bill. The Re-opened Bill value is used as follows:

- Payments are automatically created for accounts who pay their bills automatically when their bills are completed.
- If such a bill is reopened before the automatic payment is interfaced to the paying authority, the system automatically cancels the payment. The Re-opened Bill cancellation reason is placed on such payments.

Setting Up Adjustment Cancellation Reasons

Open **Admin**, and then **Financial**, and then **Adjustment Cancel Reason** to define your adjustment cancellation reason codes.

Description of Page

Enter an easily recognizable **Cancel Reason** and **Description** for each adjustment cancellation reason.

Miscellaneous Financial Controls

This section describes miscellaneous control tables.

- A/P Check Request
- Billable Charge Line Type

A/P Check Request

Adjustments whose adjustment type is marked with an A/P check request code are interfaced to your A/P system. Your A/P system then cuts the checks.

> **NOT_SUPPORTED**
>
> Refer to Controls The Interface To A/P for more information about the accounts payable interface.

You must set up at least one A/P check request code if you want A/P to cut checks.

To set up A/P check request types, open **Admin**, and then **Financial**, and then **A/P Request Type.**

Description of Page

Enter an easily recognizable **A/P Request Type** for the accounts payable request type.

Use **Due Days** to define when the check is cut. The cut date is equal to the adjustment date plus due days.

Copyright © 2026, Oracle and/or its affiliates.

Select a **Payment Method**. Choose from these options:

System Check System check

Billable Charge Line Type

> **Note**
>
> **Background information.** Before using this page, you should be comfortable with the topics described under Setting Up Billable Charge Templates and Uploading Billable Charges.

Billable charge line types will simplify the effort required to interface billable charges from an external system. Each line type contains values that will be defaulted onto the line details associated with the uploaded billable charges. Obviously, this defaulting is possible only if you specify a billable charge line type on the billable charge upload staging lines.

To set up billable charge line types, select **Admin**, and then **Billing**, and then **Billable Charge Line Type**, and then **Add**.

Description of Page

Enter an easily recognizable **Billable Charge Line External Type** and **Description**.

Use **Currency Code** to define the currency to be defaulted onto billable charge upload lines that reference this line type.

Use **Show on Bill** to define the value to be defaulted into the Show on Bill indicator on billable charge upload lines that reference this line type.

Use **App in Summary** to define the value to be defaulted into the App in Summary indicator on billable charge upload lines that reference this line type.

Use **Memo Only, No GL** to define the value to be defaulted into the Memo Only, No GL indicator on billable charge upload lines that reference this line type.

Use **Distribution Code** to define the values to be defaulted into the Distribution Code field on billable charge upload lines that reference this line type.

Payables Cash Accounting

In some areas, taxes and other 3rd party liabilities are not truly payable until the customer remits payment. We refer to this as "payables cash accounting". This practice should be contrasted with "payables accrual accounting" in which the liability is realized when the bill is created (as opposed to when it is paid).

If your organization does not practice payable cash accounting, you may skip this section as accrual accounting is the system default. If you practice payables cash accounting, the contents of this section describe how to configure the system appropriately.

- Accrual versus Cash Accounting Example

Copyright © 2026, Oracle and/or its affiliates.

- Distribution Code Controls Cash Accounting For A GL Account
- Bill Segments and Cash Accounting
- Payment Segments and Cash Accounting
- Write Down Adjustment
- Write-Offs

Accrual versus Cash Accounting Example

The following is an example of the financial events that transpire when a customer is billed and payment is received using accrual accounting.

Event	GL Accounting	Tax Payable Balance
Bill segment created	A/R 110 Revenue <100> Tax Payable <10>	(10)
Payment received	Cash 110 A/R <110>	(10)

In the above example, you'll notice that the payable is booked when the bill is created. Let's contrast this with what takes place if the payable is subject to payables cash accounting.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0

Notice that when the bill segment is produced, the liability is not booked, rather, the amount of the liability is placed in a "holding" GL account. When the customer pays, the moneys are transferred from the "holding" GL account to the true tax payable account.

> **Note**
>
> **Cash accounting is only applicable for liabilities.** In the above example, you'll notice that only the tax payable account had cash accounting implications. This is because organizations that practice cash accounting only do it for liability accounts; they never do it for assets, revenue or expenses.

If the above seems simple, consider the following complications that must be considered:

- What happens if a partial payment is received?
- What happens if there are multiple taxes subject to cash accounting rules?

Copyright © 2026, Oracle and/or its affiliates.

- What happens if the A/R is relieved via a deposit seizure (or transference of a credit balance from another SA)?
- What if, after payment, the original bill segment is cancel/rebilled resulting in a different amount of tax (keep in mind that the payable got booked when the payment was received)?
- What happens if the payment is cancelled?
- What if the payment isn't received and we have to write-off debt?
- What happens if the customer overpays?
- What happens if the customer is allowed to prepay their tax (this is a common practice in the United Kingdom) and then the tax rate changes at billing time?

The above points, and more, are discussed below.

Distribution Code Controls Cash Accounting For A GL Account

> **Note**
>
> If you do not understand the significance of distribution codes, please refer to Setting Up Distribution Codes.

Whether or not cash accounting is used for a specific GL account is defined on HOLDING GL account's distribution code (i.e., the holding GL account references the true payable account).

It is very important that unique payable and holding distribution codes be used for each type of tax subject to cash accounting rules. For example, if you have cash accounting requirements for both value-added tax (VAT) and a climate levy, you would need four distribution codes:

- VAT Payable.
- VAT Holding.
- Climate Levy Payable.
- Climate Levy Holding.

Without unique distribution codes for each payable and holding account, the system cannot keep track of how much of a given tax is being held, awaiting payment.

Bill Segments and Cash Accounting

The contents of this section describe how cash accounting is implemented when bill segments are created.

- Bill Segment Financial Transactions Are Not Affected By Cash Accounting

Bill Segment Financial Transactions Are Not Affected By Cash Accounting

Copyright © 2026, Oracle and/or its affiliates.

There are NO changes to rate calculation associated with cash accounting. This is because the rate components that calculate tax reference the HOLDING payable distribution codes.

> **Note**
>
> **Prepaid taxes - future functionality.** If your organization allows customers to prepay taxes in anticipation of a future tax increase (the customers receive the lower rate if they pay in advance), please speak to your account manager for information about when corresponding functionality will be available.

Payment Segments and Cash Accounting

The contents of this section describe how cash accounting is implemented when payment segments are created.

- Payment Segment Financial Transaction Algorithms Transfer Holding Amounts to Payable GL Accounts
- How Does The System Know What Amounts To Transfer From Holding To Payables?
- Partial Payments Result In Partial Payables
- Partial Payments Using Accounting Priority

Payment Segment Financial Transaction Algorithms Transfer Holding Amounts to Payable GL Accounts

Logic exists in the pay segment's FT algorithm that transfers amounts from payable holding distribution codes to their respective payable real distribution codes.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Payment Segment Types for how to define the appropriate FT algorithm.

The following table shows what happens to the financial transaction associated with the payment segment for a cash accounting customer.

Event	GL Accounting
Bill segment is created	A/R 110 Revenue <100> Tax Holding <10>
Payment segment relieves receivables	Cash 110 A/R <110>
Additional GL details created when the payment segment FT algorithm transfers the holding amount to a payable account	Tax Holding 10 Tax Payable <10>

Copyright © 2026, Oracle and/or its affiliates.

Net effect of the above	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>

How Does The System Know What Amounts To Transfer From Holding To Payables?

When a payment segment is created for an account that is subject to cash accounting processing, the system determines if there is a CREDIT balance for any holding distribution code in respect of the service agreement. If so, it generates additional GL details to transfer moneys from the holding distribution code to the payable distribution code in proportion to the amount of receivables relieved by the payment. Therefore, if 100% of receivables are relieved by the payment segment, 100% of the holding amounts will be transferred to payable distribution codes. Refer to Partial Payments Result In Partial Payables for an example of what happens when a partial payment is created.

Partial Payments Result In Partial Payables

The previous example showed the entire tax holding amount being transferred to the tax payable account. The entire holding amount was transferred because the service agreement was paid in full. If a partial payment is received, only part of the holding amount will be transferred to the payable amount (proportional to the amount of receivables reduced by the payment). An example will help make the point.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Partial payment received	Cash 27.50 A/R <27.50> Tax Holding 2.50 Tax Payable <2.50>	(2.50)	(7.50)

The above example assumes the use of the base product payment segment FT creation algorithm PSEG-AC to transfer the holding amount to the tax payable account. If multiple holding accounts are used, you may want to specify which holding amounts are relieved first. The base product includes an additional payment segment FT creation algorithm C1-FTGL-PSAC that handles booking holding amounts based on a priority.

Partial Payments Using Accounting Priority

To book holding amounts based on a priority, each holding distribution code must be assigned an **Accounting Priority**. When a partial payment is posted, only part of the holding amount will be transferred to the payable amount (proportional to the amount of receivables reduced by the payment). When the holding amount consists of various holding distribution codes with different accounting priorities, the amount to transfer is allocated as follows:

Copyright © 2026, Oracle and/or its affiliates.

- Holding distribution codes associated with the oldest debt are settled first
- Within the same debt age, holding distribution codes with a higher accounting priority are booked first. If more than one distribution code shares the same priority, the settlement is distributed among them in proportion to the holding account balance

The above logic is handled by the payment segment FT creation algorithm C1-FTGL-PSAC. As an example of how these rules apply, let's assume an implementation practices cash accounting; i.e. revenue, taxes and other third party liabilities are not recognized until payment is received. Also assume the following distribution codes have been configured:

Holding Distribution Code	Description	Cash Accounting Distribution Code	Accounting Priority
HLD-LPC	Late Payment Charge	R-MISC	10
HLD-RGEN	Revenue - Generation Charge	R-GEN	20
HLD-RDIS	Revenue - Distribution Charge	R-DIST	30
HLD-RTRN	Revenue - Transmission Charge	R-TRAN	30
HLD-THRD	3 rd Party Charges	R-THRD	40
HLD-VAT	VAT	A/P-VAT	90

Assume a customer has an outstanding third party charge with an arrears date of 2/Jan/2009:

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Unpaid Amount	A/R 50 HLD-HRD <45> HLD-VAT <5>	50	0	0	0	0	(45)	(5)

A bill is created for a customer and the result of the bill's financial transactions (an LPC adjustment in the amount of 10 and a bill segment in the amount of 127) include the following FT GL lines (both financial transactions have an arrears date of 15/Jan/2009):

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT

Copyright © 2026, Oracle and/or its affiliates.

Bill segment created	A/R 127 HLD-RGEN <15> HLD-RDIS <20> HLD-RTRN <55> HLD-THRD <10> HLD-VAT <27>	177	0	(15)	(20)	(55)	(55)	(32)
Adjustment created	A/R 10 HLD-LPC <10>	187	(10)	(15)	(20)	(55)	(55)	(32)

No payment is received prior to the next bill. The result of the next bill's financial transaction (a bill segment in the amount of 100) includes the following FT GL lines (this financial transaction has an arrears date of 16/Feb/2009):

Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Bill segment created	A/R 100 HLD-RGEN <15> HLD-RDIS <20> HLD-RTRN <45> HLD-THRD <10> HLD-VAT <10>	287	(10)	(30)	(40)	(100)	(65)	(42)

The following shows the result if a customer makes a payment on 20/Feb/2009. At payment time we'll build a table of holding amounts by accounting priority and debt age as follows:

Distribution Code & Priority	HLD-LPC (10)	HLD-RGEN (20)	HLD-RDIS (30)	HLD- RTRN (30)	HLD-THRD (40)	HLD- VAT (90)
Debt Age						
4 days old		(15)	(20)	(45)	(10)	(10)
36 days old	(10)	(15)	(20)	(55)	(10)	(27)
49 days old					(45)	(5)

Copyright © 2026, Oracle and/or its affiliates.

- Examples of Partial Payments Using Accounting Priority
- Adjustments That Behave Like Payments
- Overpayment Of Taxes Due To Cancel/Rebills
- Cash Refunds
- Overpayments

Examples of Partial Payments Using Accounting Priority

The examples below assume a customer has the financial history described above and attempts to illustrate the financial effect when a payment is made.

- Example 1 - Customer Pays In Full
- Example 2 - Customer Makes a Partial Payment
- Example 3 - Customer Makes a Partial Payment

Example 1 - Customer Pays In Full

Assume the customer makes a payment in the amount of 287. This amount is sufficient to satisfy all holding amounts, so the payment will have the following financial effect:

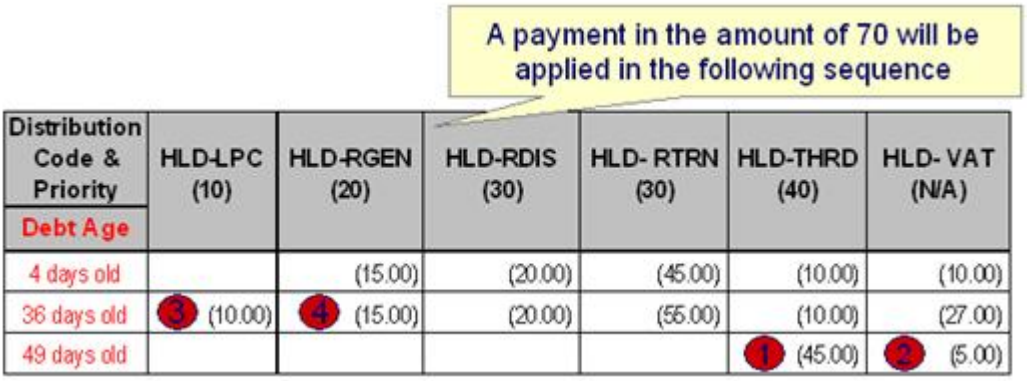
Event	GL Accounting	SA's Payoff Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 287 A/R <287> HLD-LPC 10 R-MISC <10> HLD-RGEN 30 R-GEN <30> HLD-RDIS 40 R-DIST <40> HLD-RTRN 100 R-TRAN <100> HLD-THRD 65 R-THRD <65> HLD-VAT 42 A/P-VAT <42>	0	0	0	0	0	0	0

Copyright © 2026, Oracle and/or its affiliates.

Top of the Page

Example 2 - Customer Makes a Partial Payment

Assume the same financial history described above for a customer and a partial payment in the amount of 70 is made. This amount is not sufficient to satisfy the total holding amounts of 287, so the system will start settling held amounts starting with distribution codes with the oldest debt first from highest priority until the payment amount is exhausted.



Distribution Code & Priority / Debt Age	HLD-LPC (10)	HLD-RGEN (20)	HLD-RDIS (30)	HLD- RTRN (30)	HLD-THRD (40)	HLD- VAT (N/A)
4 days old		(15.00)	(20.00)	(45.00)	(10.00)	(10.00)
36 days old	3 (10.00)	4 (15.00)	(20.00)	(55.00)	(10.00)	(27.00)
49 days old					1 (45.00)	2 (5.00)

The following describes how the holding amounts will be booked as a result of this partial payment:

- Settle oldest debt first (49 days old), i.e. 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). Note that even though these holding accounts have the lower accounting priorities, they are booked first because they have the oldest debt. An amount of 20 now remains on the partial payment.
- Next, we'll settle the 36 days old debt from the highest priority:
  - Late Payment Charge (HLD-LPC) in the amount of 10. An amount of 10 now remains on the partial payment
  - Revenue - Generation Charge (HLD-RGEN) gets the remaining payment amount of 10
- So, this partial payment in the amount of 70 will result in the following financial effect:

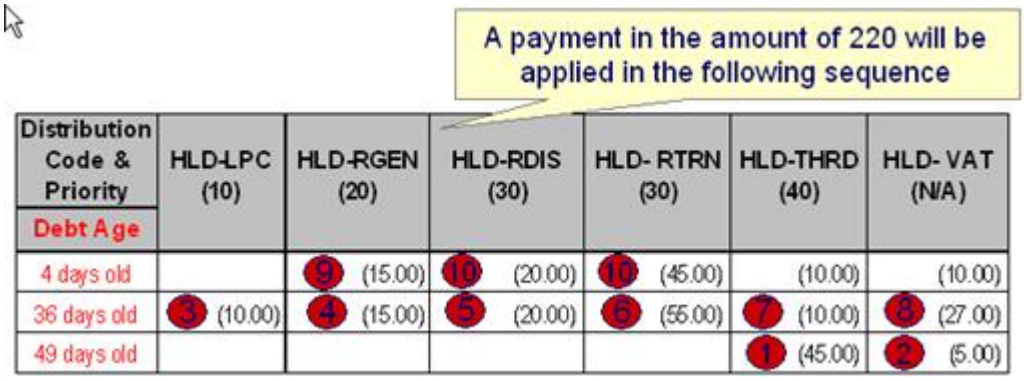
Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	SA Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 70 A/R <70> HLD-LPC 10 R-MISC <10> HLD-RGEN 10 R-GEN <10> HLD-THRD 45 R-THRD <45> HLD-VAT 5 A/P-VAT <5>	217	0	(20)	(40)	(100)	(20)	(37)

Top of the Page

Example 3 - Customer Makes a Partial Payment

Assume the same financial history described above for a customer and a partial payment in the amount of 220 is made. This amount is not sufficient to satisfy the total holding amounts of 287, so the system will start settling held amounts starting with distribution codes with the oldest debt first from highest priority until the payment amount is exhausted.



The following describes how the holding amounts will be booked as a result of this partial payment:

- Settle oldest debt first (49 days old), i.e. 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). Note that even though these holding accounts have the lower accounting priorities they are booked first because they have the oldest debt. An amount of 170 now remains on the partial payment.
- Next, we'll settle the 36 days old debt from the highest priority, i.e. Late Payment Charge (HLD-LPC), Revenue - Generation Charge (HLD-RGEN), Revenue - Distribution Charge

Copyright © 2026, Oracle and/or its affiliates.

(HLD-RDIS), Revenue - Transmission Charge (HLD-RTRN), 3rd Party Charges (HLD-THRD) and VAT (HLD-VAT). An amount of 33 now remains on the partial payment.

- Next we'll settle the 4 day old debt from the highest priority:
  - Revenue - Generation Charge (HLD-RGEN) in the amount of 15. An amount of 18 now remains on the partial payment
  - The two holding accounts at the next priority have an outstanding amount of 65. Since the remainder of the payment is not enough to satisfy this amount, the remainder of the payment is prorated amongst HLD-RDIS and HLD-RTRN as follows:
    * (Remaining Pay Amount / Total Outstanding Holding Amount) * Holding Account Amount
    * So for the Revenue - Distribution Charge (HLD-RDIS) holding account the amount booked will be (18/65 * 20) = 5.54
- So, this partial payment in the amount of 220 will result in the following financial effect:

Event	GL Accounting	SA Balance	Holding Balances					
			HLD-LPC	HLD-RGEN	HLD-RDIS	HLD-RTRN	HLD-THRD	HLD-VAT
Payment received	Cash 220 A/R <220> HLD-LPC 10 R-MISC <10> HLD-RGEN 30 R-GEN <30> HLD-RDIS 25.54 R-DIST <25.54> HLD-RTRN 67.46 R-TRAN <67.46> HLD-THRD 55 R-THRD <55> HLD-VAT32 A/P-VAT <32>	67	0	0	(14.46)	(32.54)	(10)	(10)

Adjustments That Behave Like Payments

Copyright © 2026, Oracle and/or its affiliates.

There are several types of adjustments that behave just like payments (in respect of payables cash accounting). Consider the following events:

- Seizing a deposit (i.e., transferring a credit from a deposit service agreement to a regular service agreement)
- Overpayments transferred from one service agreement to another

The above events should cause the system to transfer holding amounts to true payable amounts (notice that the above examples are all transfer adjustments).

However, there are many other adjustments that should NOT behave like payments. You control how the adjustment works by selecting the appropriate FT algorithm when you set up adjustment types (refer to ADJT-AC and ADJT-TC for a description of the base package algorithms that cause the holding amounts to be manipulated in proportion to the amount of receivable being adjusted; and to C1-FTGL-ADAC and C1-FTGL-ADTC for the base package algorithms that take **Accounting Priority** into consideration). In other words, there are adjustment FT algorithms that cause the transference of holding payable amounts to real payable amounts when the A/R balance is decreased by the adjustment.

> **Note**
>
> **Cash refunds can behave like "anti-payments".** In addition to the above examples of transfer adjustments behaving like payments, you should be aware that cash refunds may impact your holding and true payable balances. Refer to Cash Refunds for more information.

Overpayment Of Taxes Due To Cancel/Rebills

Let's assume a cancel / rebill occurs after a payment is received and the net effect of the cancel / rebill is that the customer has overpaid their taxes.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0
Cancel	A/R <110> Revenue 100 Tax Holding 10	(10)	10
Rebill	A/R 27.50 Revenue <25> Tax Holding <2.50>	(10)	7.50

You'll notice that the amount payable to the taxing authority still indicates $10 (the amount of tax that was paid by the customer). However, you'll notice that the tax holding balance is 7.50 (debit). This looks a bit odd, but it's correct. Remember that at this point, the customer has a

Copyright © 2026, Oracle and/or its affiliates.

credit balance of $75 and this will be whittled down as successive bills are produced as shown below. Note: refer to Cash Refunds for an example of what happens if you refund the credit with a check rather than letting it whittle down.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
		(10)	7.50
Bill segment created	A/R 55 Revenue <50> Tax Holding <5>	(10)	2.50
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	(10)	(7.50)

In the unlikely event of a payment being received while the tax holding has a debit balance, nothing will be done in respect of transferring funds from holding to payable (there is nothing to transfer).

Cash Refunds

If you refund moneys to a cash accounting customer, it's important to do the opposite of what was done when the payment was received (i.e., you need to transfer the payable back to the holding account). The following example should help clarify this situation (this example shows a refund due to a credit balance that occurred as a result of a cancel/rebill).

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance	SA's Payoff Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)	110
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0	0
Cancel	A/R <110> Revenue 100 Tax Holding 10	(10)	10	(110)
Rebill	A/R 27.50 Revenue <25> Tax Holding <2.50>	(10)	7.50	(82.50)
Payment refunded (via an A/P adjustment)	Cash <82.50> A/R 82.50 Tax Holding <7.50> Tax Payable 7.50	(2.50)	0	0

We understand this is tricky, but consider this - when a cash accounting customer makes a payment, the system transfers tax holding CREDIT balances to tax payable distribution codes in proportion to the amount of the receivable DEBIT amount that was reduced by the payment.

Copyright © 2026, Oracle and/or its affiliates.

Therefore, when cash is returned to the customer, the system should transfer tax holding DEBIT balances to tax payable distribution codes in proportion to the amount of the receivable CREDIT that was reduced by the refund.

> **Note**
>
> The above takes place when an A/P adjustment is created if the related adjustment type references the appropriate FT algorithm (refer to ADJT-AC and **C1-FTGL-ADAC** for a description of the adjustment FT algorithms used for adjustments that behave like payments).

Overpayments

An overpayment, by definition, does not "match" to open items. However, the match type algorithms supplied with the base package will result in a balanced match event if an overpayment is made. The following points explain how this is achieved:

- The base package's match type algorithms will distribute the payment until the customer's current debt is satisfied.
- The amount of the overpayment will be kept on a separate SA (this only happens if you plug-in the appropriate Overpayment Distribution algorithm on your customer classes). Refer to Overpayment Segmentation for more information.
- When the payment is frozen, the payment segments that satisfy current debt will be matched against their respective open-items. The payment segment used to book the overpayment (on the overpayment SA) will not be matched.
- When future bills are completed, the credit balance on this "overpayment SA" will be transferred to the "real SAs" when future bills are completed (if you have plugged in the appropriate bill completion algorithm on the overpayment SA's SA type). If the overpayment satisfies all newly calculated charges, a match event is created that matches the new charges against the funds transferred from the overpayment SA. Refer to When Are Match Events Created for information about how the system creates match events at bill completion time when the new charges on the bill are satisfied by other credits (overpayments, deposit refunds, etc.).
- At some point in the future, the overpayment will be exhausted (i.e., all funds will be transferred to "real SAs"). At that point in time, the overpayment SA will close (assuming you set up the overpayment SA's SA type as a "one time"). At close time, the system creates a match event that matches the original overpayment payment segment with the adjustments that were used to transfer funds to the "real SAs". Refer to When Are Match Events Created for information about how the system creates match events when a SA closes.

Write Down Adjustment

Writing down debt is very different from writing off debt. When you write down debt, you are removing the receivable with no expectation of it being paid. For example, most organizations write down small debit and credit balances as part of their write-off process (e.g., they don't send a very small amount to a collection agency).

Let's run through an example to illustrate this:

Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance	SA's Payoff Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)	110
Payment received	Cash 109.50 A/R <109.50> Tax Holding 9.95 Tax Payable <9.95>	(9.95)	.05	0.50
Write down cash accounting debt	Tax Holding 0.05 Write Down Expense 0.45 A/R <0.50>	(9.95)	0	0

In order to achieve the above, you must set up an adjustment type that references a special financial transaction algorithm (refer to non-accrual accounting write down algorithms ADJT-AD and **C1-FTGL-AD** for more information). This algorithm will reduce or increase the receivable balance accordingly AND cause any holding amounts to be set to zero. This adjustment type should be referenced on your write-down algorithm that is referenced on your write-off controls.

Write-Offs

At write-off time we may refund credit balances. The refunding of credit balances is handled by A/P adjustments and these have cash accounting processing as described under Cash Refunds.

If we have to write-off debt, holding balances are relieved in proportion to the amount of debt that is written off (as usual). It's important to understand that for this to work, you must set up the system as follows:

- The tax holding distribution codes must have their override distribution switch turned on.
- The distribution code on the SA type associated with the service agreement to which the written-off payables are transferred must be the REAL payable distribution codes. This is important so that if the customer pays after the payables are reversed, we will be able to debit cash and credit the REAL payable distribution code.

Let's run through an example to illustrate this.

Event	Normal SA GL Accounting	Write Off Revenue SA GL Accounting	Reverse Liabilities SA GL Accounting
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>		
Write Off Time			

Copyright © 2026, Oracle and/or its affiliates.

Reverse the held payables	Xfer 10 A/R <10>		Tax Holding 10 Xfer <10> Note, the tax holding only gets debited if you have turned on the override at write-off switch on its distribution code
Write off revenue	Xfer 100 A/R <100>	Write Off Expense 100 Xfer <100>	
If the customer subsequently pays		Cash 100 Write Off Exp <100>	Cash 10 Tax Payable <10> Note, the tax payable only gets credited if the SA type's distribution code has been defined as such

Deferred Accrual Accounting

Some implementations use a hybrid accounting method that combines cash and accrual accounting. In this case revenue, taxes, etc. are recognized on the earlier of the bill due date or the date payment is received. In this scenario, the cash accounting method is used up until the bill's due date, at which time the accrual method is enforced (let's call this "deferred accrual accounting"). A simpler flavor of deferred accrual accounting is when the revenue and liability recognition is done solely at bill due date regardless of when the payment is made.

Deferred accrual accounting affects distribution codes identified as **Use For Non-Accrual Accounting** with an associated **Accounting Method** of either Payable on Earlier of Payment or Due Date or Payable on Due Date. The system accomplishes the holding amount settlement on the bill due date using a customer class post bill completion algorithm C1-CR-BLRVWS that creates a bill review record to be processed on the bill's due date. The bill review batch process then analyzes these bill review records on the bill due date. If a bill review record is due for processing, the algorithm checks the outstanding balance of the holding accounts on each SA linked to the bill and creates a settlement adjustment for each SA.

- Deferred Accrual Accounting Examples
- Payment Cancellations and Deferred Accrual Accounting

Deferred Accrual Accounting Examples

The examples below illustrate the financial transactions that transpire under these different scenarios.

- Example 1 - Payable On Due Date, Customer Pays In Full
- Example 2 - Payable On Due Date, Customer Does Not Pay
- Example 3 - Payable On Due Date, Customer Makes A Partial Payment

Copyright © 2026, Oracle and/or its affiliates.

- Example 4 - Payable On Earlier Of Payment Or Due Date, Customer Pays In Full
- Example 5 - Payable On Earlier Of Payment Or Due Date, Customer Does Not Pay
- Example 6 - Payable On Earlier Of Payment Or Due Date, Customer Partially Pays

Example 1 - Payable On Due Date, Customer Pays In Full

The following is an example of the financial events that transpire when a customer is billed and full payment is received prior to the bill due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 2 - Payable On Due Date, Customer Does Not Pay

In the following example a customer is billed and no payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 3 - Payable On Due Date, Customer Makes A Partial Payment

In the following example a customer is billed and a partial payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)

Copyright © 2026, Oracle and/or its affiliates.

Payment received	Cash 27.50 A/R <27.50>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 4 - Payable On Earlier Of Payment Or Due Date, Customer Pays In Full

In the following example a customer is billed and full payment is received prior to the bill due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	(10)	0

Example 5 - Payable On Earlier Of Payment Or Due Date, Customer Does Not Pay

In the following example a customer is billed and no payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date. Note that if a payment is subsequently received after the settlement adjustment has been created, it's financial transaction(s) will not have any impact on the holding or liability accounts as these have already been booked.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Adjustment created on bill due date	Tax Holding 10 Tax Payable <10>	(10)	0

Example 6 - Payable On Earlier Of Payment Or Due Date, Customer Partially Pays

In the following example a customer is billed and a partial payment is received prior to the bill due date. The bill review batch process is responsible for creating the settlement adjustment for any outstanding holding amounts on the bill's due date. The accounting method in this case is Payable On Earlier Of Payment Or Due Date. Note that if a payment is subsequently received after the settlement adjustment has been created, it's financial transaction(s) will not have any impact on the holding or liability accounts as these have already been booked.

Copyright © 2026, Oracle and/or its affiliates.

Event	GL Accounting	Tax Payable Balance	Tax Holding Balance
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	0	(10)
Payment received	Cash 27.50 A/R <27.50> Tax Holding 2.50 Tax Payable <2.50>	(2.50)	(7.50)
Adjustment created on bill due date	Tax Holding 7.50 Tax Payable <7.50>	(10)	0

Payment Cancellations and Deferred Accrual Accounting

If a payment was responsible for transferring moneys from the holding distribution code to the payable distribution code, it stands to reason that if the payment is cancelled, it results in the reversal of this transfer from the holding distribution code to payable distribution code. If deferred accrual accounting is used and the payment is cancelled after the bill due date, the holding amounts that were transferred should remain booked.

Assume, for example, that the payment below was cancelled after the bill due date:

Event	GL Accounting
Payment received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>

At cancellation, the above entry will be reversed, reinstating the balance in the holding account:

Event	GL Accounting
Payment cancelled after bill due date	Cash <110> A/R 110 Tax Holding <10> Tax Payable 10

However, since the bill's due date has passed, the holding account needs to be booked. For open item accounts, the system comes with a customer class FT freeze algorithm (C1-PR-CA-RVS) that creates a bill review schedule for the affected bill, if one does not already exist. When the bill review batch process next runs, it checks the outstanding balance of the holding accounts on each SA linked to the bill and creates a settlement adjustment for each SA.

Event	GL Accounting
Adjustment created	Tax Holding 10 Tax Payable <10>

Copyright © 2026, Oracle and/or its affiliates.

Note that this solution is only applicable to open item accounting where the bill matched to the payment can be determined. If balance forward accounting is practiced, the bill or bills that the payment applied to cannot be determined. In this case, the next bill review record created for the account as part of billing will cause the balances of the holding accounts to be analyzed and the settlement will catch up at that point.

Open Item Accounting

The topics in this section provide background information about open-item accounting.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization practices open-item accounting. If your organization practices balance-forward accounting, you need only indicate such on your customer classes; no other setup is required. Refer to Open Item Versus Balance Forward Accounting for more information about these two accounting practices.

- Open-Item Versus Balance-Forward Accounting
- Accounting Method Defined On Your Customer Classes
- Match Events
- Disputing Items
- Pay Plans
- Over Payments
- Setting Up The System To Enable Open Item Accounting
- Setting Up Match Types
- Setting Up Match Event Cancellation Reasons

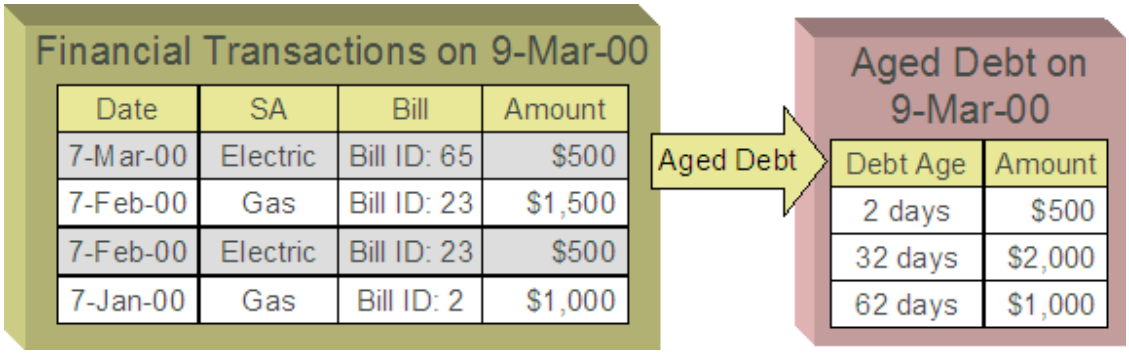
Open-Item Versus Balance-Forward Accounting

If you practice open-item accounting, you match payments against bills. The term "open-item accounting" is used to describe this accounting practice because:

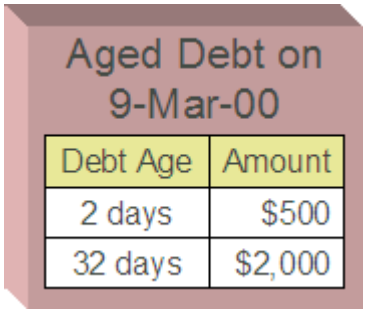
- Payments are matched against "open items" (i.e., unpaid bills and adjustments)
- Only unmatched bills and adjustments (i.e., open items) affect aged debt.

Contrast open-item accounting with "balance-forward" accounting - in a balance-forward world, payments are not matched to bills. Rather, payments implicitly relieve a customer's oldest debt. For example, consider the following unpaid financial transactions that exist for an account and the resultant aged debt.

Copyright © 2026, Oracle and/or its affiliates.

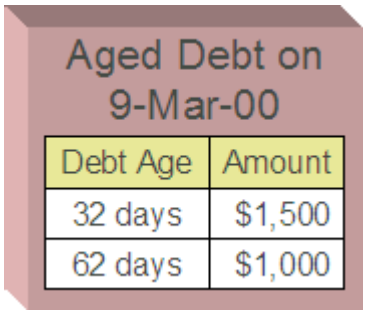


In a balance-forward world, if a $1,000 payment was made on 9-Mar-00, the customer's aged debt would look as follows:



Notice how the $1,000 payment relieves the 62 day old debt - it does this because, in a balance-forward world, payments payoff oldest debt first.

However, let's assume the customer wants the payment to settle their electric debt (e.g., because they disagree with the gas bills). If you could match the $1,000 payment to the two electric bills (i.e., open-item accounting exists), the customer's aged debt would look as follows:



In sum,

- In an open-item world, payments are matched to bills and only unpaid bills and adjustments (i.e., open items) affect aged debt.
- In a balance-forward world, payments are not matched to bills and therefore a customer's aged debt is computed by aging debits (bills and adjustments) and then relieving the oldest debits using credits (payments and adjustments).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Financial Transactions and Bills**. In an open-item world, only bill segments and adjustments are presented on a bill. When a bill is completed, only those bill segments and adjustments to be presented are swept onto a bill. Payment and payment cancellation FTs, bill segment FTs canceled before bill completion together with their corresponding bill segment cancellation FTs, and adjustment FTs marked as do not show on bill are not swept onto a bill. An adjustment's adjustment type and its algorithms determine if its FT will show on a bill by default.

Accounting Method Defined On Your Customer Classes

You define the type of accounting method that is practiced (balance-forward versus open-item) on your customer classes. For example, residential customers can practice balance-forward accounting whereas industrial / commercial customers can practice open-item accounting.

Match Events

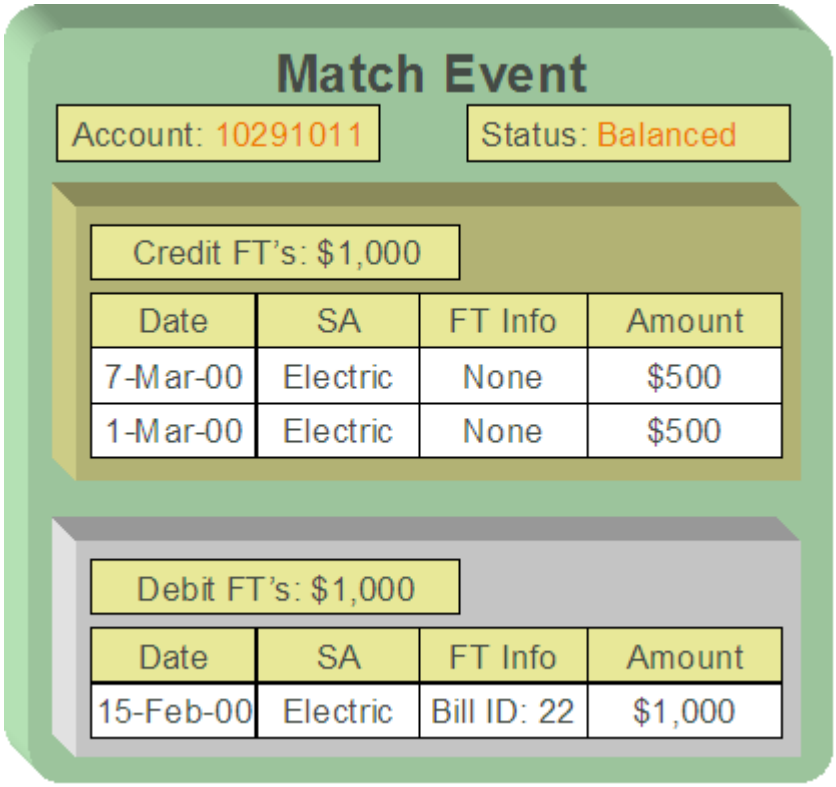
Match events are used to match open-items (i.e., debit and credit financial transactions) together. The topics in this section provide an overview of match events.

- Match Events Match Debit FTs To Credit FTs
- When Are Match Events Created?
- Match Event Lifecycle
- Payments And Match Events
- Payments Are Matched To Debit Credit FTs
- How Are Match Events Cancelled?
- Current Amount Is Matched, Not Payoff

Match Events Match Debit FTs To Credit FTs

For open-item customers, the system matches credit financial transactions (FT's) to debit FT's under a match event. The following is an example of a match event associated with two $500 payments that satisfy the debt associated with one bill (on February 2000).

Copyright © 2026, Oracle and/or its affiliates.



Notice the following:

- The match event matches 2 credit FT's against a single debit FT. A match event may contain an unlimited number of FT's.
- The match event contains FT's associated with a single account. While the FT's under a match event may belong to multiple service agreements, all FT's under a match event must belong to the same account.
- The match event only contains bill segments that belong to a single bill. If you mix multiple bills under a single match event, then an individual bill balance cannot be properly determined when partial payments exist.
- The status of the match event is balanced. This is because the sum of the debits equals the sum of the credits. If debits do not equal credits, the status of the match event would be open and the various FT's would still affect the customer's aged debt. Refer to Match Event Lifecycle for more information.

Warning

It is strongly encouraged that you refrain from mixing multiple bills on a single match event. If you stick by the rule of "just one bill per match event" you will then be able to determine the outstanding balance of a partially paid bill (see the bill page, bill summary section). However, if you mix more than one bill under a match event, a particular bill's balance may become indeterminate. Algorithm types have been provided which help to enforce this rule of "one bill per match event", please refer to Match By Bill, Pay Oldest Bill First for an example of a matching algorithm that enforces this notion.

Copyright © 2026, Oracle and/or its affiliates.

When Are Match Events Created?

The following points describe when match events are created for open-item accounts:

> **Note**
>
> Match events are only created for open-item accounts (i.e., those accounts with a customer class that indicates open-item accounting is practiced). Match events may not be created for balance-forward accounts.

- The system can create one or many match events when a payment is added. This match event matches the payment's credit FT's with the debit and credit FT's from bill segments and adjustments. The FT's that are linked to the match event are controlled by the payment's **match type** and **match value** (payments made by open-item customers must reference a match type and match value). Refer to Payments And Match Events for more information.
- The system may create a match event when any type of financial transaction is cancelled. This match event groups together the original FT with its cancellation FT. Refer to How Are Match Events Cancelled? for more information.
- The system creates a match event when a bill is completed for customers that pay automatically (i.e., direct debit customers). The match event groups together the bill's new charges against the automatic payment's payment segments.
- The system creates a match event when a bill is completed where the new charges are offset by other financial transactions. For example,
  - Consider a bill that contains a deposit refund. If the sum of the deposit refund equals or exceeds the amount of the bill, the bill's FT's can be matched against the debit refunds FT's. Refer to Refunding Deposits for more information about deposit refunds.
  - Consider a bill whose new charges are offset by a previous overpayment. Refer to Over Payments for more information about overpayments.

> **NOT_SUPPORTED**
>
> Refer to Bill Lifecycle for more information about what happens during bill completion.

- The system creates a match event when a service agreement closes and the service agreement has unmatched FT's. For example,
  - Consider a deposit service agreement that closes when the deposit is refunded to the customer. The system will create a match event with the deposit SA's FT's (the original credit and the debits used to refund the deposit) when the deposit SA closes (i.e., when its credit balance falls to zero). Refer to Refunding Deposits for more information about deposit refunds.
  - Consider a service agreement for utility debt that is written off. This service agreement closes when the system creates transfer adjustments to transfer the utility debt to a write-off service agreement (or writes down the debt). The system creates a match event to match the original debt to the transfer adjustments used to write-off the debt.

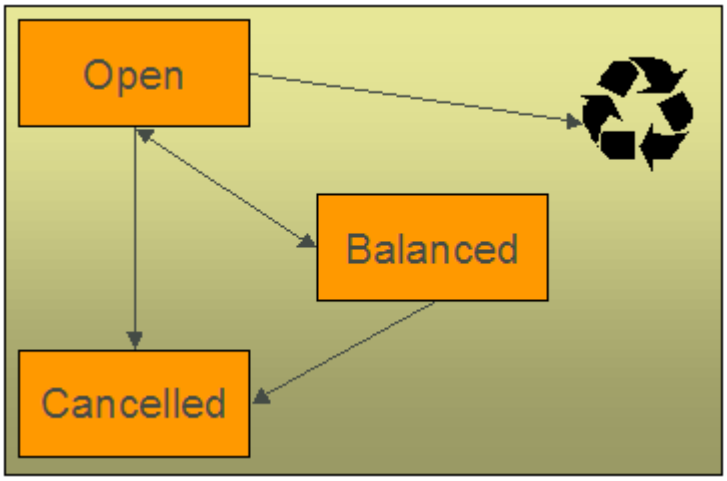
Copyright © 2026, Oracle and/or its affiliates.

Refer to How Is Deb`t Financially Written Off for more information about write-off processing.

- A user can create a match event manually at any time. Manual match events would be created under a variety of situations. For example:
  - If a customer disputes a charge. Refer to Disputing Items for more information about disputes.
  - To handle unusual situations when the system is unable to automatically match FT's together.

Match Event Lifecycle

The following diagram shows the possible lifecycle of a match event:



Match events are initially created in the open state. Financial transactions (FT's) linked to open match events affect arrears, but not in an open-item fashion. Rather, FT's linked to open match events affect arrears in a balance-forward fashion. Refer to Open Item Versus Balance Forward Accounting for more information about these two accounting methods.

A user may delete an open match event. When an open match event is deleted, its FT's may be linked to other match events.

The system automatically changes an open event's status to balanced when the sum of the debit financial transactions (FT's) equals the sum of the credit FT's for each SA on the match event. It's worth stressing that a match event may contain FT's from many SAs and each SA's FT's must sum to zero before the match event can become balanced.

A user may re open a balanced event (by adding / removing FT's so that the match event becomes unbalanced).

A user may cancel a balanced or open match event. Refer to How Are Match Events Cancelled? for more information about cancellation.

Payments And Match Events

As described under When Are Match Events Created?, the system creates a match event when a payment is added for an open-item account. The system uses the payment's **match**

Copyright © 2026, Oracle and/or its affiliates.

type and **match value** to determine the FT's (e.g., bill segments and adjustments) that will be matched with the payment's FT's (i.e., the payment segments).

Another way to think of this is as follows:

- When most payments are distributed, the system calls the payment distribution algorithm that is plugged-in on the account's customer class.
- However, a payment that is made in respect of a specific bill requires a different distribution algorithm because the payment should only be distributed amongst the debt associated with the specific bill being paid. This is accomplished by referencing a match type / match value on the payment. The match type references the appropriate payment distribution algorithm. This algorithm is used rather than the customer class distribution algorithm.

For example, if a payment were made in respect of bill ID 192910192101, this payment would reference a match type of bill ID and a match value of 192910192101. At payment distribution time, the system calls the override payment distribution algorithm associated with this match type. The base package bill ID distribution algorithm does several things:

- It distributes the payment amongst service agreements associated with the bill.
- It creates a match event and links the bill's bill segment and adjustment FT's to it.
- Refer to the Bill ID Match Type Algorithm for more information about this algorithm.

> **Note**
>
> **The match type's distribution logic is not "hard coded".** Because the match type's payment distribution logic is embedded in a plug-in algorithm, you can introduce new algorithms as per your company's requirements.

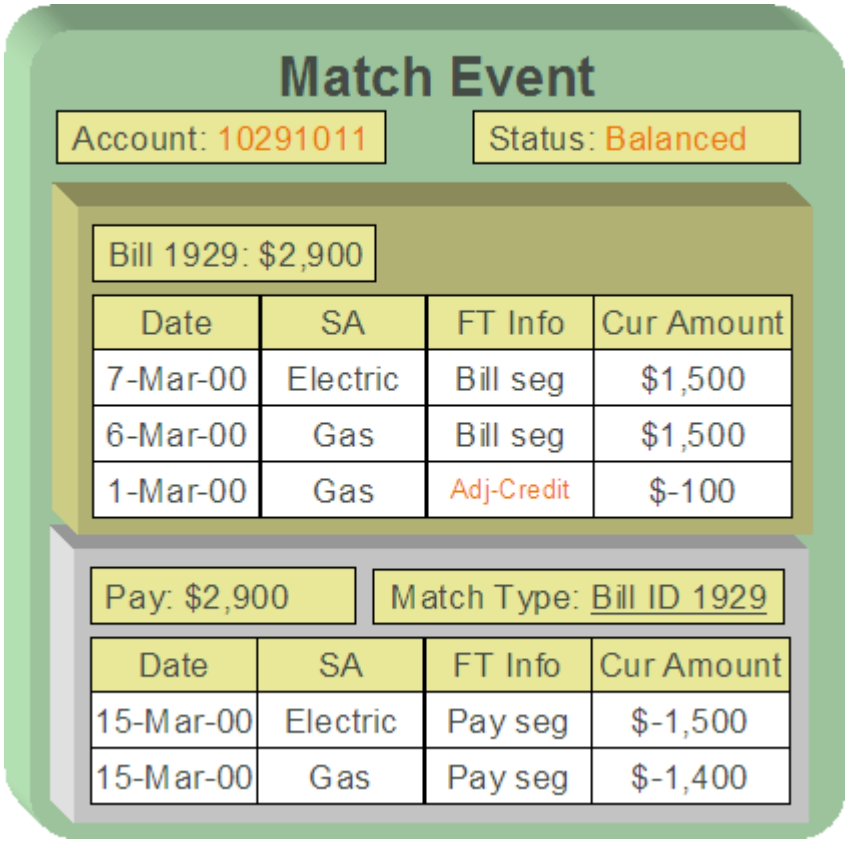
It's worth noting that payment *distribution* and *freezing* are two separate steps that typically happen in quick succession. The system's standard match event algorithms create the match event during payment distribution. This match event exists in the open state (because the payment segment's FT's have not yet been linked to the match event and therefore debit FT's do not equal credit FT's). The open match event references the debit FT's (the bill segments and adjustments) for which it pays. It is only at payment freeze time that the credit FT's (the payment segments) are linked to the match event thus allowing the match event to be become balanced.

If, at freeze time, the payment's credit FT's do not equal the debit FT's on the match event, the match event is left in the open state. An alert will appear on Control Central to highlight the existence of open match events (if the appropriate alert algorithm is plugged in the installation record). In addition, you can also set up a To Do entry to highlight the existence of open match events.

Payments Are Matched To Debit Credit FTs

While the above discussion dealt with the typical situation where the payment's credit FT's are matched against a bill's debit FT's, we want to note that a payment's FT's may be matched against debit and credit FT's. Consider the following example:

Copyright © 2026, Oracle and/or its affiliates.



Notice that:

- The $2,900 payment is distributed amongst two service agreements (electricity and gas).
- The FT's to which the payment segments are matched are both debit and credit FT's. Notice that the debit FT's (the bill segments) and the credit FT (the adjustment) sum to $2,900.

Credits may result in a situation where the total amount on a bill for an SA is negative. This would be the case if in the above example the credit adjustment were for $-1600 resulting in the total amount for the Gas SA on this bill to be $-100 (credit). Assume a full payment of $1400 is made towards this bill. The Bill ID Match Type Algorithm first allocates negative payment amounts to any SA credit amount on the bill being paid. It then carries over the credit amount to pay off other bill amounts. In this example, a "negative" payment segment is created to match the $-100 credit of the Gas SA. Using the carried over credit a $1500 payment segment is created to match the $1500 debit of the Electric SA.

How Are Match Events Cancelled?

A user can cancel an open or balanced match event at any time. When a match event is cancelled, the event's FT's again effect arrears (and they can be associated with new match events). In other words, when a match event is cancelled, its FT's are released from the match event and become open-items.

In addition to manual cancellation, the system may automatically cancel a match event when the last of its payment FT's, if any, is cancelled (if you plug-in the appropriate FT freeze plug-in on your open-item customer classes).

Copyright © 2026, Oracle and/or its affiliates.

For example, consider a match event that was created when a payment was made. If the payment is subsequently cancelled, the match event is also cancelled (thus releasing the match event's FT's) if no other payment FT's are linked to the match event. Please be aware that FT cancellation also causes a new match event to be created. This match event matches the original FT (the payment segment) and its cancellation FT. This means that the only "open items" that will exist after a payment is cancelled are the debit FT's that were originally paid.

> **Note**
>
> **Reopening bills associated with automatic payment customers.** While many payments are cancelled due to non-sufficient funds, please be aware that if you reopen a bill for which an automatic payment was created, the system will cancel the associated payment. If this payment is associated with a match event (because the account is an open-item account), the match event will be cancelled and a new match event will be created to match the original automatic payment with its cancellation details. This is necessary because a new payment will be created with the bill is subsequently completed and this payment's FT's will be matched to the bill's FT's.
>
> **Canceling a payment can result in many match events being created.** If a cancelled payment has multiple payment segments, a separate match event will be created for each payment segment.

While payment cancellation is the most common type of FT cancellation, be aware that bill segment or adjustment cancellation may also cause a new match event to be created. We don't necessarily want to always link the cancellation FT and its original FT to the same match event. For example, when the cancellation FT is swept on to the next bill it affects the next bill and not the original FT's bill. For cancellations that will not be swept on to the next bill (payment cancellation, cancellation of an adjustment that is not shown on bill, credit notes, and bill segment cancellation before the bill is completed) the system creates a new match event that matches the original FT and its cancellation FT. This way, neither FT affects aged debt. If the original FT was linked to an existing match event and no other FTs are left on this match event it is automatically canceled.

Current Amount Is Matched, Not Payoff

The system matches the current amount of financial transactions, not the payoff amount.

> **NOT_SUPPORTED**
>
> Please refer to Current Amount versus Payoff Amount for more information about current and payoff amounts.

Disputing Items

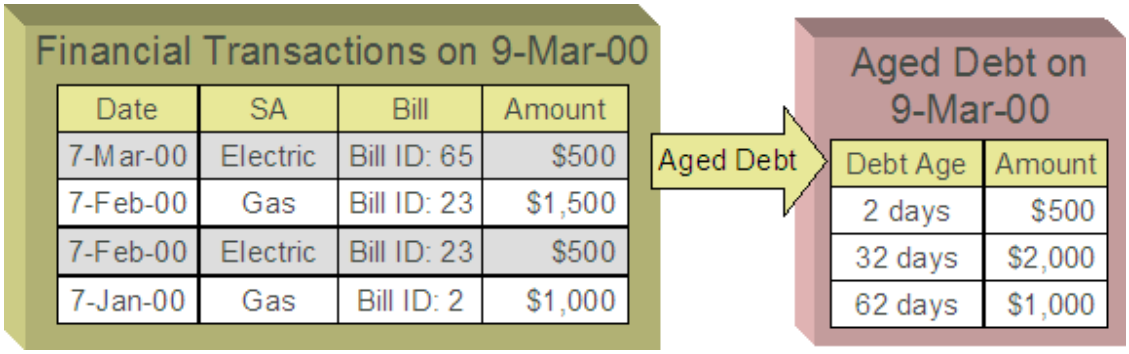
Open-item customers may dispute FT's that they are not comfortable paying. For example, a customer who receives a bill with an anomalous charge may decide to dispute it.

Copyright © 2026, Oracle and/or its affiliates.

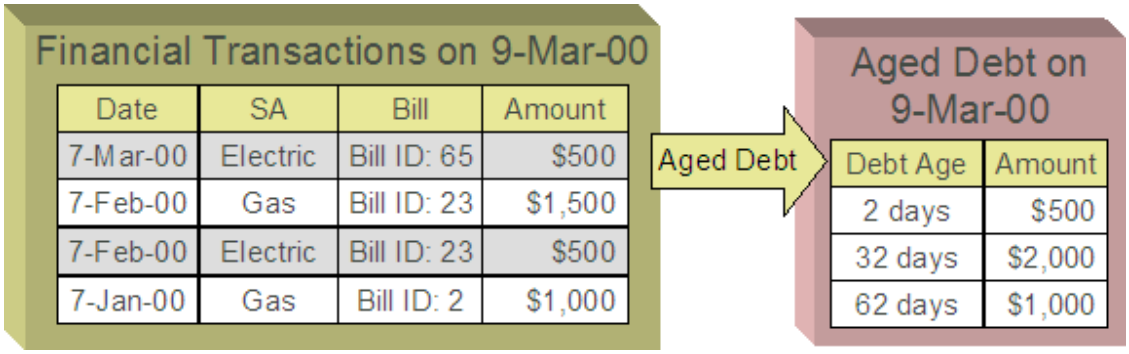
When an open-item customer disputes a charge, a user creates a match event and links the disputed FT(s) to it. This match event will be in the open state (because it does not contain FT's that sum to zero). In addition, the match event's "disputed switch" is turned on.

> **Note**
>
> **Alerts.** An alert is displayed on control central to highlight the existence of disputed match events (if the appropriate alert algorithm is plugged in). In addition, you can also set up a To Do entry to highlight the existence of disputed match events.

While the dispute is being researched, the disputed amount will not affect aged debt, but it still forms part of the customer's balance. For example, consider the following unpaid financial transactions that exist for an account:



If the customer disputes the two electric bill segments, the customer's aged debt will look as follows:

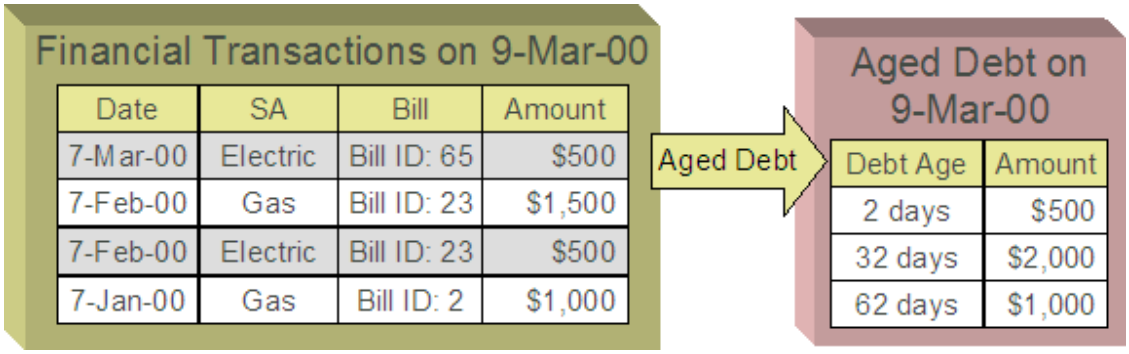


Notice how a new category of debt appears - Disputed. Also notice how the 2 day old debt disappears and the 32 day old debt is reduced by the disputed amount.

The system shows disputed debt on Control Central. In addition, in all places where aged debt appears in the system, disputed debt is shown as a separate debt category.

If the dispute goes in your company's favor, the disputed match event should be cancelled (thus allowing the FT's to again impact aged debt). For example, if we assume 3 days have passed and the dispute match event is cancelled, the customer's aged debt will look as follows:

Copyright © 2026, Oracle and/or its affiliates.



If the dispute goes in the customer's favor:

- You may decide to issue a credit note to cancel the offending bill or bill segment(s). As described above, the system in this case will automatically create new match events that match the original FTs with their cancellation FTs and cancel the disputed match event when the last item is unlinked from it.
- You may decide to cancel the offending bill segment(s) / adjustment(s). As described above, these cancellations are going to be swept on to the next bill. The system therefore will not automatically cancel the disputed match event. Notice that the cancellation effect of the disputed items is carried over on to the next bill. This means that the previously disputed items still need to be paid.

> **Note**
>
> **Cancel / rebill.** If you cancel / rebill an offending bill segment, both the cancel and the rebill will become open-items that will be matched when the next bill is paid.

- You may decide to issue an adjustment to counter the effect of the disputed FT's. In this situation, you would simply link the adjustment FT to the disputed FT's (thus allowing the match event to become balanced). It is important to use in this case an adjustment that does not show on bill.

Pay Plans

You create a pay plan when a customer agrees to make one or more scheduled payments to satisfy past (or future) debt. These payments cannot be matched to open items because it is unlikely that debit FT's exist that equal the amount of each scheduled payment. However, you must specify a match type on all payments made by open-item customers. Therefore, a conundrum exists - the system requires a match type on payments made by open item accounts, but payments made for pay plans cannot be matched to existing FT's. This conundrum is solved by the fact that match types do not have to specify an override payment distribution algorithm. The customer class's standard distribution algorithm is used for payments that reference such a match type.

You may wonder how these payments will eventually get matched to open items? If ALL payments associated with a pay plan occur before the next bill is paid (or if the pay plan exists to satisfy future debt), these payments will be swept onto the match event that is created when the customer pays their next bill. However, if the pay plan exists to payoff historical debt and this debt has not been entirely paid by the time of the next bill, an unmatched event will exist when the customer pays their subsequent bills (if the payment amount doesn't match the

Copyright © 2026, Oracle and/or its affiliates.

amount of new charges on the bill). Why? Because, the customer is not paying the entire amount of the bill and therefore the system will not be able to match the payment to open items. If this occurs, we recommend canceling the match events that are created when the customer pays their subsequent bills. When the customer finally pays off all outstanding debt, the system will create a single match event that will contain all payments and bill segments.

Over Payments

If a customer overpays a bill (i.e., we receive more cash than receivables), we strongly recommend you set up the system to NOT keep the excess credit on the customer's regular service agreements. Rather, we recommend you segregate the receivable onto an "excess credit" service agreement. If you do this, the system will transfer any excess credits to the regular service agreements at bill completion time. When this transfer occurs, the same accounting described under Payments Segment Financial Transaction Algorithms Transfer Holding Amounts To Payable GL Accounts occurs as shown in the following example. Note: this example assumes an excess credit of $110 was transferred to a normal service agreement and the normal service agreement had $10 of held payables.

NOT_SUPPORTED

Refer to Overpayment Segmentation for how to set up the system to segregate overpayments on a separate service agreement.

Note

Why not keep excess credits on a customer's regular service agreement? Because the system can't differentiate between a credit that exists as a result of an overpayment and a credit that exists because of cancel/rebills, it would be impossible for the system to know if payables should be realized as a result of the reduced credit balance. However, if you keep overpayments on an excess credit service agreement, the system knows to treat any transference of these credits as "payments" and therefore it can transfer holding balances to true payables.

Event	Normal SA GL Accounting	Excess Credit SA GL Accounting
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	
Payment of $300 is received	Cash 110 A/R <110> Tax Holding 10 Tax Payable <10>	Cash 190 Overpay <190>
Bill segment created	A/R 110 Revenue <100> Tax Holding <10>	
Transfer excess credit amount to normal service agreement (when bill is completed).	Xfer 110 A/R <110>	Overpay 110 Xfer <110>

Copyright © 2026, Oracle and/or its affiliates.

Because the transfer adjustment is the equivalent of a cash relief outstanding tax holding is relieved in proportion to the amount of receivables that are reduced by the transfer	Tax Holding 10 Tax Payable <10>	
Net effect of the transfer	Xfer 110 A/R <110> Tax Holding 10 Tax Payable <10>	Overpay 110 Xfer <110>

> **Note**
>
> **Prepaid taxes - future functionality.** If your organization allows customers to prepay taxes in anticipation of a future tax increase (the customers receive the lower rate if they pay in advance), we do not consider this prepayment to be an overpayment. Rather, it is a payment of future taxes that will be remitted to the taxing authorities at payment time (due to cash accounting). Please speak to your account manager for when corresponding functionality will be available.

Setting Up The System To Enable Open Item Accounting

The following section provides an overview of how to enable open-item accounting.

- Match Type Setup
- Match Event Cancellation Reason Setup
- Customer Class Setup
- Overpayment SA Type Setup
- Installation Record Setup
- To Do Entry Setup

Match Type Setup

The number of match types that you will need is dependent on the number of ways you want payments to be matched to open items. At a minimum, you will probably need the following match types:

- **Bill ID.** This match type should reference an override payment distribution algorithm that distributes the payment based on the bill ID specified on the payment (in match value). Refer to Payments And Match Events for more information.
- **SA ID.** This match type should reference an override payment distribution algorithm that distributes the payment based on the SA ID specified on the payment (in match value). Refer to Payments And Match Events for more information.

Copyright © 2026, Oracle and/or its affiliates.

- **Pay Plan.** This match type should NOT reference an override payment distribution algorithm (if this algorithm is blank, the customer class's payment distribution algorithm is used). Refer to Pay Plans for more information.

Match Event Cancellation Reason Setup

The number of match event cancellation reasons that you will need is dependent on the number of ways your organization can justify the cancellation of a match event. At a minimum, you will probably need the following match event cancellation reasons:

- **FT Cancellation.** This cancel reason should be referenced on the Customer Class FT Freeze algorithm that is responsible for canceling match events when one of its financial transactions is cancelled.
- **Incorrect Allocation.** This cancel reason should be specified by users when they cancel match events that were created by the system erroneously.

Customer Class Setup

The following points describe customer class oriented set up functions:

- Turn on the open-item accounting switch.
- Set up the following algorithms for each CIS division:
  - Specify a **payment freeze** algorithm that causes a payment's FT's to be linked to the match event that was created when the payment was distributed. Refer to Payments And Match Events for more information.
  - Specify a **FT freeze** algorithm that causes match events to be cancelled (and a new match event to be created) when a FT is cancelled. Refer to How Are Match Events Cancelled for more information about cancellation.
  - We strongly recommend specifying an **overpayment** algorithm that causes overpayments to be segregated onto an "excess credit / overpayment" SA. Refer to Overpayments for more information.

Overpayment SA Type Setup

Specify a **bill completion** algorithm that causes the credit amount on overpayment SAs to be transferred to newly create debt (created when the bill is created). This algorithm transfers an overpayment SA's balance to regular SAs and creates a match event if the overpayment covers the entire bill. Refer to Overpayments for more information.

Installation Record Setup

Specify an **automatic payment** algorithm that causes a match event to be created when automatic payments are created for open-item accounts. The base package algorithm will do this for you if you specify the appropriate parameter on the algorithm. Refer to APAY-CREATE for more information about this algorithm.

Copyright © 2026, Oracle and/or its affiliates.

If you want a Control Central alert to highlight when the current account has any open match events, plug in the appropriate **control central alert** algorithm on your installation record. Refer to C1-OPN-MEVT for more information about this algorithm.

If you want to enable manual pay segment distribution for open item accounts, along with other functions, you will need to plug in an installation algorithm for bill balance calculation. Refer to C1-OI-BI-AMT for more information about this algorithm.

To Do Entry Setup

Two To Do types are supplied with the base package:

- **TD-MODTL**. This To Do type highlights the presence of open, disputed match events.
- **TD-MONTL**. This To Do type highlights the presence of open, non-disputed match events.

Each of the above To Do types should be configured with the roles that work on entries of each type.

In addition, the account management group and/or divisions from which the default roles are extracted should be updated to define the role that should be defaulted for each of the above To Do types.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of To Do Lists for more information about To Do lists.

Setting Up Match Types

Most payments are distributed amongst service agreements using the payment distribution algorithm specified on the payment's account's customer class. This algorithm decides how to distribute a payment amongst an account's existing debt if the customer doesn't specify how the payment should be distributed.

A customer can specify how a payment is distributed by specifying a match type and match value on their payments. Consider the following examples:

- Customers that are subject to open-item accounting (this is defined on the account's customer class) tell the system exactly which debt is covered by their payments. For example, an open-item customer might make a payment in respect of bill ID 123919101919.
- Even non open-item customers can direct payments to specific SAs. For example, the system allows a balance-forward customer's payment to be directed to a specific service agreement (however, they cannot direct payments to specific bills as only open-item customers can do this).

Match types are used to define the specific type of debt that is covered by a payment. The match type contains the algorithm that effectively overrides the standard payment distribution algorithm defined on the account's customer class.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Background information.** Please refer to Payments And Match Events and Match Type Setup for more information about how match types are used.

To set up match types, select **Admin**, and then **Financial**, and then **Match Type.**

Description of Page

Enter an easily recognizable **Match Type** and **Description**.

Define the **Pay Dist Override Algorithm** used to distribute payments that reference this match type. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that overrides the normal payment distribution algorithm.

Setting Up Match Event Cancellation Reasons

When a match event is canceled, a cancel reason must be supplied.

> **Note**
>
> **Background information.** Refer to How Are Match Events Cancelled? and Setting Up Match Event Cancellation for more information about cancellation.

To set up match event cancellation reasons, select **Admin**, and then **Financial**, and then **Match Event Cancel Reason.**

Description of Page

Enter an easily recognizable **Match Event Cancel Reason** and **Description**.

Fund Accounting

The topics in this section provide background information about fund accounting.

> **Note**
>
> **This section is only relevant for some organizations.** The system configuration requirements described in this section are only relevant if your organization practices fund accounting (this type of accounting is typically performed by municipal utilities). If your organization does not practice fund accounting, you need only indicate such on the installation Record; no other setup is required.

Copyright © 2026, Oracle and/or its affiliates.

- Fund Accounting Overview
- Accounting Method Is Defined On The Installation Options
- Fund Controls Fund-Balancing Entries
- Building Fund-Balancing GL Details
- Setting Up The System To Enable Fund Accounting

Fund Accounting Overview

Municipal utilities, and not-for-profit organizations in general, often use a form of accounting different from that used by for-profit corporations. Municipal utilities typically practice fund accounting, whereas corporations practice corporate accounting.

Regulations or other restrictions may require a municipal utility to account for the finances of each of its departments as a separate entity. If a municipal utility provides both water and wastewater service, a municipal utility may need to track the receivables, revenue, and liabilities for water service separately from those of wastewater. In contrast, a corporation is free to co-mingle the moneys of the two services.

To track the services separately, the municipal utility sets up a fund for each department. A fund is an accounting entity with its own self-balancing set of accounts. Each fund has its own "sub general ledger" with its own chart of accounts, and within each fund, its debits equal its credits at all times. This allows the utility to report on the financial state of each fund independently.

In addition to having a fund for each department, there is also a general fund, which is used to handle inter-fund transfers as well as shared accounts.

- Fund Accounting Example
- An Example Of A Bill Segment That References Multiple Funds

Fund Accounting Example

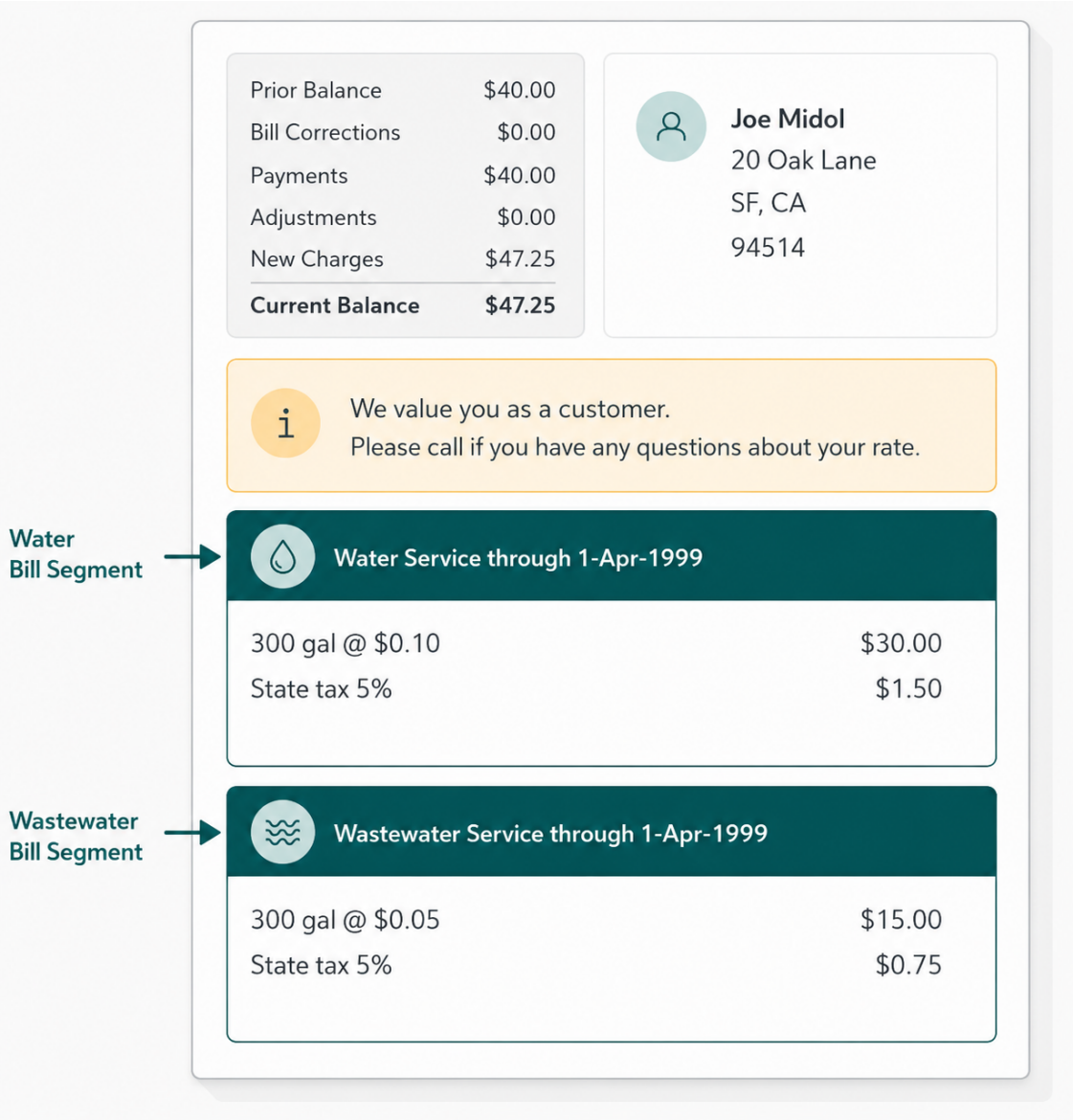
Consider a municipal utility which provides water and wastewater service. The utility has two departments: water and wastewater. Each department must track their finances separately therefore a fund is setup for each department:

- Water (fund 01).
- Wastewater (fund 02).

In addition, with fund accounting, there is always a general fund (fund 99).

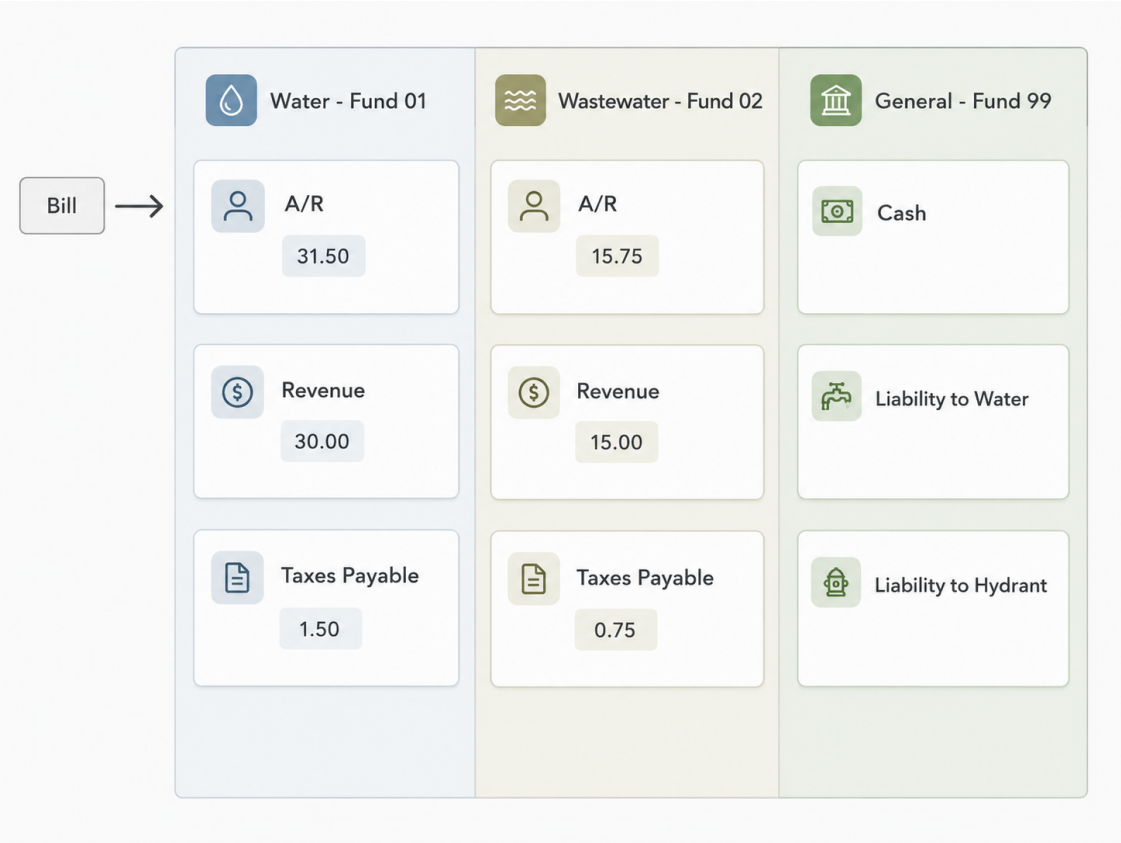
Assume the following bill is generated.

Copyright © 2026, Oracle and/or its affiliates.



The bill would produce the following GL entries:

Copyright © 2026, Oracle and/or its affiliates.

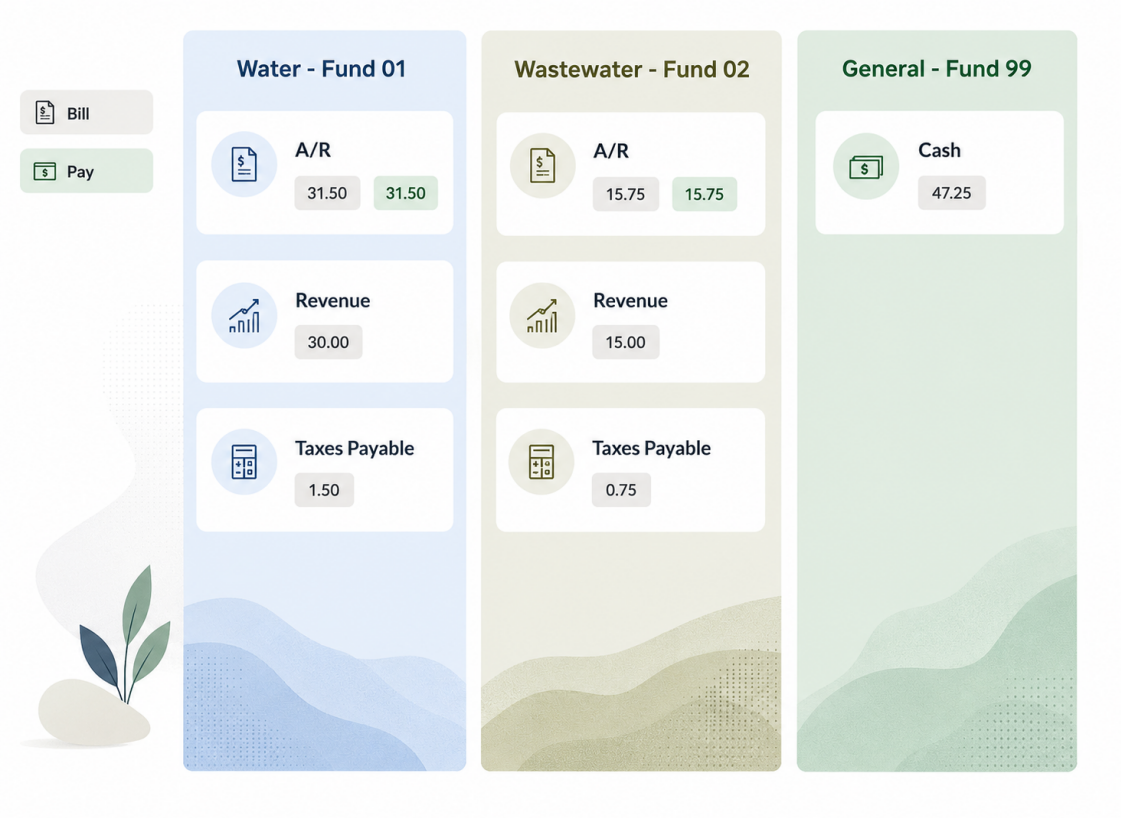


For each fund, the GL details of the bill will include a debit to the accounts receivable (A/R) account and credits to the revenue and taxes payable accounts. In organizational terms, each department is owed a portion of the overall bill by the customer, part of which is sales by the department and part of which is owed to the taxing authorities by the department. Each fund is balanced.

Note that the accounting could be identical under corporate accounting if each service is its own division with its own chart of accounts.

The following diagram illustrates the initial GL accounting that would occur when the payment arrives:

Copyright © 2026, Oracle and/or its affiliates.

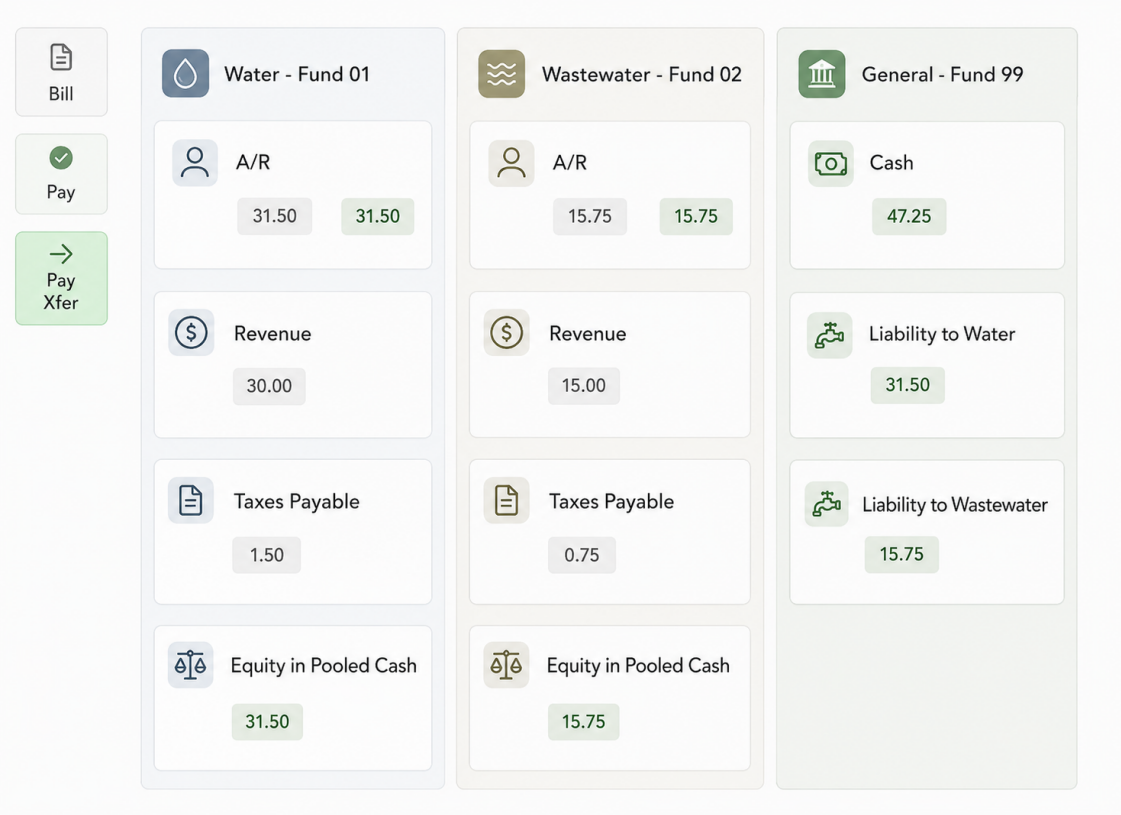


The utility's general cash account is debited, and the departmental funds' A/R accounts are credited. In other words, the cash is held by the utility as a whole but the receivables are reduced for the individual departments.

If the accounting were left in this state, the fund accounting principal - that each fund represents an independent entity with a self-balancing chart of accounts - would be violated. This violation is caused due to the fact that cash is recorded on the general fund, not the departmental funds, causing the general fund to have an excess debit and the departmental funds to have an excess credit.

From an organizational viewpoint, to make each department whole, the departments need to note what portion of the cash they own, and correspondingly, the utility needs to note what portion of the cash is owed to each department. The following diagram illustrates this point.

Copyright © 2026, Oracle and/or its affiliates.



To maintain a balance of debits and credits within each fund, the departmental funds have an "equity in pooled cash" (EPC) account and the general fund has a liability account for each departmental fund. In addition to debiting the general fund's cash account and crediting the departmental funds' A/R accounts, the departmental funds' EPC accounts are debited and the general funds liability accounts are credited.

And so, with the additional GL entries, all funds have matching debits and credits.

An Example Of A Bill Segment That References Multiple Funds

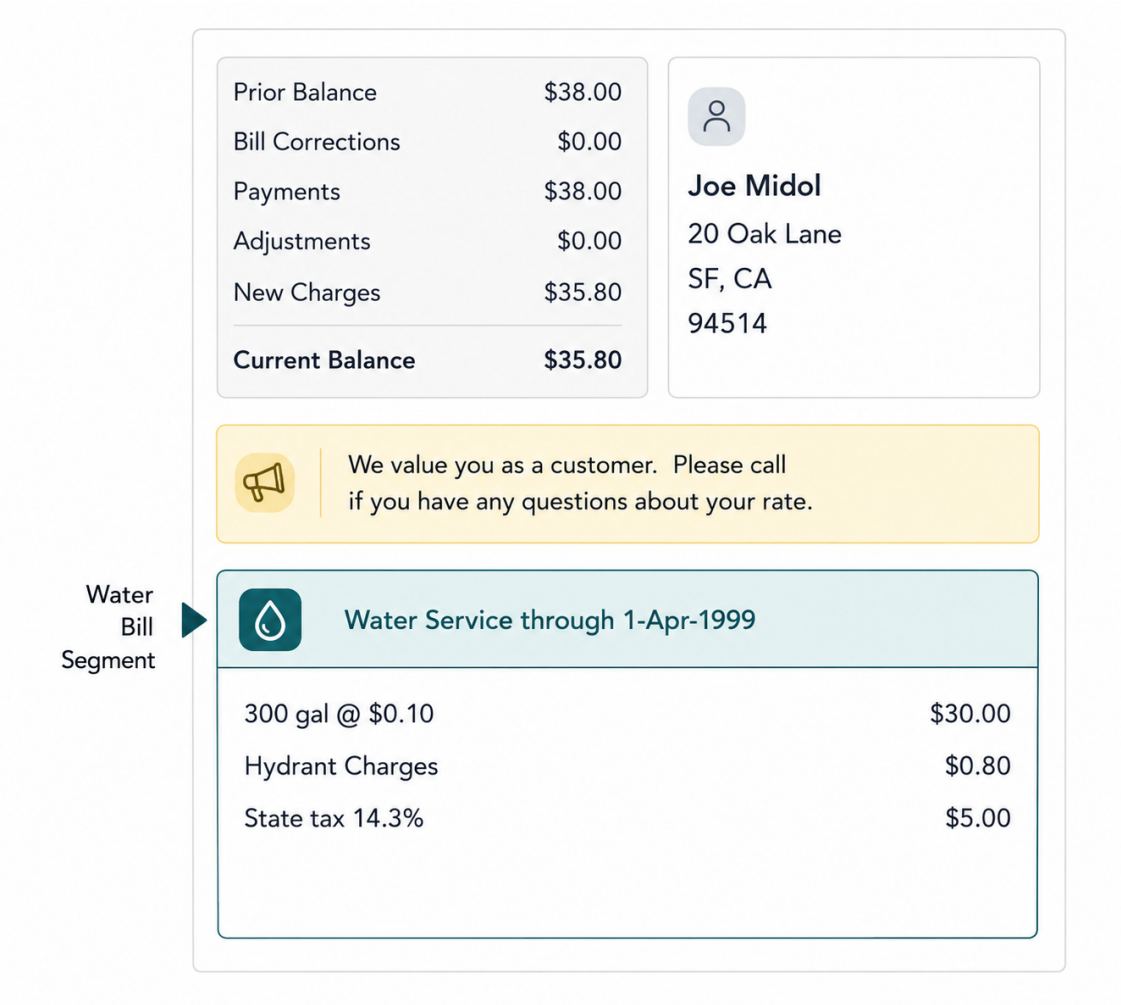
Consider a municipal utility that primarily supplies water service but is also responsible for maintaining the city's fire hydrants. The costs for fire hydrant maintenance are borne by the water customers and make up just a small portion of the overall bill. These costs are simply added to the water bill as a line item. The utility has two departments:

- Water service (fund 01)
- Hydrant maintenance (fund 39).

In addition, there is a general fund (fund 99).

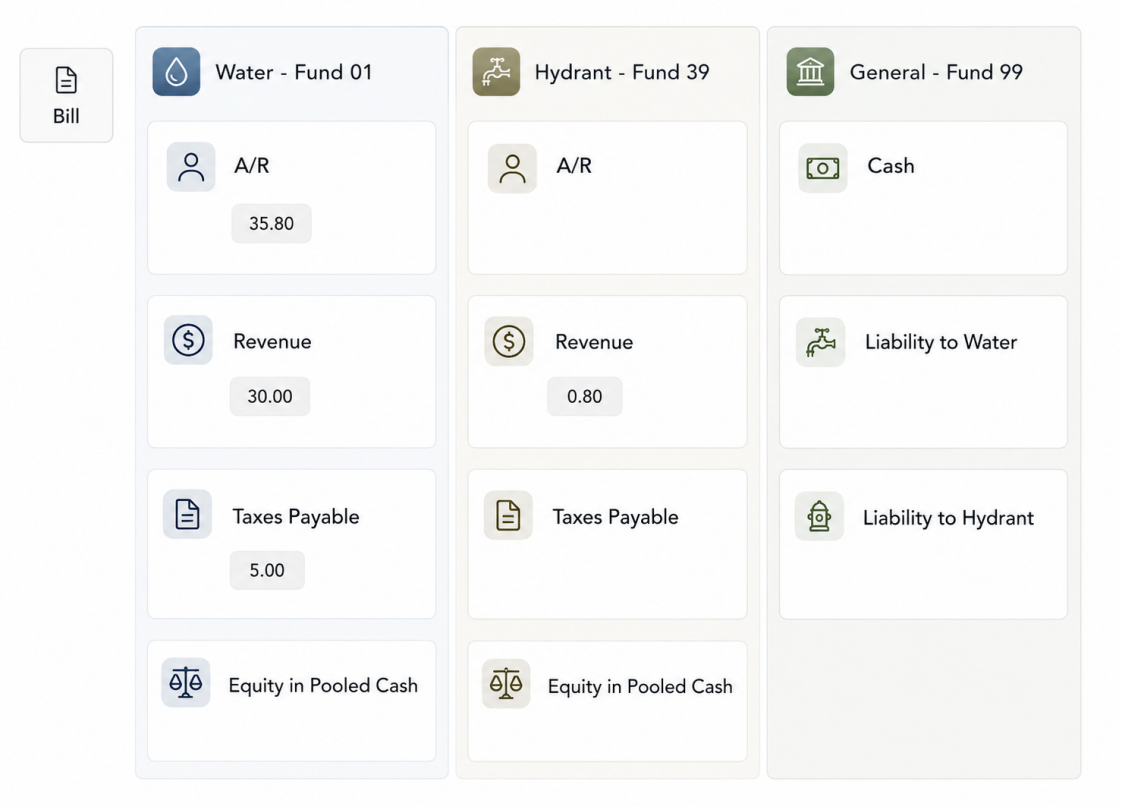
Assume the following bill is generated for water and hydrant services.

Copyright © 2026, Oracle and/or its affiliates.



The following diagram illustrates the initial GL entries for the bill:

Copyright © 2026, Oracle and/or its affiliates.

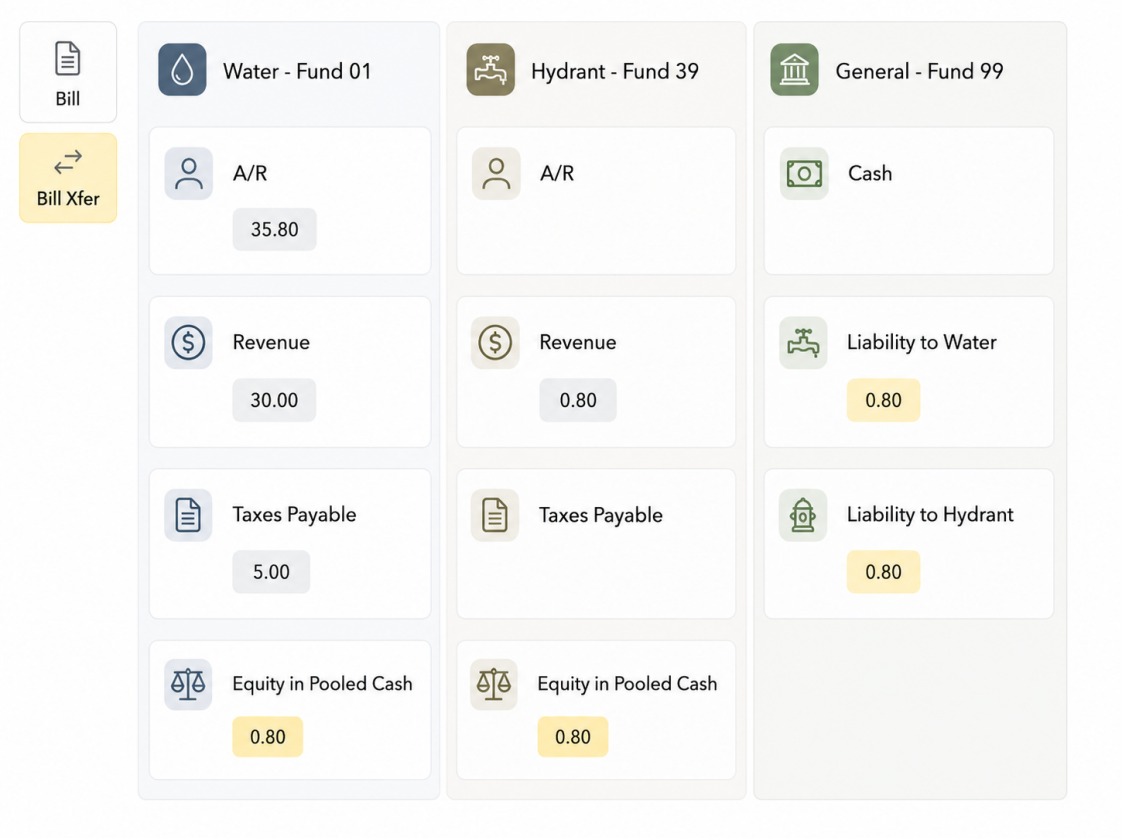


In accounting for the bill, the water fund's A/R is debited, the water and hydrant funds' revenue accounts are credited, and the water's taxes payable account is credited.

If left at this, the funds would be out of balance; the water fund would have an overall excess debit and the hydrant fund would have an equal excess credit. In organizational terms, the hydrant fund has recorded sales but that amount is recorded as being owed to the water department.

To balance each department, the water department accepts the responsibility for collecting the hydrant charges from the customer but immediately remunerates the charges to the hydrant fund.

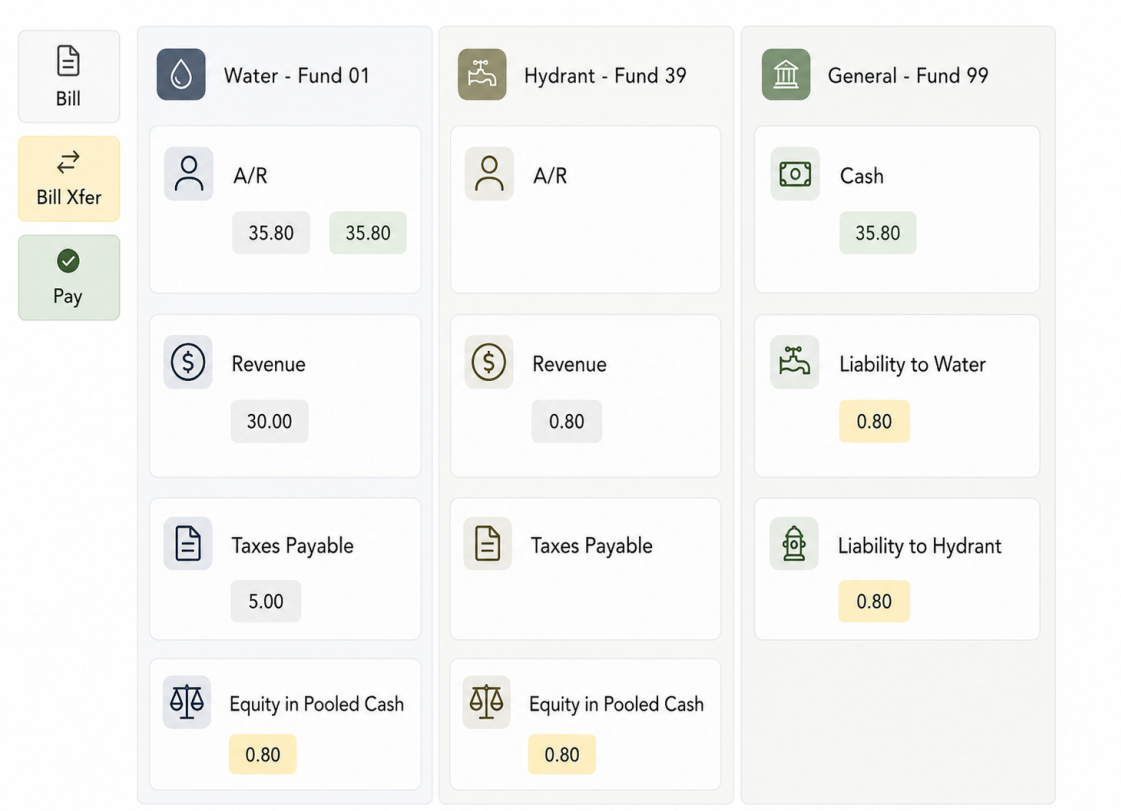
Copyright © 2026, Oracle and/or its affiliates.



This transfer is done using the general fund. The water fund's EPC account is credited and the liability to water is debited with the amount of the hydrant revenue. Also, the hydrant fund's EPC account is debited and the general fund's liability to hydrant account is credited by the hydrant revenue. In effect, the water department owes the hydrant charges to the general fund, and the general fund owes the hydrant charges to the hydrant fund.

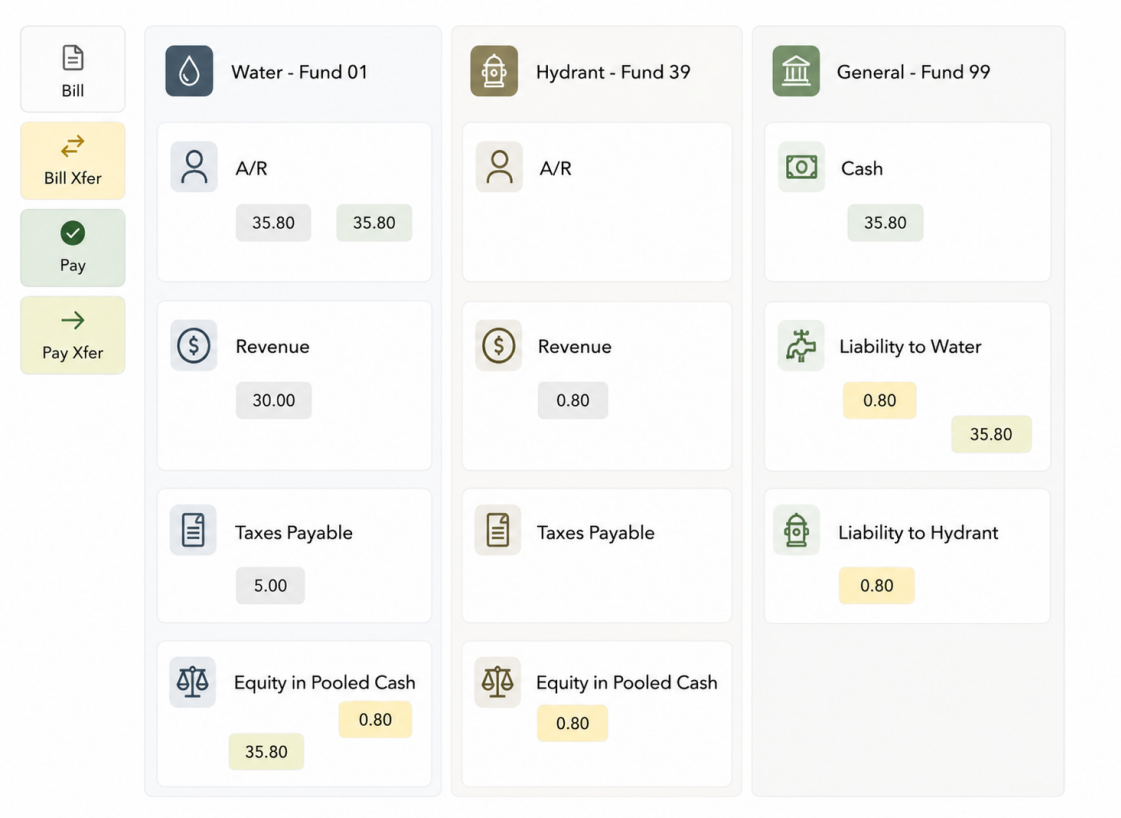
The following diagram illustrates the initial GL accounting that would occur when the payment arrives:

Copyright © 2026, Oracle and/or its affiliates.



When the payment arrives, the cash is debited to the general fund's cash account, and the water fund's A/R is relieved. Again, the funds would be unbalanced if left in this condition; the water fund would have an excess of credits and the general fund would have an excess of debits.

Copyright © 2026, Oracle and/or its affiliates.



To maintain each fund's balance of debits and credits, the general fund's liability to the water fund is credited by the amount of the department's share of the cash, and the water fund's EPC is debited. Note that the payment has no effect on hydrant fund's EPC and the general fund's liability to the hydrant fund. The hydrant department "received" its money from the water department when the bill was created.

And so, all funds have matching debits and credits.

Accounting Method Is Defined On The Installation Options

You must turn on a switch on the installation Record to enable fund accounting.

Fund Controls Fund-Balancing Entries

There are two levels of debit and credit balancing in fund accounting. There is the balancing required by double entry accounting: the total debits in the entire GL must equal the total credits. This is required regardless of whether fund or corporate accounting is used. The distribution codes for these entries come from varying sources, depending on the type of financial event.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Source Of GL Accounts On Financial Transactions for information on the sources of the distribution codes.

The second level of balancing is specific to fund accounting. Within each fund-not just across the GL-the total debits must equal the total credits. The original distribution code from the financial event has a fund specified. For example, a bill would cause a debit to a fund's A/R distribution code, and included in that A/R distribution code is the fund. It is the definition of the fund that specifies whether fund-balancing entries are required and provides the distribution codes for these entries.

For a departmental fund, the fund-balancing debit and credit would be specified. When a debit is applied to a departmental fund's GL account, an additional account (typically the general fund's liability to the departmental fund) is debited and an account (typically the departmental fund's EPC) is credited. When a credit is applied to a departmental fund's account, an additional account (typically the general fund's liability to the departmental fund) is credited and an account (typically the department's EPC) is debited.

For the general fund, no fund-balancing debits and credits are specified.

Building Fund-Balancing GL Details

Building the GL details for a financial event is a two-step process.

- First, the system generates the regular GL details for a financial transaction (FT). This is done regardless of whether corporate or fund accounting is used.
- Second, if fund accounting is activated (by turning on a switch on the installation Record), the system analyzes the distribution code on each GL detail associated with the FT. If a fund is specified on a distribution code, the system checks the definition of the fund. If fund-balancing entries are specified on the fund, two additional GL entries are added to the FT:
- An offsetting entry to the Equity in Pooled Cash account is created for the departmental fund (e.g., if the FT is debiting a given fund, an offsetting credit is created in the funds EPC account).
- Another entry to the departments Liability account is created for the general fund.

The result is a consolidated set of GL entries for the FT, incorporating the regular entries as well as the fund-balancing entries.

The topics in this section illustrate the generation of the GL details for the earlier examples.

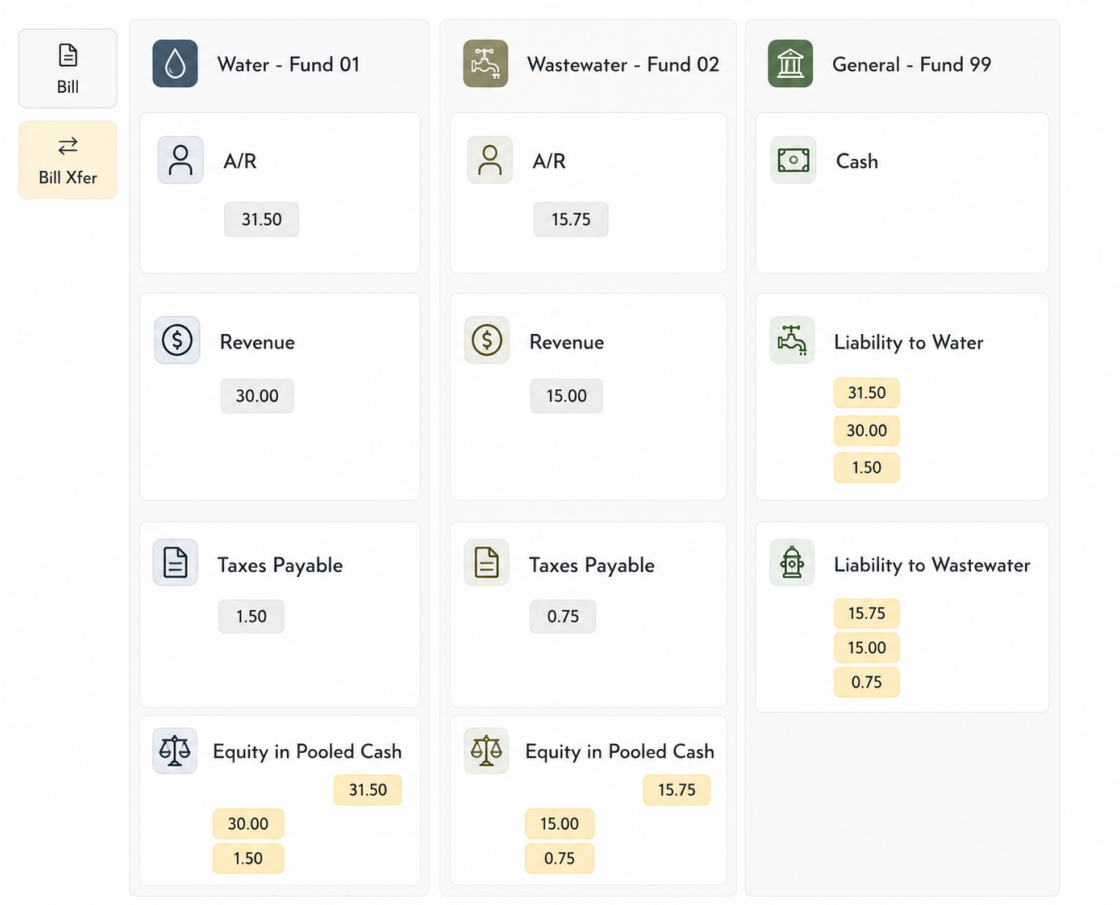
- FTs Whose GL Details All Reference The Same Fund Do Not Impact the General Fund or EPC Accounts
- An FT Whose GL Details Reference Multiple Funds

FTs Whose GL Details All Reference The Same Fund Do Not Impact the General Fund or EPC Accounts

Copyright © 2026, Oracle and/or its affiliates.

In Fund Accounting Example, where the bill's bill segments reference a single fund, the system creates a fund-balancing GL entry for each GL entry applied to a departmental fund:

- A debit to a departmental GL account triggers a debit to the general fund's liability-to-departmental-fund account and a credit to the departmental fund's equity-in-pending-cash account.
- A credit to a departmental GL account triggers a credit to the general fund's liability-to-departmental-fund account and a debit to the departmental fund's equity-in-cash account.



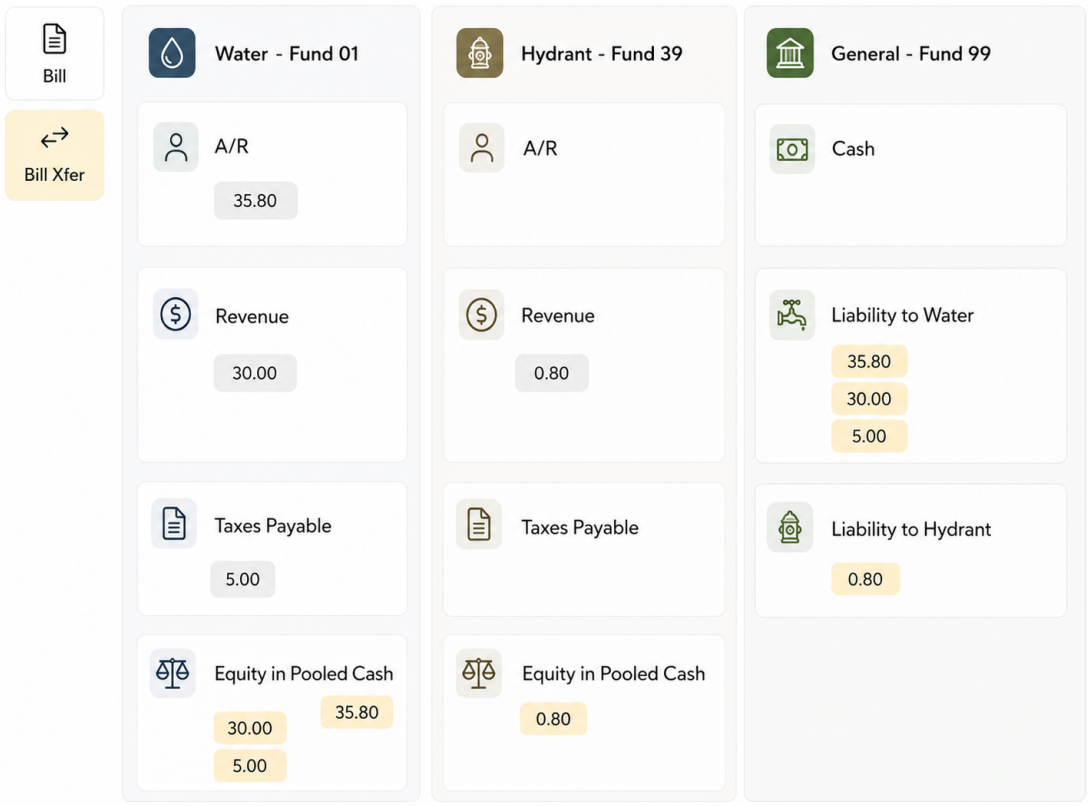
The net effect on the individual equity-in-cash and general fund's liability accounts is zero because the debits and credits net to zero for each GL account. In other words, the yellow boxes net to zero and therefore fund accounting does not impact the bill segment's financial transactions. Refer to Fund Accounting Example for the resulting consolidated GL entries.

An FT Whose GL Details Reference Multiple Funds

In An Example Of A Bill Segment That References Multiple Funds, where the bill's bill segments reference multiple funds (water and hydrant), the system also creates fund-balancing GL entries for the financial transaction:

- A debit to a departmental GL account triggers a debit to the general fund's liability-to-departmental-fund account and a credit to the departmental fund's equity-in-pending-cash account.
- A credit to a departmental GL account triggers a credit to the general fund's liability-to-departmental-fund account and a debit to the departmental fund's equity-in-cash account.

Copyright © 2026, Oracle and/or its affiliates.



The net effect of the bill on the GL is that the water fund's EPC has a credit of $0.80, the hydrant fund's EPC has a debit of $0.80, the general fund's liability to the water fund has a debit of $0.80, and the general fund's liability to the hydrant fund has a credit of $0.80. Note that, overall, the general fund's overall liability to the departmental funds nets to zero. Refer to An Example Of A Bill Segment That References Multiple Funds for the resulting consolidated GL entries.

Setting Up The System To Enable Fund Accounting

The following section provides an overview of how to enable fund accounting.

- Turn On Fund Accounting
- Defining Funds
- Distribution Codes Must Include Fund ID
- Update Your Funds With Their Respective Equity and Liability Distribution Codes

Turn On Fund Accounting

On the installation Record, indicate that fund accounting is Practiced.

Defining Funds

Copyright © 2026, Oracle and/or its affiliates.

A fund must be setup for each specific fund in your organization. Don't forget to also set up a fund for the general fund. Navigate using **Admin**, and then **Financial**, and then **Fund.**

Description of Page

Enter a **Fund** and a **Description** to identify the fund.

If this fund is used to balance other funds or to hold cash, indicate a **Fund Type** of General, otherwise indicate that it is Specific.

If the fund type is Specific, specify the **Equity Distribution Code** and **Liability Distribution Code**. These codes are used to balance financial transactions that span funds. The equity distribution code should belong to the same **Fund** as the one you are defining. The liability distribution code should belong to the general fund.

Distribution Codes Must Include Fund ID

All of your distribution codes must include their respective fund ID.

NOT_SUPPORTED

For more information, refer to Setting Up Distribution Codes.

Update Your Funds With Their Respective Equity and Liability Distribution Codes

After distribution codes have been setup, you must update your funds to indicate the equity and liability accounts used to balance inter-fund financial transactions.

Bill Taxation Threshold

Some implementations only apply taxes if the accumulated tax amount at the bill level exceeds some specified threshold amount.

- Taxation Threshold Examples
- Billing and Taxation Thresholds
- Setting Up The System For Bill Taxation Thresholds

Taxation Threshold Examples

The following examples show how taxation thresholds affect a customer's bill. In the examples a tax rate of 5% and a threshold amount of $21.30 is used.

- Example 1 - Account With Taxes Under Threshold
- Example 2 - Account With Taxes Above Threshold
- Example 3 - Account With Rounding Discrepancy

Copyright © 2026, Oracle and/or its affiliates.

Example 1 - Account With Taxes Under Threshold

This example shows the bill for an account where the accumulated tax amount is less than the threshold amount. Since the accumulated tax amount of $10.50 is less than the threshold amount, taxes are not applicable and the account's bill should be adjusted to exclude the tax amount of $10.50.

Bill Line	Amount
Standing Charge	10.00
2,000 units @ $0.10	200.00
Tax @ 5% on $210.00	10.50
Total	220.50
Adjusted Total	210.00

Example 2 - Account With Taxes Above Threshold

This example shows the bill for an account where the accumulated tax amount is greater than the threshold amount. Since the accumulated tax amount of $27.92 is greater than the threshold amount, taxes are applicable and the account should be billed for the total amount.

Bill Line	Amount
Standing Charge	10.00
4,000 units @ $0.10	400.00
Additional Charge	148.45
Tax @ 5% on $558.45	27.92
Total	586.37

Example 3 - Account With Rounding Discrepancy

This example shows the resulting bill segment calc lines for an account with 3 service agreements. Here the accumulated tax amount at the bill level is $21.29 with taxes calculated and rounded for each SA's bill segment. However, if the same taxes were calculated for each SA's bill segment and then accumulated and rounded at the bill level, the accumulated tax amount would be $21.30 implying that taxes are applicable and the account should be billed for the tax amount of $21.30 accounting for the discrepancy of $0.01.

Bill Segment	Bill Line	Amount	Pre-rounding Amount
Bill segment for SA 1	964.70 units @ $0.10	96.47	96.47000
	Tax @ 5% on $96.70	4.82	4.82350
	Bill Segment Total	101.29	
Bill segment for SA 2	2222.90 units @ $0.10	222.29	222.29000
	Tax @ 5% on $222.29	11.11	11.11450

Copyright © 2026, Oracle and/or its affiliates.

	Bill Segment Total	233.40	
Bill segment for SA 3	1072.40 units @ $0.10	107.24	107.24000
	Tax @ 5% on $107.24	5.36	5.36200
	Tax Discrepancy	0.01	
	Bill Segment Total	112.61	

> **Note**
>
> **Pre-rounding amount.** Rate application captures two calculated amounts on the resulting bill segment calc lines. The first is the calculated amount rounded to two decimal places, and the second is a raw calculated amount with a five decimal precision. The base package algorithm that calculates taxation thresholds uses both amounts to account for any rounding discrepancy; however, only the raw calculated amount is used to compare against the taxation threshold.

Billing and Taxation Thresholds

The following sections describe how taxation threshold rules are implemented when a bill is produced.

- Calculation of Taxation Thresholds
- Tax Amount Discrepancies

Calculation of Taxation Thresholds

When taxation thresholds are applied at the account's bill level, it means that the system must calculate taxes for each of the account's service agreements, then sum these tax amounts and apply any applicable rounding rules. This accumulated tax amount is compared to a threshold amount and if the accumulated tax amount is less than the threshold amount, then taxes should not be applied to the customer's bill. To ensure the accumulated tax amount is accurate, all service agreements for an account whose tax amounts should be taken into consideration when comparing to the specified threshold should be billed together and the threshold comparison should take place after the bill segments have been generated. In addition, all bill segments whose tax amounts should be taken into consideration when comparing to the specified threshold must be for the same bill period.

> **Note**
>
> **Bill segments from different periods may not appear on the same bill.** If a bill segment is canceled, all bill segments associated with that bill must be canceled and re-billed together. You cannot re-bill the bill segment on a bill for a different period.

An algorithm type is supplied with the base package that checks for taxation thresholds at bill completion time. The base bill completion algorithm type accumulates identified tax calc line

Copyright © 2026, Oracle and/or its affiliates.

amounts (accomplished using a bill segment calc line characteristic), and compares this to a specified threshold amount to determine if taxes apply for the account. If taxes do not apply, it sets a bill characteristic indicating this and regenerates the bill segment. Refer to the C1-CPBC-TAXT algorithm type for more information on how this type of algorithm operates.

> **Note**
>
> **Calculated adjustments are included in the evaluation against taxation thresholds.** Adjustments that use a rate to calculate the adjustment amount may be included in the taxation threshold evaluation if their rate's components are set up to do so. The base algorithm C1-CPBC-TAXT looks at adjustments that are about to be swept onto the bill and, if applicable, includes them in the calculation.

The calculation rule(s) that calculate taxes make use of calculation rule eligibility criteria to ensure that taxes are only computed if the bill does not have the characteristic indicating that taxes are not applicable. Note that during the initial generation of each bill segment, the characteristic will not exist and taxes will be applied. This means that if you look at a bill before it is complete, taxes may not be accurately reflected.

Tax Amount Discrepancies

Since tax calculation rules are calculated and rounded at the bill segment level, it's possible that rounding discrepancies may occur if rounding of these tax amounts occurs at the bill level instead as illustrated in the example above. To account for this, the system uses both the two decimal precision and the five decimal precision calculated amounts that rate application captures on bill segment calc lines; however, only the raw calculated amount is used to compare against the taxation threshold. If there is a discrepancy in the tax amount (as shown in example 3 above), the system captures this amount as an entry in one of the bill segment's SQ collections prior to regenerating the bill segments. This is depicted in example 3 above where the tax discrepancy SQ resulted in an additional bill segment calc line on one of the bill segments for the rounding amount of 0.01. Note that a calculation rule is configured to bill for this discrepancy amount SQI.

> **Note**
>
> **Pre-rounding amount.** Rate application captures two calculated amounts on the resulting bill segment calc lines. The first is the calculated amount rounded to two decimal places, and the second is a raw calculated amount with a five decimal precision. The base package algorithm that calculates taxation thresholds uses both amounts to account for any rounding discrepancy; however, only the raw calculated amount is used to compare against the taxation threshold.

Setting Up The System For Bill Taxation Thresholds

This section provides an overview of how to configure the system to calculate taxes at the account's bill level.

- Installation Option

Copyright © 2026, Oracle and/or its affiliates.

- Adjustment Types
- Bill Factors
- Characteristic Type
- Service Quantity Identifiers
- Algorithms
- Calculation Rules For Bill Taxation Thresholds

Installation Option

Select Freeze at Bill Completion as the bill segment freeze option. The system compares tax amounts calculated to a specified threshold amount and based on this determines if taxes should apply at bill completion time. If taxes should not be applied for the account, the system has to regenerate the bill segments as necessary.

Adjustment Types

Select Freeze at Bill Completion as the adjustment freeze option. The system compares tax amounts calculated to a specified threshold amount and based on this determines if taxes should apply at bill completion time. If taxes should not be applied for the account, the system has to regenerate the adjustments as necessary.

Bill Factors

Tax Threshold. Set up a separate bill factor for each distinct tax threshold amount.

Characteristic Type

Set up a characteristic type and value to identify the tax calculation rules and bill segment calculation lines that the system will use to compare to the specified threshold amount. Specify this characteristic type and value as parameters to the algorithms that apply the taxation threshold. This characteristic type and value must also be specified on each of your tax calculation rules that should be included in the threshold comparison.

Set up a characteristic type and value to use for the bill characteristic that indicates when bill level taxes apply. Specify this characteristic type and value as parameters to the algorithms that apply the taxation threshold.

Service Quantity Identifiers

Optional service quantity identifiers may be configured to capture the following:

- Tax Amount. The system compares the accumulated bill's tax amount to the specified threshold amount to determine if taxes are applicable. If taxes should not be applied for the account, the system regenerates the bill segments as necessary. If you wish to capture the

Copyright © 2026, Oracle and/or its affiliates.

tax amount computed for informational purposes, then a tax amount SQI should be set up and specified as a parameter on the algorithms that apply the taxation threshold.

- Tax Discrepancy Amount. Since tax calculation rules are calculated and rounded at the bill segment level, it's possible that rounding discrepancies may occur if rounding of these tax amounts should take place at the bill level. If you wish to account for these rounding discrepancies, then a tax discrepancy amount SQI should be set up and specified as a parameter on the algorithms that apply the taxation threshold. Your rates should also be configured to cater for this rounding discrepancy. The system adds an entry for the discrepancy amount to one of the bill segment's SQ collections prior to regenerating the bill segments.

Algorithms

Apply Taxation Threshold. Define an algorithm for each distinct tax threshold amount. Plug the algorithm(s) in on the appropriate customer class for the Pre Bill Completion system event. Refer to the C1-CPBC-TAXT algorithm type for more information about the base package algorithm.

Adjustment Generation - Apply Rate. Define an algorithm for each rate to be used by calculated adjustment types that are to be included in the tax threshold evaluation. Plug the algorithm(s) in on the appropriate adjustment type for the Generate Adjustment system event. Refer to the ADJG-RT algorithm type for more information about the base package algorithm.

Calculation Rules For Bill Taxation Thresholds

Your calculation rules that bill for taxes require the following:

- A characteristic that identifies them as calculation rules to include in taxation threshold comparisons
- Calculation rule eligibility criteria to ensure that taxes are not calculated if the taxation threshold algorithm dictates this

Only one eligibility group on the calculation rule is required. It would look as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Tax applies if total accumulated tax amount at the account's bill level exceeds the threshold amount	Apply calculation rule	Skip calculation rule

The following criteria will be required for this group:

Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data

Copyright © 2026, Oracle and/or its affiliates.

10	Bill characteristic: Characteristic type = Tax Not Applicable indicator	= YES	Group is false	Check next condition	Group is true
20	Characteristic Collection: Characteristic type = Tax Not Applicable	= YES	Group is false	Group is true	Group is true

> **Note**
>
> The second criterion included above is used to evaluate the applicability of the tax on calculated adjustments. Since these adjustments are not yet linked to the bill, the first criterion cannot be used to evaluate the applicability of the tax threshold. The pre-bill completion algorithm C1-CPBC-TAXT instead adds the characteristic to the characteristic collection for use by the rate application when evaluating the adjustment's rate's eligibility rules. Refer to the ADJG-RT algorithm type for more information about the base package algorithm.

> **NOT_SUPPORTED**
>
> For more information, refer to Designing Calculation Groups and Rules.

An additional calculation rule is required to bill for the tax rounding discrepancy as follows

- Calculation Rule type = Service Quantity
- UOM/TOU/SQI = the identifier of the service quantity containing the tax discrepancy amount
- Value Type = Unit Rate
- Value Source = Value
- Value = 1

Other Financial Transaction Topics

Various topics about financial transactions are discussed in this section.

- The Source Of GL Accounts On Financial Transactions

The Source Of GL Accounts On Financial Transactions

Copyright © 2026, Oracle and/or its affiliates.

The following table lists the major financial events, their standard accounting, and the source of distribution codes used to derive the GL accounts sent to your general ledger.

Financial event	GL Accounting	Source Of Distribution Code
Create a normal utility bill segment. Bill Segment FT Algorithm is Payoff Amt = Bill Amt / Current Amt = Amt Due	Debit: A/R	SA Type
	Credit: Revenue / Taxes Payable	Calculation Rule
Create a bill for company usage. Bill Segment FT Algorithm is Payoff Amt = 0 / Current Amt = 0	Debit: Company Usage Expense	SA Type
	Credit: Revenue / Taxes Payable	Calculation Rule
Create a bill for charity. Bill Segment FT Algorithm is Payoff Amt=0 / Current Amt = Bill Amt	N/A - charity bills have no effect in the GL	N/A
	N/A	N/A
Create a payment segment for a normal utility service agreement	Debit: Cash	Bank Account on the Tender Source of the Tender Control for the Payment Segment's Tender.
	Credit: A/R	SA Type
Create a payment segment for a charitable contribution service agreement	Debit: Cash	Bank Account on the Tender Source of the Tender Control for the Payment Segment's Tender.
	Credit: Charity Payable	SA Type
Create a payment segment for auto-pay at bill completion time	Debit: Cash	Bank Account on the Tender Source on the Auto-pay Route Type of the Auto-pay Source.
	Credit: A/R	SA Type
Canceling a payment	Debit: A/R	SA Type
	Credit: Cash	Bank Account specified by the user on the cancel tender page. Note that this defaults to the original tender's bank account.
Create an adjustment to levy a charge	Debit: A/R	SA Type
	Credit: Revenue	Adjustment Type

The bottom line is as follows:

- If a bill segment has a financial effect, the distribution code to debit comes from the distribution code on the SA Type, the distribution code to credit comes from the calculation rule(s) used to calculate the bill segment.

Copyright © 2026, Oracle and/or its affiliates.

- Payment segments always have a financial effect; the distribution code to debit comes from the bank account on the tender source of the tender control of the tender, the distribution code to credit comes from the SA type.
- If an adjustment has a financial effect, the distribution code to debit and credit comes from the SA type and adjustment type. If the adjustment is positive (i.e., the customer owes your organization more money), the distribution code to debit comes from the SA type; the distribution code to credit comes from the adjustment type. Vice versa if the adjustment is negative.

Defining Customer Options

The definition of a customer is someone (or something) with financial obligations with your company. These obligations ensue because the customer has agreed to purchase goods or services at an agreed price.

You may be surprised to learn that there is no "customer" record in the system. Rather, the system subdivides customer information into the following records:

- **Person.** The person record holds demographic information about your customers and every other individual or business with which your company has contact. For example, in addition to customers, person records also exist for landlords, contractors, accountants at corporate customers, guarantors of customers, energy distributors, collection agencies, etc.
- **Account.** Accounts are the entities for which bills are produced and therefore you must create at least one account for every person who has financial obligations with your company. The account record contains information that controls when the bills are created and how the bills are formatted.
- **Service Agreement.** Think of a service agreement as a contract between your company and the customer. The service agreement contains the terms and conditions controlling how the bill details are created. Every account will have at least one service agreement (otherwise, nothing will appear on the account's bills).

Before you can define persons, accounts, and service agreements, you must set up the control tables defined in this section.

NOT_SUPPORTED

For more information about how persons, accounts and premises are used by your customer service reps, refer to Understanding The "V".

Note

The tables in this section are only some of many tables that must be set up before you can bill your customers for the service(s) they consume. In this section, we limit the discussion to those tables that control basic demographic and financial information. In later sections, we describe the tables that control other billing-related functions like bill creation algorithms, meter reading and rates. It is only after all of these tables are set up that you will be able to generate bills and record payments.

Copyright © 2026, Oracle and/or its affiliates.

- Customer Overview
- Setting Up Person Options
- Setting Up Statement Construct Options
- Setting Up Account Options
- Setting Up Customer Contact Options
- Setting Up Notification Preference Options
- Setting Up Service Agreement Options
- Setting Up Order Options
- Setting Up Program Management Options
- Setting Up Letter Templates
- Setting Up Customer Service Request Types
- Defining Customer 360 Options
- Configuring the Customer Intelligence Hub
- Setting Up Customer Profile Options

Customer Overview

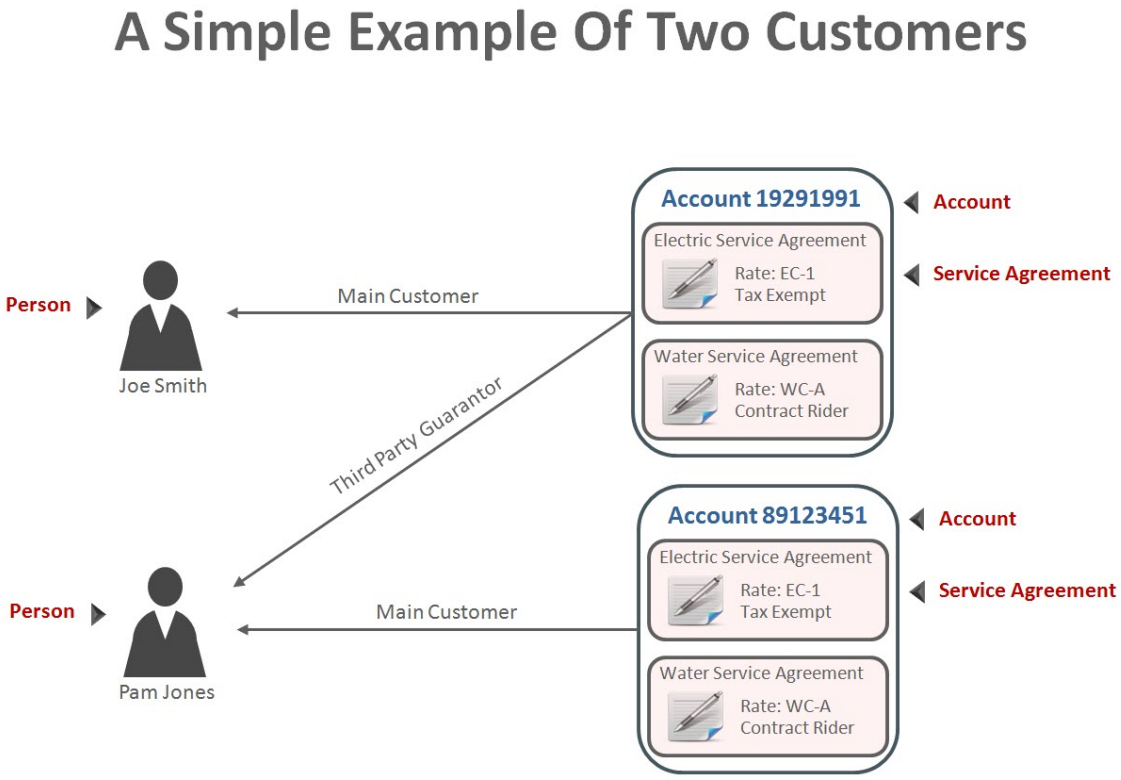
This section describes how the person, account, and service agreement records are used to record your customers' demographic and billing options.

- A Simple Example Of Two Customers
- Persons
- Accounts
- Service Agreements

A Simple Example Of Two Customers

The following picture illustrates two customers: Joe Smith and Pam Jones. Joe is the "main customer" on the first account. Pam is the "main customer" on the second account. Pam is also the "third party guarantor" on Joe's account.

Copyright © 2026, Oracle and/or its affiliates.



Persons

Person records hold demographic information about the individuals and businesses with whom your organization communicates. Demographic information includes phone number(s), names and aliases, identification numbers, life support equipment needs, employment information, etc.

In the above example, 2 person records would be needed; one for Pam Jones and another for Joe Smith.

A new person is added when you first have contact with a person; the person does not have to be a customer before it is added. So, for example, if your company is starting a new marketing campaign, you can add information about potential customers the moment they are identified.

> **Note**
>
> **Businesses are persons too.** In addition to humans, you use person records to maintain basic information about the businesses with which your organization has contact.

> **NOT_SUPPORTED**
>
> For a description of the control tables that must be set up before you can define a person, refer to Setting Up Person Options .

Copyright © 2026, Oracle and/or its affiliates.

Accounts

An account is analogous to an account at a bank:

- A person or business with no financial dealings with a bank will have no account (but the bank may choose to keep demographic information about the person as part of their marketing efforts). The exact analogy exists in this system.
- Individuals with financial dealings with a bank will have one or more accounts. The number of accounts is up to the customer. The exact analogy exists in this system.

A simple way to determine the number of accounts a customer will have is to ask "how many bills do they want each period?" because a customer receives one bill for each account. For example:

- A residential customer who also owns a small business may choose to receive two bills each month; one for the residence, the other for the business. This way, the charges for their business would be segregated from their personal charges. This customer would have two accounts.
- A conglomerate that owns several factories may want their transportation gas charges to appear on a single bill rather than have a separate bill for each factory. This customer would have a single account.

- Account ID Is Non-Intelligent
- Account / Person Cross-Reference
- When Is An Account Created?
- When Is An Account Expired?

Account ID Is Non-Intelligent

The unique number of an account is referred to as the "account ID". You are probably very comfortable with this concept. You may, however, have difficulty dealing with the fact that the account id in this system has no intelligence built into it (e.g., many systems include the bill cycle and geographic location in the account id). In this system, the account ID is a random, system-assigned value.

Because the account ID contains no meaning, it can remain with a customer for life, regardless of where they live, when they are billed, the type of service they receive, etc. This is important because it means that all of the financial history linked to the account remains with the customer for life.

> **Note**
>
> The non-intelligence of the account ID is also important from the perspective of the parallel processing that takes place when the system creates bills. Because the collection of accounts to be billed in any given bill cycle will be randomly distributed through the number spectrum, the system can distribute account number ranges to parallel threads and each thread will process roughly the same number of accounts.

Copyright © 2026, Oracle and/or its affiliates.

Account / Person Cross-Reference

A person may be linked to zero or more accounts. A person won't be linked to an account when they have no financial relationship with your organization. A person will be linked to multiple accounts when they have financial relationships with more than one account.

An account must reference at least one person (i.e., the main customer), but may reference an unlimited number of individuals. Multiple persons are linked to an account when several parties have some type of financial relationship with the account (e.g., third party guarantors, account contact, bill copy recipients, etc.).

When Is An Account Created?

A person can exist without an account until such time as the person formally requests the commencement of service. The moment the customer requests service, an account must be created (and the person must be linked to the account).

When Is An Account Expired?

Accounts never expire. Once a customer has an account, the account remains in the system forever. Linked to the account are service agreements that define the price and conditions of a service supplied to the customer. When an account has active service agreements, the system produces bills for it. If the account doesn't have active service agreements, the system will not produce a bill for it. You can think of an account without active service agreements as being "dormant", waiting for the day when the customer again starts service. If the customer never restarts, the account (along with its financial history) remains dormant forever.

Service Agreements

A service agreement is a contract (either formal or implied) between your organization and a customer. Every service agreement contains the price and conditions of a service supplied to a customer.

A service agreement is linked to an account. There is no limit to the number of service agreements that may be linked to an account.

- When Is A Service Agreement Created?
- Financial Transactions Are Linked To Service Agreements
- When Is A Service Agreement Expired?

When Is A Service Agreement Created?

A service agreement is created when the customer requests service (not when service commences). Typically, service agreements are created in the pending state and field activities are generated to connect service. When the field activities are complete, the service

Copyright © 2026, Oracle and/or its affiliates.

agreement becomes active and the billing process starts generating bill segments for the service agreement.

NOT_SUPPORTED

For more information about starting service, refer to The Big Picture Of Starting Service. For more information about bill segments, refer to Bill Details .

Financial Transactions Are Linked To Service Agreements

NOT_SUPPORTED

For more information about how financial transactions are linked to service agreements, refer to The Financial Big Picture .

When Is A Service Agreement Expired?

A service agreement is expired when the customer requests service be stopped. At that time, the service agreement is transitioned to the pending stop state and field activities are generated to stop service (these activities might involve simply reading the meter or they could involve disconnecting or removing the meter). When the field activities are complete, the system transitions the service agreement to the stopped state and the billing process generates a final bill for the service agreement. When the customer pays the final bill, the system transitions the service agreement to the closed state

NOT_SUPPORTED

For more information about stopping service, refer to The Big Picture Of Stopping Service.

Setting Up Person Options

This section describes tables that must be set up before you can define persons.

- Person Identifiers
- Setting Up Contact Routings
- Setting Up Person Contact Status
- Setting Up Person Contact Type Algorithms
- Setting Up Person Contact Types
- Choosing to Use Person Contact or Person Phone

Copyright © 2026, Oracle and/or its affiliates.

- Person Contact Status Can Be Controlled By A Process
- Maintaining Person Contact via Person Phone

Person Identifiers

This section describes the setup and use of person identifiers.

- Setting Up Person Identifier Types
- Understanding Person Identifiers

Setting Up Person Identifier Types

When you set up a person, you may define the various types of identification associated with the person, e.g., their driver's license number, their tax identity, etc. Every piece of identification associated with a person has an identification type.

> **Note**
>
> **How are person identifiers used?** The reason why identifiers are defined on a person is so that users can look for a customer using one of their person identifiers (see Control Central - Search Facilities for more information). In addition, person identifiers help prevent duplicate persons from being added to the database. The base product provides an algorithm type C1-PERSONVAL that can issue a warning or error to a user before they add a new person or update an existing person when another person exists with the same identifier. This is a BO validation algorithm on the person MO's identifying BO.
>
> **Person identifier types are optional.** An installation option controls whether at least one identifier type is required on every person.

You can access the portal from **Admin**, and then **Customer**, and then **Person Identifier Type**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Understanding Person Identifiers

Person Identifiers have many uses in the system. Their main use is to help uniquely identify persons and businesses. Another use is to store other non-uniquely identifying attributes about a person that are better suited to identifiers than they are to person contacts or characteristic. An example is a customer provided password or PIN that is verified before account details are provided.

The installation record defines if and when at least one person identifier is required on persons and businesses. Identifiers can be required when the record is created (always), when linked to an account, when linked to an account as financially responsible, or when linked to an account as the main customer. The system enforces these rules when a person is added or

Copyright © 2026, Oracle and/or its affiliates.

maintained or when persons linked to accounts are maintained. The installation record also defines the default identifier types for person and business records.

When a person identifier is required, the person identifier collection requires that one and only one record is marked as the primary identifier. Identifier types indicate if they can serve as the primary ID and also if they are validated for duplicates. Typically only ID types that should have unique values are configured to validate or duplicates.

Identifier types can be setup to indicate a format such as prefix, delimiter, and value. For example, an identifier for passport can indicate that the country code is the prefix followed by a dash then the passport number. A validation algorithm is provided to ensure all entered data conforms to this format.

Most organizations prefer that the primary identifier is uniquely identifying or a high-quality ID. However, it is not always possible to obtain such an identifier from all customers. The following are two common ways this is handled:

- An identifier type, such as a PIN, is configured and allowed to be the primary identifier, even if it considered a lesser quality or non-uniquely identifying identifier. These types should not be configured to validate for duplicates as their values are not uniquely identifying.
- The desired identifier is used, but a value such as all zeros or all nines is entered, indicating that the customer did not provide the value. In this scenario, the system cannot distinguish between an entry with a quality value and an entry with all zeros since the same ID type is used.

> **Note**
>
> Note: the customer service request-based start service process has configuration that allows your organization to differentiate between identifier types that can be primary and those that represent a high-quality ID. Refer to Verifying Identity, Checking Credit and Requiring Deposits for more information on how the start service process uses person identifiers.

Setting Up Contact Routings

Base values for Contact Routing are supplied. You can add your specific contact routings using the **Contact Routing** extendable lookup. You can add more contact routings, but the ones supplied with the system cannot be edited or deleted.

> **Note**
>
> Person contacts that reference the CSS routing type cannot be added by a user, and are not permitted to be edited or deleted. As well, records with this contact routing are only visible in the Person - Main page and are not displayed in the Account Portal or other places in the system.

Refer to Defining Extendable Lookups and for more information on lookups.

If your implementation is using notification preferences, you may need to set up delivery types before setting up contact routings. Refer to Setting Up Delivery Types for more information.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Person Contact Status

Person contacts can have a status; for example, the status of an email contact could be pending, verified, or bounced. The product is delivered with statuses that are used in conjunction with opt-in functionality. When new person contacts are added, the initial status can be defaulted from the person contact type. See The Person Contact Status Can Be Controlled By A Process, *System Person Contact Statuses* and *Opt-in Process* sections for more information.

Your implementation may choose to introduce other logic to update the status. See The Person Contact Status Can Be Controlled By A Process for more information. You can add your specific person contact statuses by updating the value for the CND_VERIF_STATUS_FLG lookup field.

When new person contacts are added, the initial status can be defaulted from the person contact type. Refer to the Person Contact zone's help text for more information.

Setting Up Person Contact Type Algorithms

Format Validation

Person contact information can be subject to format validation such as ensuring a phone number is entered in a particular format.

The base product provides the algorithms C1-VALEMFMT (Validate Email Format), C1-VALPHFMT (Validate Phone Format), and C1-VALANYFMT (Any Format Valid), which your implementation may use (and if so, no further configuration is needed). If your implementation's business rules require additional format validation algorithms, you may introduce your own.

> **Note**
>
> These algorithms confirm that a specific format exists and do not reformat or apply formatting to the corresponding field; for example, the *C1-VALEMFMT* (Validate Email Format) algorithm confirms that the email address contains required features (such as an @ symbol and domain). It will not, however, insert these features.

If your implementation's business rules require additional format validation algorithms, you may introduce your own.

Setting Up Person Contact Types

Person contact types define the format for entering person contacts and information about the person contact.

To set up contact types, select **Admin**, and then **Customer**, and then **Person Contact Type**, and then **Add.**

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Copyright © 2026, Oracle and/or its affiliates.

Choosing to Use Person Contact or Person Phone

Person contact is an updated and more configurable alternative to person phone and email. In this release, a feature is introduced that some system processes use to determine if person contact or person phone and email is being used. Person phone and email are referred to as legacy behavior. Legacy behavior is enabled when the Legacy Person Phone and Email **Option Type** on the Customer Information Options is set to Y. When set to N or not set legacy behavior is not enabled.

If your implementation uses functionality that has not been updated to use the feature, such as campaigns and order, you may wish to enable legacy behavior and continue to use person phone and email. It is important to note that Person Contacts are required for notifications.

The following system processes have been updated to alter functionality based on whether legacy behavior is enabled or not.

System Process	Behavior When Feature is Not Set	Behavior When Feature is Set
Person	The Person Phone Grid and the Person Email field are not shown	The Person Phone Grid and the Person Email field are shown. Person Contact is always shown.
Person	The functionality to maintain person contacts from person phone is not triggered even when a person contact type is associated with a phone type.	The functionality to maintain person contacts from person phone is enabled.
Account/ Person	When an email or fax bill or quote route type is chosen, then information displayed is related to person contacts.	When an email or fax bill or quote route type is chosen, then information displayed is related to person phone or email.
Start/Stop	The Person Contact collection is shown and can be edited.	The Person Phone collection and Person Email are shown and can be edited.
Case	The Preferred Contact Methods associated with legacy person phone and email are not shown.	The Preferred Contact Methods associated with legacy person phone and email are available. Person Contact is always available.
Customer Contact	The Preferred Contact Methods associated with legacy person phone and email are not shown.	The Preferred Contact Methods associated with legacy person phone and email are available. Person Contact is always available.

Person Contact Status Can Be Controlled By A Process

Overview

Person contact types can be configured to allow a status. For some person contact types, your implementation may want to control the status through a process instead of allowing the status to be set manually. An example of this is using person contact status to capture opting in for receiving text messages, which is required in some jurisdictions. Opt-in is seeking permission from a customer to use a person contact for a purpose such as sending a customer text messages.

Opt-in works in conjunction with the functionality to enable opt-in for a delivery type. Each can function independently, but together they offer a complete solution. Since they can function apart from each other, each is described independently. See Enabling Opt-in for a Delivery Type for more detail on this logic.

Copyright © 2026, Oracle and/or its affiliates.

To control a person contact status with a process, the person contact type must first be configured to allow status. The person contact type must then be set up in the opt-in section of the **Notification Preferences** master configuration. By defining a Status Script, the person contact status can only be changed by the logic contained in or initiated from the script. The product is delivered with a script that creates an opt-in service task. This is designed specifically to support opting in to receiving notifications via text messages (SMS).

Opt-In Process

To use the delivered functionality, configure a new service task type for opt-in. This task type should reference the Notification Preference Opt-In Task Type Business Object (C1-NotifPrefOptInTaskType). The product is delivered with a service task business object The service task is responsible for initiating an opt-in request, updating the person contact status, and logging the user response. An inbound web service receives responses and transitions the service task. These components are designed to integrate with Notification Center. When a customer is sent an opt-in request, they are asked to confirm. The content of the message explains the commands to the user. Examples are CONFIRM or STOP. Notification Center is configured to interpret the specific commands and map them to those expected by the inbound service, which transitions the service task appropriately.

Aside from the initial opt-in request, users can stop or unsubscribe from notifications at any time. The delivered solution supports two modes for this functionality: unsubscribing from all notifications for a particular person contact or unsubscribing from a specific notification type for a particular person contact. This is controlled through configuration, and relies on the external system such as Notification Center to determine the notification type.

The **Notification Preferences** master configuration help text contains detailed information about the configuration required to control person contact status and setting up opt-in processes.

It is important to note that while the delivered process to control person contact status was designed to work in conjunction with the opt-in process, your implementation can design a custom process that can create a service task or do something completely different. Also, it can be independent of opt-in and notifications. For example, the process can be used to verify that an email address is valid by sending the customer a verification request.

There is another concept, which is enabling opt-in for a delivery type. This is closely related to the functionality described here, but has different implications in the system. See Enabling Opt-in for a Delivery Type for more information.

There is a limitation worth noting. A person contact type may be set up to allow multiple delivery types, however the delivered solution only allows a single delivery type to be associated with the process used to automate person contact status. Without this limitation, different processes would be responsible for updating a single person contact status and those updates could be conflicting.

The System Person Contact Statuses

The opt-in solution delivered with the product utilizes the three delivered status values. They are: **Pending**, **Approved**, and **Rejected**. Your implementation can add your own person contact statuses, but they will not work with the solution delivered with the product. See Setting Up Person Contact Status.

How the System Automates Person Contact Status

There are many ways that the process that controls person contact status can be invoked. For all of these, the person contact type must be set up in the **Notification Preferences** master configuration to have its status controlled by a process and, unless noted otherwise, the person contact is not already ***Approved***.

Copyright © 2026, Oracle and/or its affiliates.

- The primary way to initiate the process to control person contact status is when a person contact is used to add a contact preference and the contact preference's delivery type matches the delivery type associated with the person contact type as defined in the Notification Preferences master configuration. This method relies on enabling opt-in for a delivery type. This is how the two processes work with each other. See Enabling Opt-in for a Delivery Type for more information. An example of this scenario is that a person contact exists and is of a type that can receive text messages, such as "Mobile Phone". Until the person contact is used on a contact preference to receive a notification, there is no reason to initiate an opt-in process.
- This process can be automatically initiated by configuring a person contact to default to an initial status of ***Pending***. When a person contact is added, the process is initiated. An example of this scenario is an email person contact initiating a process to verify the email address as soon as it is added to the system.
- When the value of an approved person contact is changed, the process is reinitiated. For example, even though the approval is attached to a specific person contact in the system, which has a unique ID, externally the approval was for a particular phone number or email address. When the person contact value (i.e. the phone number or email address) is changed, the prior approval is no longer valid. The person contact status is changed to ***Pending***.

A user can manually initiate the process. A button is provided on the Person Contact row on Person - Main. The button can be used to reinitiate the process when the customer does not have the request and needs it resent. It can also be used to manually initiate the process upfront - before the person contact is used on a contact preference.

Maintaining Person Contact via Person Phone

Person phone is referenced extensively throughout the system. All system processes have been updated to determine if phone data and email data is being maintained on person contacts or being maintained on person phone and email. See Choosing to Use Person Contact or Person Phone for more information.

To keep person contacts records in sync with a person phone, you should consider which phone types should be stored as contacts, configure the person contact type corresponding to each phone type, and update the person contact type to reference the corresponding phone type. Many phone types may be associated with a single person contact type, but a single phone type may only be associated with one person contact type.

The base product provides a batch process C1-INPUS (Create Person Contact from Person Phone/Email) that creates a person contact for all person phone records whose phone type is associated with a person contact type. The batch process can also create person contact records for the email address on the person. You would run this process to create person contact records if you already have phone or email data for persons, and you wanted to start leveraging person contact as the repository for phone and email information. See the batch process description for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> There is a primary switch on person contact records; one person contact for each unique contact routing must be set as the primary contact. When the batch process creates person contacts from person phone data, the first person contact added for each unique contact routing is set as primary. This can be controlled to some extent by your implementation by associating phone types with person contact types sequentially and running the batch process multiple times. For example, you may prefer that a home phone be the primary over a cell phone if both exist. To ensure this happens, associate the home phone with a person contact type and run the batch process. Then associate the cell phone with a person contact type and run the batch process again. This will force all home phone records to be processed first. It is recommended that regardless of how many times the batch processes run, they are run back to back. This should be thought of as a conversion of data and should be treated as such.

When a phone type is associated with a person contact type the following occurs:

- When a phone record is added, a corresponding person contact is added if one does not already exist.
- When a phone record is updated, corresponding updates are made to the person contact.
- When a phone record is deleted, the system tries to delete the corresponding person contact. All business rules for person contacts are enforced. If the person contact is associated with a contact preference, the person contact cannot be deleted, and in turn the phone record cannot be deleted.
- Person contacts associated with phone types cannot be deleted directly. The delete function is disabled.
- Contact information on person contacts associated with phone types cannot be updated directly. The contact information and extension are disabled.
- Person contacts whose person contact type is associated with a phone type cannot be added from the account contact information zone in the account portal. These need to be added through person phone.

There is no direct association between a person phone record and person contact (i.e. there are no foreign keys on either record pointing to the other). The system uses the association between phone type and person contact type and the specific phone number in order to find the associated person contact.

Your implementation may decide to use person contacts to store phone data and not associate them with phone types. Functionality related to the system use of person phone will not be available such as updating a phone number through an order.

If your implementation associates phone types with person contact types and wishes to remove this association, it is best to remove all phone types from all person contact types.

> **Warning**
>
> Making changes to the association of phone type to person contact type can have serious implications for your data. Changing the person contact type associated to a phone type will likely result in difficulty maintaining both person phone and person contact data.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Statement Construct Options

This section describes tables that must be set up before a statement construct can be set up for a person to begin receiving financial statements.

> **NOT_SUPPORTED**
>
> For more information, refer to The Big Picture of Complex Statements.

- Setting Up Statement Route Types

Setting Up Statement Route Types

Statement route types define the method used to route statements to persons. To define a statement route type, open **Admin**, and then **Billing**, and then **Statement Route Type.**

Description of Page

Enter a unique **Statement Route Type, Description** and **Statement Routing Method** for every statement route type.

> **Note**
>
> The values for Statement Routing Method are customizable using the Lookup table. This field name is STM_RTG_METH_FLG.

The next two fields control how statements that are routed using this route type are printed (both in batch and online). Refer to Technical Implementation Of Batch Statement Production for more information about producing statements in batch. Refer to Technical Implementation Of Online Statement Production for more information about online statement production.

- Use **Batch Control** to define the process that creates the flat file that is passed to your statement printing software. If you use an **Extract Algorithm** to construct the downloaded information, you can use the STMDWLD process.
- Use **Extract Algorithm** to define the plug-in component that constructs the "flat file records" that contain the information to be merged onto statements routed using this route type. This algorithm is called when a user requests an online image of a statement on Statement - Main and it may also be called by the batch statement extraction process defined above.

Setting Up Account Options

This section describes tables that must be set up before an account can receive a bill.

Copyright © 2026, Oracle and/or its affiliates.

- Setting Up Account Management Groups
- Setting Up Account Relationship Codes
- Setting Up Alert Types
- Setting Up Bill Messages
- Setting Up Bill Route Types
- Setting Up Bill Cycles
- Setting Up Customer Classes
- Setting Up Collection Classes
- Setting Up Customer Information Options

Setting Up Account Management Groups

Users are informed that something requires their attention by entries that appear in To Do lists. For example, consider what happens when billing can't find a reading (and it's not allowed to estimate):

- The billing process creates a bill segment that is in error - meter read cannot be found.
- This error bill segment, in turn, triggers the creation of a To Do entry.
- The To Do entry is addressed to a role. A role is one or more users who can "action" the To Do entry.
- When a user views their To Do entries, they see all entries addressed to all roles of which they are part.

You can optionally use account management groups (AMG) to define the respective role to be assigned to To Do entries that are associated with an account and a given To Do type. For example, you can create an AMG called Credit Risks and assign this to accounts with suspect credit. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the Credit Risks AMG. Refer to Assigning A To Do Role for more information..

> **Note**
>
> **Account management groups are optional.** You need only set up account management groups (and link them to accounts) if you wish to address specific To Do entries associated with specific accounts to specific roles.

Account management groups are defined using **Admin**, and then **Customer**, and then **Account Management Group**, and then **Add.**

Description of Page

Enter an easily recognizable **Account Management Group** code and **Description** for each account management group. Use the grid to define the **To Do Role** to be assigned to entries of a given **To Do Type** that are associated with accounts that reference the **Account Management Group**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Only To Do entries that are account-oriented take advantage of the roles defined for an account management group (because only accounts reference an account management group).

Setting Up Account Relationship Codes

When you link a person to an account, you must define in what way the person is related to the account by using an account relationship code. These codes are defined using **Admin**, and then **General**, and then **Account Relationship Type.**

Description of Page

Enter an easily recognizable **Relationship Type** and **Description** for each relationship type.

Setting Up Alert Types

Account based alerts that appear in control central have an **Alert Type**. To define valid alert types, navigate to **Admin**, and then **Customer**, and then **Alert Type.**

Description of Page

Enter an easily recognizable **Alert Type Code** and **Description** for each alert type. Specify the **Alert Days** to indicate the amount of time that alerts of this type will be effective by default. Specify a value of zero to indicate that alerts of this type will be effective indefinitely by default.

Setting Up Bill Messages

There are various informational and warning messages that may appear on an account's bills. Each message is identified with a bill message code. To define a bill message, open **Admin**, and then **Billing**, and then **Bill Message**, and then **Add.**

Description of Page

Enter a unique **Bill Message** and **Description** for every bill message.

The following attributes control how and where the bill message appears on the customer's bill:

Priority controls the order in which the message appears when multiple messages appear on a bill.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is MSG_PRIORITY_FLG.

Copyright © 2026, Oracle and/or its affiliates.

Insert Code controls whether a document should be inserted into the bill envelope when the bill message appears on a bill.

Message on Bill is the actual verbiage that appears on the customer's bill. If the message text is not static (e.g., field values need to be substituted into the body of the message), you can use the % *n* notation within the **Message on Bill** to cause field values to be substituted into a message. Refer to Substituting Field Values Into A Bill Message for more information.

Setting Up Bill Route Types

Bill route types define the method used to route bills to accounts. To define a bill route type, open **Admin**, and then **Billing** , and then **Bill Route Type.**

Description of Page

Enter a unique **Bill Route Type** and **Description** for every bill route type.

Bill Routing Method controls the type of information that may be defined when the respective **Bill Route Type** is selected on Account - Person Information. The following options are available:

- Postal: Use this method if the routing is via the postal service.
- Fax: Use this method if the routing is via fax.
- Email: Use this method if the routing is via email.

> **Note**
>
> The values for **Bill Routing Method** are customizable using the Lookup table. This field name is BILL_RTG_METH_FLG.

The next two fields control how bills that are routed using this method are printed (both in batch and online).

- Use **Batch Control** to define the background process that performs the actual download of the billing information. Refer to Technical Implementation of Printing Bills In Batch for more information about these processes.
- Use **Extract Algorithm** to define the algorithm that constructs the records that contain the information that appears on a printed bill. Refer to Printing Bills for more information.

Setting Up Bill Cycles

> **NOT_SUPPORTED**
>
> Refer to Defining Bill and Measurement Cycles for a description of how to set up bill cycles.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Customer Classes

When you set up an account, you must assign it a customer class. The topics in this section describe the customer class control table.

- Customer Class - Main
- Customer Class - Bill Messages
- Customer Class - Controls

Customer Class - Main

To set up customer classes, navigate to **Admin**, and then **Customer**, and then **Customer Class**, and then **Add** and use the **Main page** to define your Customer Class.

Description of Page

Enter a unique **Customer Class** code and **Description** for every customer class.

Use **Collection Class** to define the collection class that defaults onto new accounts that belong to this customer class. An account's collection class may be subsequently modified if the account has special collection problems or needs.

> **NOT_SUPPORTED**
>
> For more information about the significance of collection class, refer to Designing Your Collection Classes .

Turn on **Business Activity Required** if service agreements linked to accounts with this customer class require a Business Activity description to be entered.

Turn on **Open Item Accounting** if accounts belonging to this customer class are subject to open-item account. Refer to Open Item Accounting for a complete explanation of the significance of this switch.

Turn on **Non CIS Payment** if accounts belonging to this customer class are used for payments made to reduce non-CIS debt. For example, assume your company accepts payments for a county assessor and you don't want to set up a separate account for each person who pays their assessment bill. You should set up the following information to accept such payments:

- Create a new customer class called "Non CIS Customer".
- Create a SA type for each type of non-CIS payment that customers can make. Make sure to enter a distribution code on each SA type that references the appropriate revenue (or payable) account. Don't forget to indicate that each SA type is not billed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Payment Templates can be used for common types of non-CIS payment allocations. These templates are used to default the payment distribution and allow non-CIS payments to be directly allocated to specific distribution codes.

> **NOT_SUPPORTED**
>
> For more information about using Payment Templates to process non-CIS payments, refer to Non-CIS Payments.

- Create an account to which you'll book such payments. Have this account reference the new customer class. We recommend creating a separate account for each SA type that you created in the previous step.
- Create and activate a service agreement for the new account(s).

When someone pays for non-CIS debt, the operator will add a payment for the above account. On the payment, the operator should record reference information in order to know exactly why the payment was made. Refer to Payment Event - Main for more information.

You must define a variety of business rules for every division in which a customer class has customers. For example, if you operate in both California and Nevada AND you have CIS divisions for each state AND you have residential customers in each state, you must define **Customer Class Controls** for each CIS division. You do this on the Customer Class - Controls page. The grid that follows simply shows the CIS divisions for which business rules have been set up.

Customer Class - Bill Messages

When a customer class has bill messages, the system will sweep these messages onto bills created for accounts belonging to the customer class. Use this page to define a customer class's bill messages. Navigate to **Admin**, and then **Customer**, and then **Customer Class**, and then **Search** and then navigate to the **Bill Messages** page to maintain this information.

Description of Page

Use the bill messages collection to define **Bill Message** codes that should appear on bills that created for accounts that belong to a given customer class. For each message, also specify the **Start Date** and **End Date** when such a message should appear on the bill (leave **End Date** blank if the message should appear indefinitely).

Where Used

The system snaps customer class bill messages on a bill during bill completion. For more information about bill messages, refer to The Source Of Bill Messages.

Customer Class - Controls

You must define a variety of business rules for every division in which a customer class has customers. For example, if you operate in both California and Nevada AND you have CIS

Copyright © 2026, Oracle and/or its affiliates.

divisions for each state AND you have residential customers in each state, you must define **Customer Class Controls** for each CIS division in respect of the residential customer class. Open **Admin**, and then **Customer**, and then **Customer Class**, and then **Search** and then navigate to the **Controls** page to maintain this information.

Description of Page

The **Customer Class Controls** scroll contains business rules governing accounts that belong to a **CIS Division** and **Customer Class**. The following fields should be defined for each **CIS Division**:

- Use **Days Till Bill Due** to define the number of days after the bill date that the customer's bill is due. If the due date is a weekend or company holiday, the system will move the due date forward to the next workday (using the workday calendar defined on the account's CIS division).
- Specify the **Budget Plan** that defaults onto new accounts belonging to this customer class. Please note that an account's budget plan may be subsequently modified if the account has special budget processing needs. Refer to Setting Up Budget Plans for more information.
- Use **Min Credit Review Freq (Days)** to define the maximum number of days that can elapse between the reviews of an account's debt by the account debt monitor. Note, a value of zero (0) means that accounts in this customer class will be reviewed every day.
- Use **Credit Review Grace Days** to define the number of days after the bill due date that an account should be reviewed by the account debt monitor.
- Turn on the **Late Payment Charge** if customers in the class / division combination are eligible for late payment charges.
- Use **LPC Grace Days** to define the number of days after a bill's due date that a late payment charge will be generated (if the various LPC algorithms allow such - refer to How Late Payment Charges Get Calculated for the details). If the grace date falls on a weekend or holiday, the system moves the grace date to the next available workday (using the workday calendar defined on the account's CIS division).
- Specify an **Access Group** to default an access group onto an account based on the account's CIS Division and customer class. This will override the access group defaulted from the user's default access group. Note: the access group defined here will not apply if a Customer Class Control - Determine Access Group algorithm is defined and successfully returns an access group.

The grid that follows contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

> **Warning**
>
> These algorithms are typically significant system processes. The absence of an algorithm may prevent the system from operating correctly.

You can define algorithms for the following **System Events**:

Copyright © 2026, Oracle and/or its affiliates.

System Event	Optional / Required	Description
Auto Pay Amount Over Limit	Optional	This algorithm is called to handle the situation when a system-initiated automatic payment is created that exceeds the customer's maximum withdrawal limit. Specifically, this algorithm is called when: - The account has a maximum withdrawal limit on their automatic payment options - The system attempts to create an automatic payment that exceeds this amount - The automatic payment algorithm that's plugged into the installation record has logic that invokes this algorithm when the above conditions are true If you do not plug-in this type of algorithm and the above situation is detected, the automatic payment will be created and no error will be issued. Refer to How To Implement Maximum Withdrawal Limits for more information.
Bill Completion	Optional	When a bill for an account is completed, bill completion algorithms are called to do additional work. Refer to the description of the Complete button under Bill Lifecycle for a description of when this algorithm is called during the completion process.
Bill Eligibility	Optional	Algorithms for this plug-in spot are called when generating a bill in batch billing. It provides the ability to determine if an account is ineligible for billing and should therefore be skipped from further processing. If an eligibility algorithm is not used, a bill is created for any account in the open bill cycle and is later deleted by the billing process if it detects that there is no information linked to the bill.

Copyright © 2026, Oracle and/or its affiliates.

Bill Segment Freeze/Cancel	Optional	When a bill segment for an account in this customer class / division is frozen or canceled, an algorithm of this type may be called to do additional work. Refer to Bill Segment Lifecycle for more information about freezing and canceling bill segments.
Cancel Bill	Optional	This algorithm provides the ability to include additional cancel logic when canceling online. Algorithms of this type can be called in two modes: D (Determine Bill Page Buttons) and X (Cancel Bill). Mode 'D' governs whether an action button to cancel the bill will appear on the Bill page and mode 'X' performs the actual cancellation logic.
Determine Access Group	Optional	When an account is added or an account's CIS Division or customer class is changed, this plug-in spot can be used to determine the appropriate access group. This plug-in spot allows more complex logic to be defined, such as checking the access group on the main person's others account's. If a determine access group is not used, the access group can be defaulted from the customer class control or the user.
FT Freeze	Optional	When an FT is frozen, this algorithm is called to do additional work. For example, if you practice Open Item Accounting, you will need such an algorithm to handle the cancellation of match events when a financial transaction is canceled that appears on a match event. Refer to How Are Match Events Cancelled? for more information about cancellation.

Copyright © 2026, Oracle and/or its affiliates.

	LPC Eligibility	Required if the customer class / division is eligible for late payment charges	This algorithm is called by the late payment process to determine eligibility for late payments. Just because an account's customer class allows late payment charges to be calculated doesn't mean the account's delinquent service agreements will be levied late payment charges. In addition, a delinquent service agreement's SA type must reference a late payment charge algorithm. Refer to SA Type - Main for more information about SA type late payment charge issues. Refer to How Late Payment Charges Get Calculated for more information about late payment charges in general.

Note

Only One Algorithm. Only one late payment charge eligibility algorithm may be defined for a customer class / CIS division combination.

Copyright © 2026, Oracle and/or its affiliates.

Levy NSF Charge	Optional	This algorithm is called when a payment is canceled with a cancellation reason that indicates an NSF. Refer to NSF Cancellations for more information about what happens when a payment is canceled due to non-sufficient funds. **Note** **Only One Algorithm**. Only one algorithm to levy an NSF charge may be defined for a customer class / CIS division combination.
Order Completion	Optional	When an order is completed for a customer linked to this customer class, this algorithm is called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when an order is completed for customers who belong to this customer class.

Copyright © 2026, Oracle and/or its affiliates.

Overpayment Distribution	Required	When a customer pays more than they owe, this algorithm is called to determine what to do with the excess funds. Refer to Overpayment Segmentation for a description on how to configure the system to handle your overpayment requirements.

> **Note**
>
> **Only One Algorithm**. Only one overpayment distribution algorithm may be defined for a customer class / CIS division combination.

Copyright © 2026, Oracle and/or its affiliates.

Override Due Date	Optional	An account's bill due date will be equal to the bill date plus its customer class' Days Till Due. If you need to override this method for accounts in a specific customer class, specify the appropriate algorithm here. **Note** **Only One Algorithm**. Only one due date override algorithm may be defined for a customer class / CIS division combination.
Payment Cancellation	Optional	Algorithms of this type are called when a payment is canceled.

Copyright © 2026, Oracle and/or its affiliates.

Payment Distribution	Required	This algorithm is called to distribute a payment amongst an account's service agreements. Refer to Payment Distribution for more information about how payment distribution works. **Note** **Only One Algorithm**. Only one payment distribution algorithm may be defined for a customer class / CIS division combination.
Payment Freeze	Optional	When a payment is frozen, this algorithm is called to do additional work. If you practice Open Item Accounting, you will need such an algorithm to link the payment's financial transactions to the match event that was originally created when the payment was distributed. Refer to Payments and Match Events for more information.

Copyright © 2026, Oracle and/or its affiliates.

Post Bill Completion	Optional	When a customer class has algorithms of this type, they are called after the completion of a bill for an account linked to this customer class. Refer to the description of the Complete button under Bill Lifecycle for a description of when this algorithm is called during the completion process.
Pre Bill Completion	Optional	When a customer class has algorithms of this type, they are called immediately before completion starts for an account linked to this customer class. These algorithms have the potential of: • Deleting a bill. You might want a pre completion algorithm to delete a bill if a condition is detected that should inhibit the sending of a bill to a customer (e.g., the bill just contains information about recent payments). • Aborting the completion process and creating a bill exception. If the algorithm indicates this should be done, the bill is left in the pending state and a bill exception is created describing why completion was aborted. You might want a pre completion algorithm to do this if, for example, integrity checks detect there is something wrong with the account or its service agreements. If the integrity check fails, the bill can be left in the pending state and a bill exception created describing why. Refer to the description of the Complete button under Bill Lifecycle for a description of when this algorithm is called during the completion process.

Copyright © 2026, Oracle and/or its affiliates.

Quote Completion	Optional	When a quote is completed for a customer linked to this customer class, this algorithm is called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when a quote is completed for customers who belong to this customer class.
Write Off Method	Required if you allow users to write-off debt real time using the write off transaction	When a user presses the create button on the write off transaction, this algorithm is executed to write-off the selected debt. Refer to The Ramifications of Write Offs in the General Ledger for more information.

Setting Up Collection Classes

NOT_SUPPORTED

Refer to Setting Up Collection Classes for a description of how to set up collection classes.

Setting Up Customer Information Options

When you add a new person, the system is set up by default to add an account for the person and go to the Start Service page when you save the new person information. You can change this functionality by configuring the following option types on the Customer Information Options feature configuration:

- **Add Account and Start Service Default:** Indicates whether the Add Account and Start Service option is selected on the person page, by default. If this option is not configured, the Add Account and Start Service option is selected by default.
- **Post Add Person BPA Script:** Indicates a BPA script to invoke when a user successfully adds or changes a person on the person page.

When you use control central to search for accounts, the system limits your search results based on your access rights. You can change this functionality by configuring the following option type.

- **Search All Accounts:** Indicates whether a Control Central search should allow a user to search all accounts without validating the user's access rights. If the user tries to select an account without having the required access, they will not be able to navigate to the Account Information tab on Control Central for the selected account.

For more information about Feature Configurations, see Defining Feature Configurations.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Customer Contact Options

This section describes tables that must be set up before you can define customer contacts.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Customer Contacts for more information about customer contacts.

- Setting Up Customer Contact Classes
- Setting Up Customer Contact Types
- Customer Contact Tags

Setting Up Customer Contact Classes

Every customer contact record has a contact type that classifies the record for reporting purposes. And every contact type, in turn, references a customer contact "class". The class categorizes customer contacts into larger groupings for reporting purposes.

Open **Admin**, and then **Customer**, and then **Customer Contact Class** to define your customer contact classes.

Description of Page

Enter a unique **Contact Class** and **Description** for each customer contact class.

After you have created your customer contact classes, you'll be ready to setup your customer contact types.

Setting Up Customer Contact Types

Every customer contact record has a contact type that controls the behavior of the customer contact.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Customer Contacts for more information about customer contacts.

Open **Admin**, and then **Customer**, and then **Customer Contact Type**, and then **Add** to define your customer contact types.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Every customer contact type is identified by a unique combination of **Contact Class** and **Contact Type**.

Enter a brief **Description** of the customer contact type.

Only specify a **Contact Shorthand** if customer contacts of this type can be added in the Customer Contact Zone. The value you specify in this field is what the user selects to add a customer contact in this zone.

If your implementation requires restricting customer contact types by CIS Division, specify a **CIS Division** on the customer contact type.

Person Usage controls the applicability of a person on customer contacts of this type. Select ***Required*** if a person must be defined on this type of customer contact. Select ***Optional*** if a person can optionally be defined on this type of customer contact. Select ***Not Allowed*** if a person is not allowed on this type of customer contact.

Account Usage controls the applicability of an account on customer contacts of this type. Select ***Required*** if an account must be defined on this type of customer contact. Select ***Optional*** if an account can optionally be defined on this type of customer contact. Select ***Not Allowed*** if an account is not allowed on this type of customer contact.

Premise Usage controls the applicability of a premise on customer contacts of this type. Select ***Required*** if a premise must be defined on this type of customer contact. Select ***Optional*** if a premise can optionally be defined on this type of customer contact. Select ***Not Allowed*** if a premise is not allowed on this type of customer contact.

> **Note**
>
> Person usage and premise usage cannot both be set to not allowed, as the customer contact must at least one of them. Account usage is dependent on person usage.

Entity Relationship controls the validation of relationships on customer contacts of this type. *All* ensures that the account belongs to the person and that the account and premise are related. *Person and Account Only* disables the account and premise validation. *Premise and Account Only* disables the person and account validation. ***None*** disables all validation and allows any values. This setting only impacts the validation of values entered and works in conjunction with the three usage fields described above.

Use **Contact Action** if something should be triggered when customer contacts of this type are added or if the contact has a special use. When set to *Send Letter*, you must also specify a **Letter Template**. Refer to Printing Letters for more information about how letters are produced. Set to *Record Notification* if this contact type is used on template-based notification types as a record of the communication.

The **Business Object** defined here controls how customer contacts of this type are show on the customer contact portal. The system is delivered with three customer contact business objects. The value is defaulted based on the contact action.

> **Note**
>
> Unlike many objects, the customer contact business object is not stored directly on the customer contact. The system obtains it from the type and uses it to control how the customer contact is displayed, such as which fields are shown. It does not impact the behavior of the customer contact.

Copyright © 2026, Oracle and/or its affiliates.

Set the **Custom Archive Eligibility Criteria** flag to Applicable if customer contacts of this type require additional custom eligibility criteria checks when being evaluated for archiving.

> **Note**
>
> The base product provides an ILM Eligibility algorithm for Customer Contact (C1-CCILMELIG) that marks a customer contact as eligible for archiving based on the customer contact's age and the state of the related person's information. This algorithm will not mark the customer contact as eligible for archive if the customer contact type's Custom Archive Eligibility Criteria flag is Applicable. Your implementation is responsible for providing the additional criteria checks for customer contact types that require custom eligibility criteria.

Use the **Customer Contact Type Characteristics** collection to define characteristics that can be defined for contacts of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on customer contacts of a given type. Turn on the **Default** switch to default the **Characteristic Type** when customer contacts of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

Customer Contact Tags

Customer contact tagging is a way of assigning keywords or labels to a customer contact. This is done using customer contact characteristics.

Your organization can use tags to add more information or better capture the reason for the interaction with the customer. Tags can provide more specifics than the customer contact class and type and can include topics discussed that the contact class and type may not capture.

The system recognizes a customer contact characteristic as a tag when the characteristic type has the characteristic entity "characteristic tag "in addition to the customer contact characteristic entity. Like all customer contact characteristics, these must be configured as allowable customer contact characteristics on the type.

The customer contact user interface displays and maintains these characteristic types differently than other customer contact characteristics. The characteristic value is not shown and tag characteristics are not shown in the main characteristic collection.

> **Note**
>
> Characteristic types with the characteristic tag entity must be adhoc characteristic types. The system assigns a characteristic value of "1" to characteristics of characteristic types. It is the presence of the characteristic type, not its value, that is important.
>
> Currently, only customer contact has features on the user interface to display characteristic tags differently. It is not recommended to add the characteristic tag entity to non-customer contact characteristic types unless your implementation introduces a customization to display or maintain these differently.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Notification Preference Options

The topics in this section provide background information about notification preferences.

- Notification Preferences Overview
- Setting Up Notification Event Types
- Template-Based Versus Message-Based Notification Types
- Push Versus Subscription Notification Types
- Parent versus Individual Push Notification Types
- Delivery Type
- Enabling Opt-In for a Delivery Type
- Some Notification Types Use Service Task
- Designing Notification Types
- Setting Up Notification Type Algorithms
- Setting Up Delivery Types
- Setting Up Notification Types
- Setting Up the Outbound Message Type
- Setting Up the Message Sender
- Setting Up the External System and Configure the Messages
- Setting Up Notification Preferences Master Configuration
- Other Edge Application Notification

Notification Preferences Overview

The Notification Preferences functionality is comprised of two main components: sending messages out of the system and maintaining customers' communication preferences for receiving these messages.

Notifications are created by an algorithm plugged into the Create Notification plug-in spot on the Notification Type (BO). The base product provides the C1-CRE-NTF algorithm, which creates the notification via an outbound message.

The following sections describe specific notification preference concepts and how to configure the system to use notification preferences.

Setting Up Notification Event Types

This portal is used to display and maintain a Notification Event Type.

Notification Event Types are system events used to send communications or notifications to customers.

You can access the portal from **Admin**, and then **Customer**, and then **Notification Event Type**.

Copyright © 2026, Oracle and/or its affiliates.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Template-Based Versus Message-Based Notification Types

The application supports two different notification formats: template-based and message-based. For message-based notifications, the outbound message contains the message that is delivered. It contains the fixed and variable information and comes from the message catalog. For template-based, the outbound message contains the data necessary for an external system to create the notification. In this model, the external system owns the message format and fixed content and uses the data from the system for the variable information. The notification type's business object has an option that indicates the format. This business object option is used by the programs that initiate notifications from the application. The programs that initiate a notification and need to know the notification type's format and must have the responsibility to populate the correct elements before calling the create notification plug-in spot. These programs know how to prepare notification information based on the format before calling the notification type's BO create notification plug-in spot. That plug-in spot's API has elements that support both notification formats.

Push Versus Subscription Notification Types

Notification types are defined as either push – which means that a customer need not sign up to receive notifications of this type, or subscription – which means that a customer must sign up to receive this type of notification. As you can tell, they have different uses and different behaviors.

Push notification types allow users to specify communication preferences for a general type of notification, which are referred to as parent push notification types. These preferences include contact preferences and opt-out preferences, which are applied to individual notifications belonging to the parenrt type. For example, "Products and Offers" may be the parent type, and the individual notification types may include several different types of product and offers notifications that contain different messages and are targeted at different customers. Refer to Parent versus Individual Push Notification Types for more information.

In the absence of any contact preference or opt-out preference, push-based notifications can determine a default contact and send the notification using it. The logic to determine a default contact is defined via an algorithm plugged into the Determine Default Contact plug-in spot on the Notification Type (BO). The base product provides the C1-NTF-DFPRF (Determine Default Contact for Notification Type) algorithm, which looks to retrieve a default contact from the account's main person.

Subscription notification types define a particular notification and its message. Customers must have an active contact preference to receive these notifications. Subscription-based notification types can use an underlying service task. Refer to Some Notification Types Use Service Task for more information. A subscription-based notification type can also be set up to allow routing of bills or quotes to multiple destinations.

> **Note**
>
> For template-based notifications, there are distinct business objects for subscription, parent push and individual push notification types. For message-based notifications, there is a single business objects and the behavior of the notification type must be defined when configuring the notification type.

Copyright © 2026, Oracle and/or its affiliates.

Parent versus Individual Push Notification Types

Push notification types are configured as either a parent or individual notification type. An individual notification type references a parent notification type; a parent cannot reference another parent. Customers define communication preferences for parent push notification types. Notifications are sent out for individual push notification types. The preferences defined for parent push notification types are applied to the associated individual push notification types.

This means that customers do not have to supply their preferences for every individual notification that your implementation may send. Storing customer preferences for a parent notification type permits those preferences to be inherited for individual notification types added later in time under the same parent. The parent push notification types can be thought of as categories of notifications for which your customers define contact preferences and opt-out preferences.

When system processes or your custom processes attempt to send out a push notification, it should be for an individual notification type. The system will retrieve the parent notification type from the individual notification type when checking for communication preferences.

One function available to push-based notifications as described in Push Versus Subscription Notification Types is the ability to determine a default contact in the absence of a contact preference or opt-out preference. If your implementation wants to take advantage of this function, but would like the customer to supply preferences for receiving an individual notification, a parent push notification type can be set up with only one individual push notification type.

Delivery Type

Delivery Type is the channel used to send a notification. Examples are email, text message (SMS), fax, and outbound IVR. Delivery type is used throughout notification preferences. If your implementation is not using notification preferences, you would not utilize delivery type.

- Notification types list the supported delivery types. For example, a bill ready notification type may support email and SMS while a notification type to encourage customers to enroll in a budget plan may only support email. The message subject and body are configured for each notification type's supported delivery type.
- Person contact types list the allowable delivery types. For example, a cell phone person contact type may allow SMS while a home phone person contact type may not allow any delivery types. An email person contact type would allow an email delivery type unless the person contact type exists to capture a customer's email address and not to be used in conjunction with notification preferences. Note: the delivery types that can be specified on a person contact type are restricted to those defined on the person contact types contact routing. Refer to Setting Up Contact Routings for more information.
- A contact preference for a customer consists of the combination of notification type, delivery type, and person contact.

Copyright © 2026, Oracle and/or its affiliates.

Enabling Opt-In for a Delivery Type

Overview

Opt-in can be enabled for a delivery type and is configured using the **Notification Preferences** master configuration; for example, opt-in can be enabled for SMS, but not email.

To help clarify some terminology, as explained in Person Contact Status Can Be Controlled by a Process, the product is delivered with a process that initiates a service task to request opt-in or permission to use a person contact. This solution requires that opt-in is enabled for the delivery type associated with the person contact type. In this regard, the two concepts are intertwined, but enabling opt-in for a delivery type can function independently.

Enabling opt-in is centered on the person contact status, while the process that controls the status can vary; therefore, it is possible to use enable opt-in for a delivery type while manually controlling the person contact status. Enabling opt-in for a delivery type has two key impacts system behavior.

- When a contact preference is added and the delivery type is enabled for opt-in, there is logic that checks if the person contact associated with the contact preference is in the ***Approved*** status. If it is not, and the person contact type is associated with a process to control the status, that process is initiated. If the person contact type is not associated with a process to control its status, a To Do Entry is created. The To Do Type is configured on the ***Notification Preferences*** master configuration. The system is delivered with To Do Type Opt-In Errors (C1-OPTIN) for this purpose.
- Notifications will not be delivered to person contacts that are not ***Approved***. When a system process initiates an outbound notification based on an active preference, if the person contact associated with the contact preference is not ***Approved***, a flag is set in the notification indicating that it should not be delivered. It is up to the receiving system, such as Notification Center, not to deliver the notification based on this flag. This is designed so that a record of the notification being generated is still captured. Also, C1-NTF-DFPRF (Determine Default Contact for Notification Type) will not choose a person contact that is not approved.

The **Notification Preferences** master configuration help text contains detailed information about the configuration required to enable opt-in for a delivery type.

Some Notification Types Use Service Task

Prior to the introduction of notification preferences functionality, the system supported notifications set up through self-service. These notifications used service tasks to store customer preferences and in some cases trigger notifications. The destination information for the notification existed outside of the system. Notification preferences, and more particularly communication preferences, define the user's preferences on how to be notified for each notification type and where the notification is to be sent. This reduces, but does not eliminate, the need to use a service task to store a customer's preference. These service tasks will not contain the self-service information when created internally by the system

There are four existing notification types that rely on service tasks to trigger a notification. Additionally, one of the prepaid notifications uses a service task type business object with specific fields for that notification type. Refer to Designing Notification Types for information regarding the configuration requirements for certain notification types.

Copyright © 2026, Oracle and/or its affiliates.

Notification Preferences supports configuring notification types with or without a service task and only subscription-based notification types can be configured with an underlying service task.

Designing Notification Types

Some notification types require a specific configuration in order for the system to create notifications for that type.

Service Task Based Notification Types

There are system processes that rely on a hard-coded notification type lookup, referred to as notification type (legacy) on the notification type configuration. The system identifies the user-defined notification type from the lookup value.

The following identifies the configuration requirements for the notification types that are supported by existing processes in the system that rely on service tasks.

Notification Type (User Defined)	Push/Subscription	Controlled By	Notification Type (Legacy)	Service Task Type
Prepaid billing new charge	Subscription	Service Task	C1PC	Self-Service Task Type with BO C1-NotifyTaskType
Prepaid billing payment request	Subscription	Service Task	C1PP	Self-Service Task Type with BO C1-PPBPaymentNotifyTaskType
Payment Received	Subscription	Service Task	WSPR	Self-Service Task Type with BO C1-NotifyTaskType
Bill Ready	Subscription	Service Task	WXBR	Self-Service Task Type with BO C1-NotifyTaskType
Bill Due	Subscription	Service Task	WXBD	Self-Service Task Type with BO C1-NotifyTaskType
Late Payment	Subscription	Service Task	WSLP	Self-Service Task with BO C1-NotifyTaskType

Non-Service Task Based Notification Types

The following identifies the configuration requirements for the notification types that are supported by existing processes in the system that rely on the user-defined notification type to be configured.

> **Note**
>
> Marketing and forms notification types can be set up one of two ways: as subscription-based or as push-based. For it to be push-based, it would be an individual push, meaning that it references a parent push notification type specified by the implementer.

Copyright © 2026, Oracle and/or its affiliates.

Notification Type (User Defined)	Push/Subscription	Controlled By	Notes
Marketing Preference (Lead Event)	Subscription or Individual Push	N/A	Notification type configured on lead event type of BO C1-LeadEvtTypeNotificationPref
Forms Update	Subscription or Individual Push	N/A	Notification type configured on form task type of C1-FormTaskTypeForNotification
Forms Closed	Subscription or Individual Push	N/A	Notification type configured on form task type of C1-FormTaskTypeForNotification

Bill and Quote Routing

Notification types can also be used to define contact preferences for bill and quote routing for email and fax. This can enable a bill or quote to create a routing for each active contact preference. For example for a single person on an account that receives a copy of the bill, the bill can be sent to more than one email address.

Some specific configuration is needed to support this.

The notification types are to be set up as follows and must use the message-based notification type business object.

Notification Type (User Defined)	Push/Subscription	Controlled By	Bill/Quote Route Type	Algorithm
Bill Route	Subscription	Bill Route	Specify the Bill Route Type	Use the Suppression Algorithm for Bill/Quote Route
Quote Route	Subscription	Quote Route	Specify the Quote Route Type	Use the Suppression Algorithm for Bill/Quote Route

Configure only the delivery type that corresponds with the bill or quote route's routing method. For example, if the bill route type's routing method is email, configure the email delivery type on the notification type.

Configure only person contact types whose contact routing corresponds with the bill or quote route's routing method. For example, if the bill route type's routing method is email, configure only person contact types with a contact routing of email. Ensure that the person contact type allows only the delivery type specified on the notification type.

This very specific configuration is needed because the system will override the existing logic that creates bill and quote routings when a notification type exists as described above and there are active contact preferences for that notification type with the specified delivery type and the person contact's person contact type.

Collection Events

Copyright © 2026, Oracle and/or its affiliates.

The system is delivered with a collection event algorithm type that supports template-based notifications. To send collection based notifications, define individual push notification types for each collection notification. These notification types should reference a parent notification type such as “Collection Notices”. Configure algorithm(s) using the delivered algorithm type, specifying the individual push notification type(s) in the algorithm parameter. Create collection event types using these algorithms and incorporate them into your collection process templates.

Setting Up Notification Type Algorithms

Suppression Criteria

Notification types can be suppressed from appearing in a list for a particular customer. This is done by plugging in a suppression criteria algorithm on the notification type.

The base product provides the C1-ALWY-SUP (Always Suppressed) algorithm, which suppresses the notification type for all customers. This does not impact active contact preferences for the notification type, nor does it prevent processes from creating notifications of this type. This should be used with business process changes to stop sending notifications of this type. This can also be used temporarily after a notification type has been configured, but before it should be used.

The base product provides the C1-CKSP-PPB (Check Prepaid Biller Suppression) algorithm, which suppresses the notification type if the customer does not have an active prepaid biller task. This should be used on prepaid related notification types so that only customers on prepaid billing receive these notifications.

If your implementation's business rules require additional notification type suppression rules, you may introduce your own algorithm.

Override Do Not Disturb

Do not disturb information can be captured on person contacts. Person contacts are associated with a notification type to define a contact preference. When notifications are sent out, do not disturb information is included in the message to the receiving system. Some notification types might be important enough to not honor the customer’s do not disturb information. For example, a notification might be sent to a subset of customers to warn of a gas leak and evacuation and needs to be sent regardless of the customers do not disturb wishes. This is done by plugging in an override do not disturb algorithm on the notification type. Note: this does not actually override the customer’s information, but sends an indicator in the message to the receiving system that the do not disturb information should be overridden or ignored.

The base product provides the C1-OVR-DND (Override Do Not Disturb) algorithm, which returns a value of true that is passed to the receiving system in the notification.

If your implementation has more specific business rules related to overriding a customer's do not disturb preference, you may introduce your own algorithm.

Setting Up Delivery Types

Base values for Delivery Type are supplied. You can add your specific delivery types by updating the value for the C1_SS_DELIVERY_TYPE lookup field.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Notification Types

Each notification type defines the controls and behavior for defining communication preferences and sending out notifications of that type.

There are two main categories of notification types: subscription and push. These behave differently. Refer to Push Versus Subscription Notification Types for more information.

Push notification types are configured as either parent or individual push notification types. A parent push notification type is used to define communication preferences and an individual push notification type is used to send notifications. Refer to Parent versus Individual Push Notification Types for more information.

The base product supplies one Notification Type business object for message-based notification types and distinct business objects for template-based notification types. Refer to Template-Based vs. Message-Based Notification Types for more information. For more details, refer to the specifications for these business objects.

Some notification types require specific configuration. Refer to Designing Notification Types for more information.

To set up notification types, select **Admin**, and then **Customer**, and then **Notification Type +.**

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Setting Up the Outbound Message Type

An outbound message type is required for the notification outbound message. This outbound message type must reference the base C1-NotifPrefOutboundMessage (Notification Preference Outbound Message) business object.

Setting Up the Message Sender

A Message Sender is required to define how to send notifications. Use the context of the Message Sender to define the web service interface.

Setting Up the External System and Configure the Messages

Define an external system and configure the valid outbound message types and the method of communication for each (XAI, Batch, or Real Time); Real Time is generally the appropriate choice notifications).

Setting Up Notification Preferences Master Configuration

General configuration details for the notification preferences functionality is captured in the **Notification Preferences** master configuration.

For more information about specific fields in the master configuration, refer to the embedded help.

Copyright © 2026, Oracle and/or its affiliates.

Other Edge Application Notification

Other applications such as *Oracle Utilities Network Management System* or *Oracle Utilities Meter Data Management* may initiate notifications; for example, the network management product can create notifications for outages or restoration information. The customer's preferences to receive these notifications will reside in Oracle Utilities Customer Care and Billing, since this is where their contact information exists. The other applications need to be notified when an active contact preference is created for one of their notifications. The **Notification Preferences** master configuration provides the section to indicate if a Notification Type is owned by another edge application. When a contact preference is created or inactivated for one of these notification types, an outbound message is sent to notify the other edge product. When the system event in another edge application occurs that initiates a notification, that application calls and inbound web service that creates the notification.

The **Notification Preferences** master configuration help text contains detailed information about other edge applications.

Setting Up Service Agreement Options

This section describes tables that must be set up before you can define service agreements.

- Setting Up Standard Industry Codes (SIC)
- Setting Up Tax Exempt Types
- Setting Up Contract Quantity Types
- SA Type Controls Everything
- Financial Controls

Setting Up Standard Industry Codes (SIC)

A service agreement for non-residential service should reference a standard industry code (SIC). This code is used to categorize service agreements for reporting purposes. To define a SIC, open **Admin**, and then **Customer**, and then **SIC Code.**

Description of Page

Enter a unique **SIC** and **Description** for the SIC.

Setting Up Tax Exempt Types

Your rates will probably have provisions for the calculation of taxes of one type or another. Frequently you will have customers who are completely or partially exempt from these taxes. The service agreements for these customers will need to have tax exemption information in order for them to be billed properly. Tax Exempt Type is used to define the precise nature of the applicable exemption. To define the Tax Exempt Types you will use, open **Admin**, and then **Customer**, and then **Tax Exempt Type.**

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Enter a unique **Tax Exempt Type** and **Description** for each type of tax exemption.

Setting Up Contract Quantity Types

You may have customers whose contracts (service agreements) have contractual consumption limits. The service agreements for these customers must have information regarding this quantity in order to be billed properly. Contract Quantity Type is used to precisely define the nature of the quantity. To define the Contract Quantity Types, open **Admin**, and then **Rates**, and then **Contract Quantity Type.**

Description of Page

Enter a unique **Contract Quantity Type** and **Description** for each type of contract quantity.

SA Type Controls Everything

Every service agreement references a SA type. The SA type controls all aspects of a service agreement's behavior including how service is started, how bills are created, how its financial transactions are booked in the general ledger, and much more. We don't explain how to set up SA types in this section because it's only after you have set up all of the control tables in this manual that you'll be able to finally define your SA types.

NOT_SUPPORTED

For more information about SA types, refer to Defining Service Agreement Types .

Financial Controls

NOT_SUPPORTED

There are also a number of control tables that must be set up to control the bills, payments, and adjustments that are linked to a service agreement. For more information about these tables, please refer to Defining Financial Transaction Options .

Setting Up Order Options

This section describes tables that must be set up before orders can be used to start service.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to The Big Picture of Campaigns, Packages and Orders.

- Setting Up Column References
- Setting Up Order Cancellation Reasons
- Setting Up Order Hold Reasons
- Setting Up Order Feature Configurations

Setting Up Column References

A column reference must be created for each miscellaneous field that's captured on an order that doesn't reside in a characteristic. Refer to Determine The Properties Of Every Miscellaneous Field for more information.

Open **Admin**, and then **Sales & Marketing**, and then **Column Reference**, and then **Add** to define your column references.

Description of Page

Enter an easily recognizable **Column Reference** code and **Description** for each column reference.

Specify the **FK Reference** to use if this column reference uses field values from another table. Use **Long Description** to describe the data that fields using this column reference capture.

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated. You can define algorithms for the following system events: Post when order completed, Retrieve current value, Validate field value, Service task order processing, and Pre-process field value. Refer to Extract Column References for a description of these events.
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

Warning

These algorithms are typically significant processes. The absence of an algorithm may prevent the system from operating correctly.

Setting Up Order Cancellation Reasons

An order cancellation reason must be supplied when an order is cancelled. Open **Admin**, and then **Sales & Marketing**, and then **Order Cancel Reason** to define your reason codes.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Enter an easily recognizable **Order Cancel Reason** and **Description** for each order cancellation reason.

Where Used

Cancellation reasons are used when an order is canceled.

Setting Up Order Hold Reasons

An order hold reason must be supplied when an order is held. Open **Admin**, and then **Sales & Marketing**, and then **Order Hold Reason** to define your reason codes.

Description of Page

Enter an easily recognizable **Order Hold Reason** and **Description** for each order hold reason.

Where Used

Hold reasons are used when an order is held.

Setting Up Order Feature Configurations

Defining a feature configuration with a feature type of Order Configuration can increase performance of the Order page when campaigns have a large number of packages or criteria. Open **Admin**, and then **Feature Configuration**, and then **Add** to define a configuration for the feature type Order Configuration.

> **Note**
>
> **Only one**. The system expects only one order configuration feature configuration to be defined.

Description of Page

The following points describe the various **Option Types** that may be defined:

- Eligibility Tree - Suppress Error Packages node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains packages with errors in their eligibility criteria. This is an optional setting. If the option type is not defined, the error packages node is displayed, if applicable.
- Eligibility Tree - Suppress Ineligible Packages node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains packages that are not applicable to the customer based on the eligibility criteria. This is an optional setting. If the option type is not defined, the ineligible packages node is displayed, if applicable.
- Eligibility Tree - Suppress Other Campaigns node. Select this option type and define a value if you would like the Order Eligibility Tree to suppress the node that contains other eligible campaigns. This is an optional setting. If the option type is not defined, the other campaigns node is displayed, if applicable.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Program Management Options

This section describes how to set up the system for program management.

For more information about program management, see The Big Picture of Program Management

- Designing Initiative Eligibility Criteria
- Designing Related Object Criteria
- Setting Up Lead Event Types
- Overriding Initiative Sign Up Options
- Program Management Master Configuration

Designing Initiative Eligibility Criteria

The following describes guidelines for designing initiative eligibility criteria. These guidelines apply when using the base-supplied business objects and any additional business objects that the implementation defines.

There are two general types of initiative eligibility criteria business objects provided in base: specific and freeform. Your implementation decides whether to use only one of these types or a mixture of both.

> **Note**
>
> Initiative criteria has three categories: eligibility criteria, related object criteria, and participation criteria. (The participation criteria type is currently not used.) During criteria maintenance and lead generation, the system identifies the category to use through a business object option called Initiative Criteria Type. Eligibility criteria business objects are defined with an Initiative Criteria Type of either C1EC (Eligibility Criteria) or C1BT (Both Eligibility Criteria and Participation Criteria).

Specific Criteria Business Objects

Most of the base-supplied business objects are designed to perform comparisons to a specific field. The algorithm type for retrieving/determining the value to compare with is already preset in the business object, such that the Algorithm values that the user sees on the user interface are restricted to the instances of that algorithm type.

Refer to the following base business objects for more details:

- Communication Channel Criteria
- Conservation Program Criteria
- Contract Option Criteria
- Rate Schedule Criteria
- SP Type Criteria
- Service Type Criteria
- Is Account Current Criteria

Copyright © 2026, Oracle and/or its affiliates.

- Account is on Autopay Criteria
- Account is on Budget Criteria
- Number of Days Since Last Lead Criteria
- Account Customer Class Criteria
- Average Service Quantity Criteria
- Premise Cities Criteria

To add similar business objects:

- Identify the specific value for comparison. This could be a reference to an object or entity in the system, or a character string; such as, true/false.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Configure your business object schema so that the Algorithm field retrieves all algorithm instances of the algorithm type you created.
- Define algorithm instances of your algorithm type.

Freeform Criteria Business Object

The Freeform Criteria business object, on the other hand, is designed to perform comparisons for different types of values. The comparison value is determined by a Derive Value Algorithm Type that is defined as an option on this business object. A number of these Derive Value Algorithm Types are supplied with this business object and your implementation can add to these, as necessary.

> **Note**
>
> The supplied Derive Value Algorithm Types perform similar comparisons as those in the base-supplied specific criteria business objects (listed above). The Freeform Criteria business object should only be used if the available specific criteria business objects are not sufficient to perform the needed field value comparisons.

To add Derive Value Algorithm Types:

- Identify the specific value for comparison. It could be a reference to an object or entity in the system, a character string (such as, true/false), or a scalar value (such as, number of days or number of kwh).
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Define an algorithm instance for your algorithm type.
- Update the Freeform Criteria business object to add a Derive Value Algorithm Type business object option referencing your algorithm.

Repeat the above steps for each specific field to compare.

Designing Related Object Criteria

Copyright © 2026, Oracle and/or its affiliates.

The following describes guidelines for designing related object criteria. These guidelines apply when using the base-supplied business objects and additional business objects that the implementation defines.

An initiative defines the related object(s) to associate with leads via the Object Nomination algorithm on the initiative business object. The lead generation process executes this object nomination algorithm after executing initiative eligibility criteria. The base supplied parent Initiative business object (C1–Initiative) is configured with an object nomination algorithm that retrieves the premises that are associated with the account's active and pending start service agreement's service points. This algorithm applies the related object criteria, which is configured on the initiative, to each premise. If the premise is determined to be eligible, it is returned as a related object for the lead. To override the base object nomination algorithm, add another instance of the Object Nomination System Event and specify your own algorithm.

The base product supplies related object criteria business objects that look for one or more premise to associate with a lead.

There are two general types of related object criteria business objects provided in base - specific and freeform. Your implementation decides whether to use only one of these types or a mixture of both.

> **Note**
>
> Initiative criteria has three categories: eligibility criteria, related object criteria, and participation criteria. (The participation criteria type is currently not used.) During criteria maintenance and lead generation, the system identifies the category to use through a business object option called Initiative Criteria Type. Related object criteria business objects are defined with an Initiative Criteria Type of C1EO (Related Object Criteria).

Specific Criteria Business Objects

The base product supplies two business objects that perform comparisons to a specific premise-related field. The algorithm type for retrieving or determining the comparison value is preset in the business object, such that the Algorithm values that the user sees on the user interface are restricted to the instances of that algorithm type.

Refer to the following base business objects for more details:

- Premise SP Type Criteria
- Geographic Value Criteria

To add similar business objects:

- Identify the specific comparison value. It could be a reference to an object or entity in the system (such as, SP Type), or any other attribute that can be used to find the related object.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Configure your business object schema so that the Algorithm field retrieves all algorithm instances of the algorithm type you created.
- Define algorithm instances of your algorithm type.

Copyright © 2026, Oracle and/or its affiliates.

Freeform Premise Criteria Business Object

The Freeform Premise Criteria business object, on the other hand, is designed to perform comparisons to different types of values. The comparison value is determined by a Derive Value Algorithm Type that is defined as an option on this business object. Derive Value Algorithm Types are supplied with this business object and your implementation can add to these, as necessary.

> **Note**
>
> The supplied Derive Value Algorithm Types perform similar comparisons as those in the base-supplied specific criteria business objects (listed above). The Freeform Premise Criteria business object should only be used if the available specific criteria business objects are not sufficient to perform the needed field value comparisons.

To add Derive Value Algorithm Types:

- Identify the specific comparison value.
- Develop the algorithm program to retrieve or determine the comparison value.
- Configure your algorithm type using the Initiative Criteria - Derive Field Value algorithm entity and reference the program you created.
- Define an algorithm instance for your algorithm type.
- Update the Freeform Premise Criteria business object to add a Derive Value Algorithm Type business object option referencing your algorithm.

Repeat the above steps for each specific field to compare.

Setting Up Lead Event Types

Each initiative can define a structured marketing effort for the events that take place during the life of a lead. For example, one event could be that a letter or an email is sent to the customer.

Each event references a lead event type. An initiative defines a lead event type profile that specifies one or more lead event types and the sequence in which those event types should occur. Refer to The Big Picture of Initiatives for more information.

Some types of lead events may also involve sending outbound messages to an external system (*e.g.*, to notify a third party representative about a customer who has signed up). The lead event type business object can specify the outbound message type(s) to use.

Event types that involve additional workflow activities can define one or more steps; for instance, when work is outsourced to a third party representative (after customer sign-up), the third party lead event includes a list of steps to complete. Each event step has information (such as, completion date) that can be used to track progress.

The base product supplies a number of Lead Event Type business objects. For more details, refer to the specifications for these business objects.

To set up lead event types, select **Admin**, and then **Sales & Marketing**, and then **Lead Event Type.**

The Lead Event Type portal contains the following zones:

Copyright © 2026, Oracle and/or its affiliates.

Lead Event Type List

The Lead Event Type List zone lists all lead event types. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent lead event type.
- The standard actions of **Edit**, **Delete** and **Duplicate** are available for each lead event type.

Click the **Add** link in the zone's title bar to add a new lead event type.

Lead Event Type

The Lead Event Type zone contains display-only information about the Lead Event Type. This zone appears when a Lead Event Type has been broadcast form the Lead Event Type List zone or if this portal is opened via drill down from another page.

Please see the zone’s help text for information about this zone’s fields.

Overriding Initiative Sign Up Options

The behavior of the Sign Up action on Lead maintenance is controlled by two business object options on the initiative:

- Sign Up UI Map
- Sign Up Service Script

When invoking a lead’s Sign Up action, the system retrieves these options and displays or executes them accordingly.

The base-owned parent initiative business object, C1–Initiative, is supplied with the Sign Up Service Script option referencing the C1-IntvSignU script. To override this, add another *Sign Up Service Script* option with a higher sequence number.

The Sign Up UI map, on the other hand, is not plugged in on the base BO, because this could vary across implementations. The Sign Up UI map C1-SignUpResponseCh provided in base simply prompts for a Response Channel. Implementations can use this map or their own specific map.

> **Note**
>
> The system does not issue an error if the Sign Up UI map is not specified. It proceeds with executing the Sign Up Service Script and, if there are no problems, displays a sign up confirmation. The base-supplied C1-IntvSignU script includes logic to recheck the account’s eligibility for the initiative and update the *Has Signed Up* indicator on the lead.

Program Management Master Configuration

The master configuration business object for Program Management defines general configuration details for the program management functionality.

Navigate using **Admin**, and then **General**, and then **Master Configuration.** In the Master Configuration list, scroll to Program Management Configuration and click the add icon if there is no record yet or the edit icon to modify an existing record.

Copyright © 2026, Oracle and/or its affiliates.

For more information about specific fields in the master configuration, refer to the embedded help.

Setting Up Letter Templates

You can set up a customer contact type to generate a form letter whenever a customer contact of this type is added. In fact, this is the only way to generate a letter in the system.

> **NOT_SUPPORTED**
>
> Refer to Printing Letters for more information about how letters are produced.

Every customer contact that causes a letter to be sent must reference a unique letter template. To define a letter template, open **Admin**, and then **Customer**, and then **Letter Template.**

> **Note**
>
> **Document composition application users.** If you use the document composition software to produce letters, there will be a template in the software associated with each letter. The name of the template must be the same as the code associated with the letter template set up in the system.

Refer to the embedded help for details on the letter template fields.

Setting Up Customer Service Request Types

This portal is used to display and maintain a Customer Service Request Type.

A Customer Service Request Type controls how a specific customer service request is processed.

Refer to Understanding Customer Service Requests for more information.

To maintain customer service request types, open **Admin**, and then **Customer**, and then **Customer Service Request Type**.

You are brought to a query portal with options for searching for a specific customer service request type. Once a question is selected, you are brought to the maintenance portal's Main tab, to view and maintain the selected record.

The Duplicate action includes options for copying questions, service eligibility crtieria and/or response-related updates with the customer service request type.

Refer to the zone's embedded help for more information.

The Questions tab is used to configure one or more questions to ask during the flow of a start/stop/transfer service request. Refer to Setting Up Customer Service Request Questions.

The Service Eligibility tab is used to configure one or more service eligibility criteria that are applied using question responses and existing information for the customer or premise request. Refer to Setting Up Service Eligibility Criteria.

Copyright © 2026, Oracle and/or its affiliates.

The Related Updates tab is used to configure how question responses should be stored when the service request is processed and/or completed. Refer to Setting Up Response-Related Updates.

- Setting Up Consumer Contract Request Types
- Setting Up Retroactive Budget Enrollment Request Types
- Setting Up Start, Stop and Transfer Request Types
- Questions and Response Processing

Setting Up Consumer Contract Request Types

A consumer contract request type is a Customer Service Request Type that controls how a contract change request is processed.

Refer to Understanding Customer Service Requests for more information on customer service requests.

For contract change requests, the Consumer Contract Request Type also defines the service location business object that is applicable to a market and action (i.e. Add to Contract, Remove from Contract, Meter Change, etc.)

Setting Up Retroactive Budget Enrollment Request Types

A customer service request type is used to define business rules (controls) that are used by the Retroactive Budget Enrollment Process. The system is delivered with the **Enroll Customer on Budget Request Type (C1-EnrollCustBudgetRequestType)** business object intended for this use. Refer to the embedded help for details of the controls.

To enable retroactive budget enrollment functionality, your implementation needs to:

- Configure a customer service request type.
- Configure an action method using the **Enroll Customer on Budget Action Method (C1-EnrollCustBudgtActionMethod)** business object. See Action Method for more information about setting up action methods.
  - Use the customer service request type you created.
  - Set the process flow type. The system is delivered with a process flow type **Enroll Customer on Budget (C1-ENROLLCUSTOMERBUDGET)** intended for this use.

> **Note**
>
> If your business rules differ by customer class or CIS division, you can configure multiple customer service request types and multiple action methods so that the system knows which customer service request type to use for the account.

- Define a Bill Cancel Reason with **System Default** set to *Retroactively Enroll On Budget*.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This process is not supported when **Bill Correction Option** on the Installation record is set to either *Credit Note* or *Correction Option*.

In addition to defining the controls, the customer service request type also defines the contact scripting configuration for retroactive budget enrollment. See Agent Contact Scripting for more information about agent contact scripting. The script points for retroactive budget enrollment are defined on the **Budget Enrollment Contact Script Point (C1-BudgetEnrollContactScriptPt)** extendable lookup.

Setting Up Start, Stop and Transfer Request Types

The following sections describe how the request types and process flow types provided in the base product can be extended for your specific business rules.

- Verifying Identity, Checking Credit and Requiring Deposits When Starting Service
- Introducing Your Business Rules For Starting Or Transferring Service
- Understanding UI Override Controls
- Displaying More Details for the Move-to Premise
- Capturing Custom Elements in the Start, Stop, and Transfer Processes
- Adding Custom Text to the Start, Stop, and Transfer Processes
- Using the Start Service Process with Multiple Jurisdictions

Verifying Identity, Checking Credit and Requiring Deposits When Starting Service

The Start Service Request Type (request type) provides controls to help verify customers' identity and/or check their credit. The start service process provided with the system can begin with or without a person being passed into the process. The process orchestrates the following:

- When there is no person passed into the process, the assumption is that this is a new customers, but the first step is to ensure they are not already in your system. This is done by performing a person search using, name, phone, email or person identifiers. The search options are configured on the start service type. When a record is not found, the last search criteria are retained to minimize asking the customer for the same information again. If an existing person is found, the record can be selected and the process continues as if the process began with an existing customer. For this purpose, existing customers are simply those in your system before beginning the start service process. They may be long-standing customers with current service, or a person record added moments before beginning the start service process. For this reason, the start service process follows the same steps after choosing to add a new customer or selecting an existing one.
- Ensuring customers are who they say they are is often an important step in starting service, especially for new customers. This is often done by asking the customer for a unique identifier such as a government issued ID like a driver's license, or Social Security number. This information is often verified with a third-party and is sometimes combined with a credit check.

  The start service process allows you to configure what identifier types your organization accepts when both new and existing customers are starting service. This may be a subset

Copyright © 2026, Oracle and/or its affiliates.

of all of the person identifier types defined in your system. Refer to Understanding Person Identifiers for more information. Note: This help topic pertains to the primary identifier; the start service process supports additional, non-primary identifier types separately.

When setting up the start service type and configuring the primary identifier types, you need to indicate their usage. The usage for most uniquely identifying identifiers is probably **New & Existing**. These are identifier types that you accept and can uniquely identify a person or business and are probably the primary ID type on the person record. If there is an identifier that was once accepted and is still considered valid for existing customers, but is not used for new customers, these can be set to **Existing Only**. Lastly, some identifier types such as a PIN or password are used when an identifier is required, but the customer is not willing or is unable to provide a high-quality unique ID. These identifier types should be set to **New & Non-Identifying**. This allow agents to input an ID when one is required, but the process knows that the record is not a unique identifier.

For existing customer, the start service process sets an indicator when an existing identifier is found from the list of identifiers valid for existing customers. Note: the identifier need not be the primary, although it is likely that it is. For new and existing customers, a different indicator is set when a new uniquely identifying identifier is provided during the start service process. ID types that are configured as non-identifying are not considered when setting the indicator. It is not possible for both indicators to be set since the user is not prompted to enter a new ID when a valid existing one is found. These indicators are passed into plug-in spots called during the start service process.

- Your business rules, might include asking customers for their date of birth. This could be used when deciding whether or not to perform a credit check or ID verification or if it does not factor into the decision, it could be passed to the third party as part of the credit check or ID verification process. If date of birth is not part of the credit or ID process, it may be requested only if the customer is eligible for service and may be used when determining if a deposit is required. Since this could be considered sensitive data, it is possible that is used during the start service process, but not saved on the person record.

  To support a wide range of business rules, the start service process provides options for capturing and saving date of birth. When the use of date of birth is enabled, there are options to determine when the date of birth field is shown. It can be show always. It can be show only if a credit check or ID verification is going to be performed. This means the presence of the date of birth cannot factor into the decision to perform a credit or ID check. Lastly it can be shown only if service can start. There is also an option to save the date of birth as a person characteristic. For existing persons, if the date of birth already exists, it is shown, but cannot be updated during the start service process. Note: the date of birth functionality is only available for persons and not businesses.

- After the data entry is complete, the start service process calls the algorithm(s) plugged into the ID/Credit Check Should be Performed plug-in spot, passing the two ID indicators and the date of birth if enabled, to determine if an identity or credit check should be performed. This can be used to ensure the customer has provided the necessary information to pass to an external service to check credit or verify ID. This can also check the indicator that person has an existing ID and no further checking is needed. This is a multi plug-in spot so you can configure as many business rules as necessary. Each algorithm can indicate if the next algorithm should be called. The system is delivered with a sample algorithm that chooses to run a credit check based on the availability of a unique ID and a sample algorithm that bases the decision on the account history. Refer to the algorithm types for this plug-in spot for more details.

- If a credit check should be performed, the user is prompted to perform the credit check or ID verification. The algorithm plugged into the ID/Credit Check plug-in spot is then called. The user is shown if the credit check passed or failed, but the process relies on subsequent plug-in spots to determine use the results of the credit check or ID verification.

Copyright © 2026, Oracle and/or its affiliates.

- After checking the credit or verifying ID, the next step is ensuring the criteria has been met for staring service and, if so, identifying if the person or business has existing accounts that eligible for service. This is done by calling the algorithm(s) plugged into the Start Service Eligibility plug-in spot. These algorithms receive ID indicators, date of birth if enabled, and the output from the ID/credit check plug-in spot, which can be used to determine if service can start. Algorithms in this plug-in spot are also responsible for returning the eligible accounts if there are any. For example, accounts previously used for bankruptcy or those dormant too long might not be eligible for new service based on your business rules. This is a multi plug-in spot so you can configure as many business rules as necessary. Each algorithm can indicate if the next algorithm should be called. When there are existing accounts that are eligible for service, a control on the start service request type indicates if new accounts are also allowed. You would set the control if a new account can be created even when there are existing eligible accounts. Algorithms in this plug-in spot also support returning a list of ineligible accounts that are shown on the start service process to show information about all of the customer's accounts. If your business rules do not support using existing accounts for new service, your algorithms would not return any eligible accounts. The system is delivered with a sample algorithm that determines start eligibility and account eligibility. Refer to the algorithm type for this plug-in spot for more details.

  > **Note**
  >
  > If the CIS Division is configured to ensure the same CIS Division is used on related accounts, premises and service agreements (**Restrict to Account CIS Division** is set to *Restricted*), an account is eligible if its CIS Division is the same as the current CIS Division. Otherwise, the account is ineligible. The current CIS Division is determined based on the following hierarchy:
  >
  > – Premise
  >
  > – Customer Service Request Type

- Your business rules may require a deposit before a customer starts service. If so, the deposit functionality should be enabled for the start service process and optionally for the transfer service process. There is a control on the customer service request type that allows you to define if a deposit is always required or if the determination is situational, in which case algorithms in the Determine Deposit Required plug-in spot are called. The following explains the plug-in spot.

  Since deposits can be based on more than one factor, this is a a multi plug-in spot. When the plug-in spot is called, the deposit required indicator is not set and a generic message is set that a deposit is not required. The system then relies on the algorithms plugged into the plug-in spot to indicate that a deposit is required, or update the message with a specific reason that a deposit is not required.

  When called from within the Start Service process flow, the ID indictors, date of birth if enabled, and the results of the ID verification or credit check are passed into this plug-in spot and can be used in determining if a deposit is required. Note: the transfer service process uses the same plug-in spot, but is not passed any date of birth, ID or credit information since transfer only applies to accounts with current service and the transfer process does not contain the identifier and credit check logic.

  There are four sample algorithms provided to determine if a deposit is required. The following illustrates how these algorithms can be used to support both new and existing customers. When a specific condition requiring a deposit or exempting a customer from a deposit is met, the reason is output from the algorithm and displayed to the user in order to communicate with the customer. As mentioned, the system's default assumption is not required and a default message would be shown to the user if none of the algorithms

Copyright © 2026, Oracle and/or its affiliates.

output a more specific message explaining why the deposit is or is not required. Refer to the algorithm type descriptions for more detail information about each.

- C1DEPEXBAGE illustrates how an algorithm is used to waive a deposit when a condition is met, which in this case is the customer's age. This sample indicates that other algorithms in the plug-in spot should not be performed when the condition is met. If your business rules waive deposits when a condition is met and no further checking is required, you'd plug-in the algorithm(s) that check for this first. These algorithms should tell the system not to perform the additional algorithms in this plug-in spot.
- C1REQDEPAID uses the results from the ID or credit check to determine if a deposit is required. This algorithm has logic for both new and existing customers. It uses inputs including an indicator if an ID or credit check was run and, if run, the results of the ID or credit check. Based on the results, this algorithm can require a deposit. This algorithm tells the system not to perform the subsequent algorithms in this plug-in spot when a deposit is required.
- C1REQDEPBACR is intended for existing customers and uses the account's internal credit rating to determine if a deposit is required. It's possible that you'd not verify ID nor run an external credit check for existing customers, so the preceding algorithm would not have required a deposit, leaving downstream algorithms such as this one to make the determination. If the customer's internal credit rate is too low, a deposit is required, and this algorithm tells the system not to perform the subsequent algorithms in this plug-in spot.
- C1DEPEXMPBAH is the last sample and is based on account history. If the customer has recent or current service, the algorithm indicates that is deposit is not required and updates the message explaining that a deposit is not required. If there is no recent account history, the algorithm indicates that a deposit is required. In this example, this is the last algorithm, but it could be performed before other algorithms. To accommodate calling it in any sequence, this algorithm tells the system not to perform the subsequent algorithms only when a deposit is required.

> **Note**
>
> If your business follows the model where a deposit is required unless there is a reason to waive it, then your algorithms will behave in the inverse from the delivered samples. Your first algorithm plugged into the plug-in spot must set the deposit required indicator and update the default message. Subsequent algorithms or subsequent steps in the first algorithm can then clear the deposit required indicator when conditions to waive the deposit are met.

The delivered algorithms for these four plug-in spots work with each. You can design the algorithms in these four plug-in spots to work together or independently to implement your business's rule for adding new customers or ensuring existing persons or businesses meet the requirements for verifying ID or credit, starting service, choosing an account, and checking if a deposit is required.

Introducing Your Business Rules For Starting Or Transferring Service

The Start Service and Transfer Service requests are designed to allow you to define pre-requisites before customers start service. Each request has an "Waiting for Additional Actions" status where you can plug-in your business rules that define the actions required to start service. The requests have monitor algorithms to transition into and out of the waiting status.

Copyright © 2026, Oracle and/or its affiliates.

For example, if you require a deposit to be paid before starting service, you can plug-in your business rule that ensure that the deposit is paid before our process starts the customer's service.

Understanding UI Override Controls

Process flow type panels are made up of a data area or UI map fragment. The delivered start, stop, and transfer process flow type panels are built using data areas. Instead of a single data area, these panels are built using nested or included data areas. This is done to facilitate custom extension of the data area at the lowest. This is useful when your business rules include adding something at the end of one of the delivered sections. However, there are times when your business rules are more complex than just adding on to a delivered section. Some of the UI sections that are delivered can to be overridden with a custom replacement UI section. This logic is enabled through UI Override Controls on the request types.

The start, stop, and transfer processes each contain a summary header that is show at the top of each step in the process that can be overridden. By overwriting this, you can create a custom header that is displayed throughout the process flow.

When starting service, especially for a new customer, business rules can differ between different organizations or within different operating units of the same organization. For example, a utility with customers in different states may have different requirements around verifying identity or checking credit. The sections responsible for new customer and existing customer identification can be replaced with a custom experience using the UI override controls.

When the request type indicates that the UI section is overridden, the extension data area replaces the UI component of the delivered data area that it extends. It does not override or replace the data area's schema

Displaying More Details for the Move-to Premise

When starting or transferring service, additional scrutiny may be given to the move-to premise. Some of the pertinent information might related to knowing if the premise has had frequent move-ins and move-outs and what the outstanding balance is on the previous account(s). Perhaps the meter information or usage history is important.

The system is delivered with a feature that is not enabled by default. On the start and transfer customer service request type business objects, you can specify a BPA script in the **Move-To Premise Details BPA Script** BO Option type. This script should invoke a map that contains the additional premise information that is relevant to your business. The Move-To Premise steps on the start and transfer service processes display an icon next to the selected premise when this option is configured. The system is delivered with a sample BPA script (**C1SSDisPrDe**) that can be used as a guide for any custom script you wish to introduce.

Capturing Custom Elements in the Start, Stop, and Transfer Processes

Your business rule may require capturing information beyond what is delivered in the start, stop, and transfer processes. Additional elements can be added to specific places in the processes and the values then made available to the requests and therefore the request's lifecycle algorithms.

To create custom extensions in the processes, create a new data area to hold the custom elements that extends the Panel Extension Data Area where you want to make the extension. The data areas need to use UI Hints or map fragments.

Copyright © 2026, Oracle and/or its affiliates.

To pass the custom elements values from the process to the request, make the identical extension to the corresponding Request Extension Data Area. The system handles copying the values using “move by name”.

Process	Panel	Panel Extension Data Area	Request Extension Data Area
Start	Move-To Premise	C1-StartMoveToPremPanelExtend	C1-StartMoveToPremCSRExtend
	Customer Identification	C1-StartCustIDPanelExtend	C1-StartCustIDCSRExtend
	Services to Start	C1-StartSvcToStartPanelExtend	C1-StartSvcToStartCSRExtend
	Person and Account Details	C1-StartAcctDetailPanelExtend	C1-StartAcctDetailCSRExtend
Stop	Services to Stop	C1-StopSvcToStopPanelExtend	C1-StopSvcToStopCSRExtend
	Person and Account Details	C1-StopAcctDetailPanelExtend	C1-StopAcctDetailCSRExtend
Transfer	Move-To-Premise	C1-TransferMoveToPremPanelExt	C1-TransferMoveToPremCSRExt
	Services to Start	C1-TransferSvcToStartPanelExt	C1-TransferSvcToStartCSRExt
	Services to Stop	C1-TransferSvcToStopPanelExt	C1-TransferSvcToStopCSRExt
	Person and Account Details	C1-TransferAcctDetPanelExtend	C1-TransferAcctDetCSRExtend

Adding Custom Text to the Start, Stop, and Transfer Processes

When setting up start, stop and transfer customer service request type, you can configure text that appears in specific places or script points of the start, stop and transfer processes using Agent Contact Scripting. Refer to the embedded help on the Agent Contact Scripting sections for more information on how to configure this feature. The script points for start, stop, and transfer processes are defined on the **Start Service Contact Script Point (C1-StartServiceContactScriptPt)**, **Stop Service Contact Script Point (C1-StopServiceContactScriptPt)** and **Transfer Service Contact Script Point (C1-TransferSvcContactScriptPt)** extendable lookup, respectively.

Using the Start Service Process with Multiple Jurisdictions

If your organization operates in multiple jurisdictions, it is likely that you have used CIS division to segment your accounts by jurisdiction, require premises to have a CIS division by requiring it on the premise types, have distinct customer service request (CSR) types for each CIS division, and use action method to map the CSR Types to CIS Division. The functionality delivered with the system relies on this configuration.

Copyright © 2026, Oracle and/or its affiliates.

The start service process has specific functionality to support multiple jurisdictions. This functionality is enabled when the installation option to Control by CIS Division is set. Before beginning the start service process flow, the user is presented with a premise search to obtain the CIS division from the premise. This is then used to call the action method to determine the correct customer service request type and/or process flow type, assuming that the action method has been configured with specific types based on CIS division. The "Move-to-Premise", typically the first step in the start service process, is then presented to the user with the previously selected premise loaded. Further searching from this panel is not allowed. If the chosen premise is not correct, the user needs to exit the current flow and re-initiate the start service process.

Refer to The Start Service Process and Request for specific features of the start service process based on the system configuration used to support multiple jurisdictions.

Questions and Response Processing

A customer service request type defines the questions to ask the customer during the flow of self-service start / stop / transfer service requests. Questions are presented, based on the applicability to the customer and / or the premises / SPs / SA(s) being started or stopped.

The responses to questions can be inputs to service eligibility criteria processing, to determine: start options (for new/additional services), additional SAs to start (e.g. non-SP-based SAs) and changes to whether or not selected services (for start / stop) are eligible to be started / stopped.

Questions can also be configured to update person / account / premise / SP / SA attributes and characteristics with the captured responses when the start / stop / transfer service request is processed or completed.

The following sections describe question configuration and response processing.

- Understanding Customer Service Request Questions
- Understanding Service Eligibility
- Understanding Response-Related Updates
- Setting Up Customer Service Request Questions
- Setting Up Service Eligibility Criteria
- Setting Up Response-Related Updates

Understanding Customer Service Request Questions

One or more questions can be asked during the flow of a self-service New Service / Add Service / Stop Service / Transfer Service request. The questions can vary across customer service request types. For instance, the questions asked during New Service or Add Service request differ from the questions asked during a Stop Service request. Transfer Service requests can ask questions for both stopping service and starting service.

The base product provides a Customer Service Request Type Question business object (**C1-CSRTQuestion**) for configuring questions. Refer to the business object in the application for more information

The following sections describe the functionality that this business object supports.

Copyright © 2026, Oracle and/or its affiliates.

Question Structure and Valid Answers

A Customer Service Request Type Question references a Customer Question that defines the structure of the question - i.e., question text, answer style (i.e., data type), and valid answers (if applicable).

Refer to Defining Customer Questions for more details.

Question Applicability

The questions to ask during the flow of a customer service request can be determined using any of the following applicability criteria:

- The customer starting/stopping service is a Person or Business
- Account's Customer Class
- Start / Stop Premise's Premise Type
- SP Type(s) on the Premise(s) being started / stopped

Applicability criteria is optional. General questions can be configured to be always applicable.

Questions are marked as either Required or Optional. When marked as Optional on the CS Request Type Questions, it can be overridden as Required for certain SP Type(s) or Premise Type(s) or any other custom applicability critiera.

Your implementation can extend applicability criteria by:

- Adding elements to the CS Request Type Question - Main (**C1-CSRTQuestionMain**) data area, and / or
- Adding lookup values for Question Related Object Type (QSTN_REL_OBJ_TYPE_FLG)

Note that if you choose to extend the Question Related Object Types, you will also need to customize the **Read Questions** algorithm plugged in on the Customer Service Request Type BOs, to handle your additional applicability criteria.

Answer-Based Question Relationships (Conditional Questions)

A question can have one or more valid answers. Each valid answer, in turn, can determine the next question to ask.

A Customer Service Request Type Question can reference a related question for each Valid Answer. The related question must be one that is already configured on the Customer Service Request Type. Therefore, questions configuration should be a two-step process:

- Add all questions to the Customer Service Request Type.
- For each question, set up the related question(s) based on its valid answer(s).

The system has validation in place to ensure that question relationships do not result in recursion.

Question Placement Groups

A customer service request type can logically group questions based on where they are shown on the user interface. For instance, with a Transfer Service Request, the questions for stopping service are shown separately from the questions for starting service. For cases like this, the Question Placement Group should be used, to ensure that the correct set of questions are retrieved and shown on the user interface.

Copyright © 2026, Oracle and/or its affiliates.

A Customer Service Request Type Question can be designated with a Question Placement Group, which is an extendable lookup (**C1-QuestionPlacementGrpLookup**) field. Base-owned lookup values are provided for Self-Service Start Service and Self-Service Stop Service. Each lookup value defines the customer service request type business objects that can use the Question Placement Group. Your implementation can define additional lookup values, if the groupings need to be more specific / granular.

Retrieving Applicable Questions

The **Read Questions** plug-in on the Customer Service Request Type BO is responsible for retrieving the questions that are applicable to a customer service request type. Given the inputs (Customer Service Request Type, Person ID, Account ID and Premise ID), the questions are read based on applicability criteria: Person / Business, Customer Class, Question Placement Group, Premise Type, SP Type(s) and/or any implementation-specific question applicability criteria (configured in Question Related Objects).

The **Read CS Request Type Questions By Applicability (C1RDCSRTQSTN)** base algorithm will apply Person / Business, Customer Class, Premise Type and SP Type criteria. Any additional implementation-specific question applicability should be handled via custom algorithm.

The Read Questions plug-in spot is invoked using a common service script, **Read CS Request Type Questions By Applicability (C1RDCSRTQSTN)**. The Service Questions operation of the **Start Stop Transfer Tasks** inbound web service calls this script to read the applicable questions to show on the self-service user interface.

- Customizing the Questions Tree

Customizing the Questions Tree

The Questions Tree (**C1-CSRTQuestionTree**) is built on the Tree object in Framework. Refer to Trees for more information.

C1-CSRTQuestionTree is configured to display up to 25 levels of related / conditional questions. Implementations can extend this by adding tree nodes to this base-owned tree.

The tree nodes are configured with Populate Node algorithms that use messages to display the information:

- **Tree Node - CS Request Type Question Root (C1CSRTQROOT)** - populates the information on the top-level nodes. It uses the messages (11131,12409) and (11131,12410) to display the question's sort sequence, question text and status (if Inactive).
- **Tree Node - CS Request Type Question - Related Questions (C1CSRTQRELQ)** - populates the information on the child nodes that display related/conditional questions. It uses the messages (11131,12411) and (11131,12412) to display: valid answer, question's sort sequence, question text and status (if Inactive).

The text of these messages can be overridden to change the order in which the information is shown - e.g. to move the sequence number at the end.

If the base-supplied node information is not suitable to the implementation, the Populate Node algorithm can be overridden by plugging in a custom Override Information algorithm on the base tree node.

Lastly, the base product provides a Populate Node algorithm that checks if there are more related question levels than can be displayed, based on the configured number of tree nodes - i.e., either the 25 levels provided in base or the implementation-specific number of levels. To

Copyright © 2026, Oracle and/or its affiliates.

use this algorithm, configure an extra tree node level at the end and specify the Populate Node algorithm, **Tree Node - Check If More Related Questions Exist (C1CSRTQCHKRQ)**.

Understanding Service Eligibility

A service eligiblity criterion has four components:

- The value to compare. This can be either a response to a question, or a value that is algorithm-derived.
- The comparison method, which consists of an operator and a comparison value.
- The instruction or action to take, based on the comparison result, which can be any of: True, False or Insuffiicent Data. For each result, the instruction can be any of: Eligible, Not Eligible for SA Type(s), Not Eligible for Start Option(s) and Check Next Condition.
- SA Type(s) to start (or stop) and any applicable Start Options when the comparison results in an Eligible instruction.

The base product provides a CS Request Type Svc Eligibility Criteria business object (C1-CSRTSvcEligCriteria) for configuring service criteria. Refer to the business object configuration in the application for more information.

The following describe the functionality that this business object supports.

Criteria Basis and Comparison Method

The **Criteria Basis** on CS Request Type Service Eligibility Criteria determines the value to compare:

- **Question Response** uses the response to a question that is configured on the CS Request Type.
- **Derive Value Algorithm** retrieves a field or characteristic value from an existing object, such as Person, Account or Premise. The base product provides algorithms to retrieve: an attribute from Account, an attribute from Premise and a characteristic value from Person / Account / Premise / SP / SA. Refer to the algorithm types for the **CS Request Type Svc Criteria - Derive Value** algorithm entity, for details.

The **Criteria Comparison** defines the operator to use and the value to compare with either a question response or an algorithm-derived value. The base product supports the operators: =, <>, >, >=, <, <=, Between and In. The comparison value is a free-form string, which should contain comma-separate values when the operator is Between or In.

Configuring Eligible SA Types and Start Options

At least one Eligible SA Type must be configured on a service eligibility criterion. If configuring multiple SA Types, designate one of them as the default. Configure any applicable Start Option(s) and designate one as the default.

> **Note**
>
> If there is a need to set a default SA Type and Start Option without relying on a question response, configure a Derive Value algorithm that simply returns 'true' and plug in the algorithm in the service eligiblity criteria that defines the default SA Type and Start Option.

Copyright © 2026, Oracle and/or its affiliates.

Applying Service Eligibility Criteria

The **Apply Svc Criteria** plug-in on the Customer Service Request Type BO is responsible for processing service eligibility criteria, given the following inputs: the selected SPs and SAs to start / stop, the list of question responses and the service task ID.

The **Apply Service Eligibility Criteria (C1CSRTAPSEC)** base algorithm reads the service eligibility criteria and performs the criteria comparisons for each.

The **Apply Svc Criteria** plug-in spot is invoked from the common service script for response processing, **Process Question Responses (C1ProQstnRsp)**, which in turn, is called by the Process Services To Start and Process Services To Stop operations of the **Start Stop Transfer Tasks** inbound web service.

Enabling Service Eligibility Criteria

To enable service eligibility criteria:

- Identify the CS Request Type Questions for which the responses will be used as the values to compare on service eligibility criteria.
- Determine the 'Derive Value' processing that your implementation needs:
  - To use the base-provided **CS Request Type Svc Criteria - Derive Value** algorithms:
    - Configure algorithm instances for the **Retrieve Account Attribute (C1SCRETACCT)** algorithm type.
    - Configure algorithm instances for the **Retrieve Premise Attribute (C1SCRETPREM)** algorithm type.
    - Configure algorithm instances for the **Retrieve Effective-Dated Characteristic Value (C1SCRETEFFCH)** algorithm type.
  - Identify and develop any additional **CS Request Type Svc Criteria - Derive Value** algorithms.
- Configure the Service Eligibility Criteria for your implementation's self-service New Service, Add Service, Stop Service and Transfer Service customer service request types. For each criterion, specify:
  - The criteria comparison details - criteria basis, question or algorithm (depending on the criteria basis), operator and comparison value.
  - The actions to take for comparison results True, False and Insufficient Data.
  - The eligible SA Types and any applicable start options.

  Refer to Setting Up Service Eligibility Criteria for more details.

Understanding Response-Related Updates

The questions configured on the customer service request type can be set up to perform related updates when the start / stop / transfer service request is processed and / or completed.

A response-related update can store the response value in a specific attribute or characteristic on objects such as Person, Account, Premise, SP and SA. In some cases, the the response value can be used to derive the actual value to store on a particular field, e.g. when the field is a foreign key to an admin table.

Copyright © 2026, Oracle and/or its affiliates.

The base product provides a CS Request Type Related Update business object (**C1-CSRTRelUpd**) for configuring response-related updates. Refer to the business object configuration in the application for more information.

The following describes the functionality that this business object supports.

Related Update Entity Defines The Object To Update

A response-related update specifies a Related Update Entity, which identifies the maintenance object - i.e. Account, Person, Premise, SP or SA - and the part of the object to update - i.e., table field, characteristic or a custom element in the CLOB (XML) field.

Related Update Entity is an extendable lookup (**C1-CSRTRelUpdEntity**) that has base-supplied values for updates to the following:

- Fields on the Person, Account, Premise or SP tables
- Characteristics on Person, Account, Premise, SP or SA
- Custom elements in the Person or SP CLOB fields

Your implementations can add other objects to update by adding values to this lookup.

Response-Related Updates Are Staged For CS Request Processing

Response-related updates are staged into the CS Request Person / Account / Premise or directly on the CS Request, so that all object updates are done at the same time - i.e. when the CS Request transitions to a specific status.

For instance, with a Start Service Request, the Person(s) and Account are updated when the CS Request transitions to the Service Processing state. An algorithm on that state transitions the CS Request Person and CS Request Account staging records, that will in turn, update the Person(s) and Account. The CS Request SA records are also created and transitioned at that point. Thus, the updates to SA characteristics are also applied at that time. The CS Request Premise(s) are transitioned when the CS Request transitions to the Completed state. The same applies to SP-related updates that are staged directly on the CS Request, as there is no staging object for SP.

Each response-related update specifies a Processing Algorithm that is responsible for staging the update on the applicable CS Request staging object. The base product provides processing algorithms for the Related Update Entities that are supported in base. Refer to the algorithms types for the **CS Request Type Related Update - Process** algorithm entity, for more details. If your implementation opts to extend the values of the Related Update Entity lookup, you must develop the corresponding algorithms.

Staging Related Updates

The **Stage Related Updates** plug-in on the CS Request Type (BO) is responsible for invoking a customer service request type's response-related updates. It reads the Response-Related Update configuration and invokes the associated **CS Request Type Related Update - Process** algorithm for each response-related update.

The **Stage Related Updates** plug-in spot is invoked using a common service script, **Process Related Updates (C1ProcRelUpd)**.

When a self-service start / stop / transfer request is finalized, an enter algorithm on the service task's Service Processing state calls the Process Related Updates service script before transitioning the CS Request (that in turn, transitions CS Request Person, CS Request Account and CS Request SAs). Refer to the Self-Service Start/Stop/Transfer Task (C1-SSStartStopXferServiceTask) business object, for details on this lifecycle processing.

Copyright © 2026, Oracle and/or its affiliates.

Premise and SP updates are applied when the start/stop request completes and when a transfer request pre-completes. Refer to the Start Service Request (C1-StartServiceRequest), Stop Service Request (C1-StopServiceRequest) and Transfer Service Request (C1-TransferServiceRequest) business objects, for details on this lifecycle processing.

> **Note**
>
> For more information on start/stop/transfer finalization, refer to the Start Service Finalization, Stop Service Finalization and Transfer Service Finalization operations.

Enabling Response-Related Updates

To enable response-related updates for configured CS Request Type Questions:

- Determine the Related Update Entities that your implementation needs to support:
  - To use the base-provided Related Update Entities, configure the **CS Request Type Related Update - Process** algorithm instances for the specific fields and / or characteristics that should be updated from the responses.
  - Configure any additional **Related Update Entity** lookup values and develop the corresponding **CS Request Type Related Update - Process** algorithms for them.
- Identify the CS request type questions that will trigger response-related updates. Configure a Response-Related Update for each of these questions, specifying the applicable Related Update Entity and Processing Algorithm.
  Refer to Setting Up Response-Related Updates for more details.

Setting Up Customer Service Request Questions

A customer service request type can define one or more questions to ask during the flow of customer service requests, such as self-service New Service, Add Service, Stop Service and Transfer Service.

Refer to Understanding Customer Service Request Questions for more information.

To configure questions, open **Admin**, and then **Customer**, and then **Customer Service Request Type** and navigate to the **Questions** tab page.

The Questions List zone displays all questions that are configured for the customer service request type. Search filters are provided, to allow specific search using any of the following criteria: Person/Business, Customer Class, Premise Type (applicability), SP Type (applicability), Sort Sequence (range) and Status.

The Question zone displays when a record is broadcast from the Questions List. Refer to the zone's embedded help for more details.

The Questions Tree zone shows the hierarchical structure of the questions, including conditional questions. Each node navigates to the Customer Question configuration.

> **Note**
>
> The Questions Tree is built on the Tree object in Framework. Refer to Trees and Customizing the Questions Tree for more information.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Service Eligibility Criteria

A customer service request type can define one or more service eligibility crtieria to apply, using the responses to questions and/or information that is already stored in the system (e.g. person / account / premise attributes and characteristics).

Service eligiblity criteria processing can determine one or more of the following:

- Applicable start option(s) for SP-based service agreements that are are selected for starting service.
- Additional service agreements that are eligible to start. For example, non-SP-based service agreements, such as charitable contributions, equipment rentals, loans, etc.
- A change in whether or not a selected SP-based service agreement can be started/ stopped.

Refer to Understanding Service Eligibility for more information.

To configure service eligibility criteria, open **Admin**, and then **Customer**, and then **Customer Service Request Type** and navigate to the **Service Eligibility** tab page.

The Service Eligibility Criteria List zone displays all service eligibility criteria that are configured for the customer service request type. Search filters are provided, to allow specific search using any of the following criteria: Sort Sequence range, Question text and Status.

The Service Eligibility Criteria zone displays when a record is broadcast from the Service Eligibility Criteria List. Refer to the zone's embedded help for more details.

Setting Up Response-Related Updates

Question responses can trigger updates to Person, Account, Premise, SP and / or SA attributes and characteristics, when the request is processed or completed. A Response-Related Update defines: the object to update, the object's atrribute / field to update and the algorithm that will process the update. The base product provides algorithms that store question responses in: characteristics, table fields (on the object's primary table), or a custom element in the object data area (CLOB) field on Person or SP.

Refer to Understanding Response-Related Updates for more information.

To configure Related Updates, open **Admin**, and then **Customer**, and then **Customer Service Request Type** and navigate to the **Response-Related Updates** tab page.

The Response-Related Updates List zone shows all questions that are currently configured for the customer service request type. Each can be configured with a related update that stores the response to the question in an attribute / characteristic on Person, Account, Premise, SP or SA. Click **Add Related Update** or **Edit Related Update** to configure the related update.

The Response-Related Update zone displays when a record is broadcast from the Response-Related Updates List. Refer to the zone's embedded help for more details.

Defining Customer 360 Options

The following section provides an overview of how to enable Customer 360.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Customer 360 for more information on this functionality.

- Setting Up The System To Enable Customer 360

Setting Up The System To Enable Customer 360

To set up the system for Customer 360, do the following:

- Define Display Icon Override Configuration settings.
- Define Billing and Usage Configuration settings.
- Display Icon Override Configuration
- Billing and Usage Display Configuration

Display Icon Override Configuration

This section describes the process for configuring display icons that are used within the various zones for Customer 360. For example, icons that appear alongside nodes within the Premise Context tree, as well as icons that appear within records of the Customer Activity and Financial History zones.

By default, display icon references are configured at the Maintenance Object or Business Object level using the **Default Icon Reference** Option Type. For certain Maintenance Objects, these icons can be overridden based on their type fields. For example, a Customer Contact can have an override icon based on its Contact Class or Contact Type, a Service Agreement can have a override icon based on the SA Types Service Type, etc.

Display icon overrides are configured by adding or changing a Master Configuration. You can access the portal through **Admin**, and then **General**, and then **Master Configuration**.

From the Master Configuration list, add or edit the **Display Icon Override Configuration** record. Use the Add button to configure for the first time. If a record has already been added, click the Edit button instead. Use the embedded help to guide you through the options.

Billing and Usage Display Configuration

There are several options available to tailor your Customer 360 Billing & Usage data display via Master Configuration - Billing and Usage Display Configuration. The units of measure, the specific service quantities to display per rate, the default time period, as well as the summarization of the data within the bar chart can all be adjusted per your organization's needs.

Billing and Usage display settings are configured by adding or changing a Master Configuration. You can access the portal through **Admin**, and then **General**, and then **Master Configuration**.

From the Master Configuration list, add or edit the **Billing and Usage Display Configuration** record. Use the Add button to configure for the first time. If a record has already been added, click the Edit button instead. Use the embedded help to guide you through the options.

Copyright © 2026, Oracle and/or its affiliates.

Configuring the Customer Intelligence Hub

The Customer Intelligence Hub includes several features that need to be set up and configured in order to function properly.

> **Note**
>
> Before you begin, verify that you have access to **C1AIINSIGHTS** - Live insights Application Service.

This section discusses the tasks you need to complete before using the Customer Intelligence Hub, which include:

- Configuring Recommendation Cards
- Configuring Account Alert Banners
- Configuring the Customer Activity Summary
- Configuring Billing Highlights
- Configuring the Billing History Charts
- Configuring Usage History Charts
- Configuring Service Type Icons
- Configuring Fusion Knowledge Management
- Configuring Fusion Chat
- Configuring Fusion Email

- Configuring Recommendation Cards
- Configuring Account Alert Banners
- Configuring the Customer Activity Summary
- Configuring Billing Highlights
- Configuring the Billing History Charts
- Configuring Usage History Charts
- Configuring Service Type Icons
- Configuring Fusion Knowledge Management
- Configuring Fusion Chat
- Configuring Fusion Email

Configuring Recommendation Cards

The Recommendations zone of the Customer Intelligence Hub provides recommended actions or insights you can review and take for the customer. Recommendations are displayed as cards across the top of the Overview tab, and you can use the arrows to the right and left of the cards to scroll through them. Some cards are actionable, and include buttons that let you to perform an action, such as enrolling a customer in an e-bill program.

To display cards in the Recommendations zone, you must complete the following steps:

Copyright © 2026, Oracle and/or its affiliates.

- Create an Insight Group of Insight Class **CX Billing Account Insights - Agent** (C1XC) and configure Card Type Insight types.
- If your card includes book icons that launch knowledge articles, see Configuring Fusion Knowledge Management for additional configurations.
- For live insights, use extendable lookup **C1-SERVICEGUIDE** and configure the business object as follows to enable live insights:
  - Feature Flag - Live Insights: Enabled
- Optionally, you can add AI glossary terms in the **F1-AIGlossary** extendable lookup. You can the relate the terms to the insight types you create so that AI can better detect specific insight topics in the conversation transcript. This is an optional step meant to improve quality of the recommendations based on a live conversation.

Configuring Account Alert Banners

This configuration enables selected insights to appear as banners at the top of the Customer Intelligence Hub. You can use any type of account-based insights to display as banners.

If you are using account alerts, you can convert selected alerts to insights, where they can then be used as banners.

Oracle recommends being selective in the number of banners you display, as the alerts can overwhelm the screen if there are a large number of them.

Configuring Insights for Banner Alerts

Some accounts have important information that should be surfaced immediately to a customer service agent during a call. For example, if a customer is currently experiencing an outage, or is currently in the severance process, having a message at the top of the screen that notifies the agent of these circumstances can help the agent set the correct tone for the call.

You use insight types, insight groups, and the insight class **Customer Intelligence Hub Banners (C1CH)** to create these banner alerts. All insight types that are associated with the insight group that references the C1CH insight class will display as banners. If you do not associate any insight types with the C1CH insight class, no banners are displayed.

You can create your own insight types, or you can use delivered insight types such as **Account Outage Events (C1-OUTAGE-LIST)**.

To configure a banner alert for the Customer Intelligence Hub:

1. Create an insight group using the following values:
   - **Insight Class**: Customer Intelligence Hub Banners (C1CH)
   - **Valid Visual Structures**: Only Card and List structures are valid for banner messages.
2. Add related insight types to the insight group. For example, add **Account Outage Events (C1-OUTAGE-LIST).** Note that the insight types you include should be account-related alerts, as they will display in the context of an account.

For additional information about insights, see Contextual Insights.

Converting Account Alerts to Banner Alerts

If you are using account alerts, you can convert specified alert types to insight types. You then use the instructions in the previous task to configure the insight types so that they display as banners in the Customer Intelligence Hub.

Copyright © 2026, Oracle and/or its affiliates.

For each alert type you want to convert to a banner message, you will create two algorithms, then you will create a related insight type. After you have your alert insight types, you complete the steps listed in the section above, "Configuring Insights for Banner Alerts".

To convert an account alert to a banner alert:

1. Locate the alert you want to convert, and make note of the alert code. For example, the code might be **PA-BLOCK** for an alert that notified you an account was not eligible for payment arrangements.
2. Access algorithm type **C1ACCTALERTS**.
3. From the drop-down arrow next to the **Algorithm Type** field, select **Got to Algorithm** and then select **Add**.
4. Complete the following fields:
   - **Algorithm**: Name the algorithm, and note the name for use in a later step.
   - **Description**: Enter a description of the algorithm.
5. In the Algorithm Versions section, enter the alert code in the **Value** field for the **Alert Type** parameter. For example, enter **PA-BLOCK**.
6. Select **Save**.
7. Access algorithm type **C1CIALERT**.
8. From the drop-down arrow next to the **Algorithm Type** field, select **Got to Algorithm** and then select **Add**.
9. Complete the following fields:
   - **Algorithm**: Name the algorithm, and note the name for use in a later step.
   - **Description**: Enter a description of the algorithm.
10. In the Algorithm Versions section, enter the name of the algorithm you just created in the **Value** field for the **Control Central Alert Algorithm** parameter.
11. Select **Save**.
12. Create an insight type using either Card or List visual structure.
13. Complete the following fields:
    - **Insight Type**: Name the insight type, and note the name for use in a later step.
    - **Description**: Enter a description of the insight type.
14. In the **Insight Algorithm** field, enter the name of the second algorithm you created.
15. Select Save.
16. In the Insight Preview section of the form, verify that the alert looks as expected.
17. Now that you have your insight created, complete the steps listed in the section above, "Creating a Banner Alert".

For additional information about algorithms and insights, see:

- Contextual Insights
- Algorithms

Configuring the Customer Activity Summary

The Customer Activity Summary zone of the Overview tab provides an AI summary of the activities associated with the customer's account over the past year. This zone provides a

Copyright © 2026, Oracle and/or its affiliates.

quick and easy way to catch up on the important activities that could be impacting your customer, and can help you better understand the state of their account.

> **Note**
>
> This feature is not available in all regions. Contact Oracle customer support to determine if the AI service is available in your region.

To enable the display of the Customer Activity Summary zone, complete the following steps:

1. Navigate to Master Configuration, and then select **Activity History Configuration**.
2. In Activity History Configuration, navigate to the AI Summary section, and verify that the **Active** option is selected. This will activate the AI summarization feature. Unchecking it will disable the AI summarization feature, and the Customer Activity Summary zone will also be disabled

   > **Note**
   >
   > To disable the display of the Customer Activity Summary Zone, deselect this option.

3. Specify the **Number of Days** of activity history the system should retrieve for AI summarization. For example, to see a summary of the last month, enter 30 days.
4. Specify the **Maximum Number of Records** to limit the records sent to be summarized. This limit is applied in addition to the number of days.
5. Specify the **Minimum Number of Records** required to generate a summary. For example, a customer may not have enough activity records in the specified number of days to make summarization meaningful. If the number of activity records for the specified time period is below this number, the summarization is not performed and is not displayed in the zone.
6. Optionally, utilities can specify **Override Details** based on CIS Division or Customer Class. If you specify values in the **Override Details** section, the system checks the overrides before using the default values.

Configuring Billing Highlights

The Billing Highlights section of the Billing tab provides key data points about the customer's current billing situation, including whether they are enrolled in certain billing programs. Billing highlights are configured by the utility, and can include items such as:

- Auto Pay Enrollment
- Budget Billing Enrollment
- Paperless Billing (E-Bill) Enrollment

You specify which insights are available in this section by setting up insights and insight groups, as described below.

Configure the Insight Group

1. Open Insight Group Maintenance and create an Insight Group.
2. Enter the group description and detailed description, and select the **C1PE** (Program Enrollment) Insight Class.

Copyright © 2026, Oracle and/or its affiliates.

3. Under Valid Visual Structures, add the **List Insight** structure.
4. Save the insight group.

Associate Insight Types

1. Open the insight group you just created and select **Related Insight Types**.
2. Select **Edit** and then add one row for each Insight Type you want to display in Billing Highlights section.
3. For each row, enter a unique Sequence value and select the required **Insight Type**. The sequence controls the display order.
4. Save your changes.

The following example uses the `ZZ-PROGRAMENROLLMENT` Insight Group with the Program Enrollment Insight Class:

Sequence	Insight Type
30	Auto Pay Enrollment (`C1-APAYENR-LIST`)
40	Budget Billing Enrollment (`C1-BUDGENR-LIST`)
50	E-Bill Enrollment (`C1-EBILLENR-LST`)
60	Customer Web Enrollment (`C1-WEBNRL-LST`)

Verify the Configuration

After you have completed the configuration, follow these steps to verify the setup is correct:

1. Open Customer Intelligence Hub for an account that meets the configured Insight Type conditions.
2. Open the Billing tab and verify that Billing Highlights displays the expected information in the configured sequence and supported visual structure.

Configuring the Billing History Charts

The billing history charts in the Billing tab provide multiple views of the customer's billing history. The small chart at the bottom right of the Billing tab provides an at-a-glance view of up to the last six months of billing history. Each month has two bars:

- **Top bar:** This bar can be segmented to show the billed amount for each service on the account.
- **Bottom bar:** This bar represents the total ending balance for the month.

You can configure the bars to use different colors for different service types.

Configure Service Type Colors

1. Open Master Configuration Query and select **Billing and Usage Display Configuration** (`C1-C360Configuration`).
2. Select **Edit**.
3. Expand **Cost - Year View**, and then expand **Service Types and Color Configuration**.
4. Select the **Default Color** to use when a service type does not have a specific mapping or matching override.
5. Add a row for each service type, and complete these fields for each row:

Copyright © 2026, Oracle and/or its affiliates.

- Display Sequence
- Service Type
- Color
- Save the Billing and Usage Display Configuration.

6. Save your changes.

This table provides sample data you can use to configure your chart colors:

Display Sequence	Service Type	Color
1	Electric Service	Orange 080
2	Water Service	Yellow 100
3	Miscellaneous Service	Red 100
4	Gas Service	Brand 070

Configure Service-Agreement Overrides

If needed, you can use Override SA Types in the applicable service-type row if specific service agreement context requires a different color.

> **Note**
>
> The system uses override colors before using the default color for the service type. For example, an Electric Service override can specify the California CIS Division and the Electric residential SA Type.

1. Add an override row.
2. Select the applicable **CIS Division**.
3. Select the applicable **SA Type**.
4. Select the override **Color**.
5. Save your changes.

Verify the Configuration

1. Open Customer Intelligence Hub for an account with billing data for the configured service types.
2. On the Billing tab, verify that the Billing History chart legend, series colors, and display order match the configuration.
3. Open **Billing History Detail** and verify that each configured service type uses the expected color. Where an override applies, verify that it takes precedence over the default service-type color.

Configuring Usage History Charts

The usage history charts in the Usage tab provide multiple views of the customer's usage history.

The Billing and Usage Display Configuration (`C1-C360Configuration`) controls the Usage Values drop-down, Usage - Year View chart series, and Cost and Usage - Table View settings. Use the details below to configure your charts to display as you want them.

Copyright © 2026, Oracle and/or its affiliates.

Configure the Usage Values Drop-Down

1. Open Master Configuration Query and select Billing and Usage Display Configuration, select **Edit**, and expand **General Configuration**.
2. Expand **Usage Values Drop-Down**.
3. Add a row for each value that you want to display in the chart and complete these fields for each:
   - Display Sequence
   - Service Type
   - Unit of Measure
   - Cost/Usage

   > **Note**
   >
   > Select the Cost / Usage value that matches the applicable service and business requirement.
4. Save your changes.

Configure Usage - Year View Chart Series

Use the steps below to configure your usage charts. Be aware of the following:

- Use a separate row for each consumed, generated, or time-of-use series to display.
- Use distinct colors for series that display together.
- For time-of-use configurations, add rows for the applicable time-of-use values, such as Off peak, On peak, and Shoulder peak, and assign a distinct color to each displayed series.

1. Expand **Usage - Year View**, and then expand **Service Quantities and Color Configuration**.
2. Add or select the applicable rate schedule.
3. For each chart series, set the following values:
   - Display Sequence
   - Unit of Measure
   - Time of Use (if applicable)
   - SQI
   - Color
4. Save your changes.

Configure Cost and Usage - Table View

Configure the service-type selector and the service quantities to display.

1. Expand **Cost and Usage - Table View**, and then expand **Service Type Selector**.
2. Add the service types that users must be able to select and set their display sequence.

Copyright © 2026, Oracle and/or its affiliates.

Configure Service Quantities to Display

1. Expand **Service Quantities to Display**.
2. Add or select the required rate schedule.
3. For each row, complete the following fields:
   - Display Sequence
   - Unit of Measure
   - Time of Use, if applicable
   - SQI
4. Verify that the rows align with the corresponding Usage - Year View series.
5. Save your changes.

Verify the Configuration

After you complete the configuration, verify that the charts display correctly in the Customer Intelligence Hub.

1. Open Customer Intelligence Hub for an account with data that matches the configured service type and rate schedule.
2. On the Usage tab, verify that the drop-down, chart series, and colors match the master configuration.

Configuring Service Type Icons

You can configure different icons for each service type that your utility supports. This configuration is optional. If you use service type icons, they will appear in the following locations:

- Bill Summary drawers
- Payment Summary drawer

Configure Service Type Icons

1. Open Master Configuration Query and select **Display Icon Override Configuration** (`C1-DisplayIconOverrideConfig`)
2. Select **Edit**.
3. Add a row for each service type, and complete these fields for each row:
   - Service Agreement
   - Service Type
   - Display Icon
4. Save your changes.

Verify the Configuration

1. Open Customer Intelligence Hub for an account with billing data for the configured service types.
2. On the Billing tab, and then select a bill to open the Bill Summary drawer.
3. Verify that each configured service type uses the expected icon.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Fusion Knowledge Management

The Customer Intelligence Hub includes book icons you can configure to open articles that are stored in Oracle Fusion Knowledge Management (KM). When a user selects the icon, the corresponding article opens and provides information that can help resolve a call. The user can also search for and open additional articles.

To enable the display and use of book icons in the Customer Intelligence Hub, see Knowledge Articles.

Configuring Fusion Chat

This section discusses the Fusion-side setup and Service Center extensibility changes required to initiate Multi-Channel Adapter (MCA) chat, and enable an agent to launch Customer Cloud Service from a Fusion chat screen pop. The implementation uses Fusion Service chat, MCA/Omnichannel routing, the Service Center `cxSvcNavigation` screen-pop event, and the Customer Cloud Service launch script `C1CxLoad`.

The overall chat flow should follow this basic pattern:

1. Enable Fusion Chat and create an agent user with the **Chat Agent** role.
2. Add the agent to a chat queue, and set up any additional chat features, such as capacity and assignment rules, so they can receive pushed chat offers.
3. A customer initiates a chat through the supported Embedded Service client in the DCS or third-party portal.
4. The Fusion agent accepts the chat, and Fusion Service Center triggers the `cxSvcNavigation` event with `navigationType = SCREENPOP`.
5. The Service Center extension event listener calls the `cxSvcNavigationListener` action chain.
6. The action chain reads the MCA chat payload and calls the **Fusion Contact API**.
7. If the caller is known and the Fusion contact has a Customer Cloud Service source-system reference id, Customer Cloud Service is opened using `C1CxLoad` with the resolved person id.
8. If the caller is unknown, the Fusion contact does not exist, or the source-system reference id is null, Customer Cloud Service is opened using `C1CxLoad` with the Customer 360 Search page displayed.

Detail about setting up this flow are provided in this section. For additional information about using and setting up Fusion Service chat, MCA, Omnichannel routing, and Service Center, see the following topics on Oracle Help Center:

- Configure screen pop rules for Service Center
- Extend Screen Pop
- Extend Service Center navigation
- How do I enable and set up chat?
- Set Up Profile Options for Omnichannel, Chat, and MCA
- Create chat agent resources
- How do I create queues?
- Configure presence and availability privilege

Copyright © 2026, Oracle and/or its affiliates.

- Configure Oracle Cloud Applications custom roles for VB Studio
- Create an Extension in Visual Builder Studio
- MultiChannelAdapterContext
- How do I implement embedded service?
- Overview of Web Chat for Service
- Oracle Embedded Service Guide

Create an Agent User for Chat

Complete these steps to create an agent user:

1. Create or identify the Fusion Agent user who will receive MCA chat offers. See Create Sales Users for detailed instructions.
2. Confirm the user is set up as a resource (agent) according to the Fusion Service implementation model.
3. In **Security Console**, assign the Chat Agent role: `ORA_SVC_CHAT_AGENT_ABSTRACT`.
4. Assign the required service role for the Service Center user. For example, use the Customer Service Representative role.
5. Confirm the agent can sign in to Fusion Service Center and access the Omnichannel toolbar.

Enable MCA Chat and Set Profile Options

In Fusion Service, enable Omnichannel, Chat, and MCA by selecting **Setup and Maintenance**, **Service**, and then **Communication Channels**.

Use this table to set the profile option values:

Profile Options	Expected Value
ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED	Set this based on the shell mode used by the Service Center implementation. `Yes` = Global shell `No` = Nested shell
SVC_ENABLE_CHAT	Set this option to `Yes` to enable chat offers for agents.
SVC_CHAT_INLAYS_ACCESS_ENABLED	Set this option to `Yes` to enable chat inlays and bootstrap access required for the chat entry flow.
ORA_SVC_ENABLE_DEC_WITH_LIVEWINDOW	Set this option to `No`.
SVC_CHAT_ANONYMOUS_ACCESS_ENABLED	Set this option to `Yes`.
SVC_ENABLE_OMNI	Set this option to `Yes` to enable Omnichannel routing.
SVC_OMNICHANNEL_BROWSER_NOTIFICATION_ENA	Set this option to `Yes` if browser notifications are required.
SVC_OMNICHANNEL_DESKTOP_NOTIFICATION_ENA	Set this option to `Yes` if desktop notifications are required.
SVC_MCA_SHOW_CONTACT_ON_WRAP_UP	Set this option to `Yes` if contact details must appear during wrap-up.
ORA_SVC_MCA_WRAP_UP_TIMEOUT	Set this option to `0` if unlimited wrap-up is required.

Copyright © 2026, Oracle and/or its affiliates.

Profile Options	Expected Value
ORA_SVC_ENABLE_MSI_WORKSPACE	Set this option to `Yes` and `Site level`. If you are customizing the MSI screen pop behavior, enable this screen pop in addition to the standard screen pop configuration so that the customer can add additional logic to screen pop behavior for MSI.
ORA_UER_IFRAME_SOURCE_CIS_URL	Specify the Customer Cloud Service base URL. For example `https://test-ccs-corp.appoci.example.com:8084`. The Service Center action chain should use this URL instead of a hard-coded URL.

Configure the Chat Queue

Before you begin, identify or create the queue you want to use for MCA chat routing. Then, complete these steps:

1. Access the queue record.
2. Set the queue distribution mode to **Automatic (Push)** for Omnichannel routing.
3. Add the chat agent user to the queue.
4. Configure assignment rules so WCFS portal chats route to this queue.
5. Verify that the queue id and routing setup are connected to the WCFS chat initiation flow.

Optionally, you can configure the queue to bypass all rules by completing these steps:

1. Obtain the queue ID you want to set up.
2. From Fusion Setup and Maintenance, search for and open the task **Manage Service Assignment Rules**.
3. Select **Generic Queuing Rules** and then select your queue.
4. Drill down into the Rules section and specify your queue ID.

Verify Your Setup

After completing the above steps, you can initiate a chat and verify that it behaves as expected. To initiate a test chat:

1. For production validation, initiate chat through the Embedded Service client.
2. Enter the target environment and required test-customer details.
3. Start the chat and confirm the offer reaches the configured Fusion queue and agent.
4. Accept the chat in Service Center.
5. Verify the expected screen-pop behavior:
    - For a known customer with a valid Customer Cloud Service reference, the Customer Intelligence Hub page opens using the account ID linked to the resolved person ID.
    - For an unknown customer, missing Fusion contact, or missing reference ID, the Customer 360 search page opens.
6. Confirm that the agent can continue the service flow after the screen pop.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Fusion Email

The Customer Intelligence Hub includes an **Email customer** option in the Action menu that enables an agent to use Fusion Service email templates to send preconfigured emails to a customer. When the agent selects this action, the system opens an email window based on whether the person is a business or person, and the **to:** field is preloaded with the contact emails associated with the caller.

Configure Fusion URLs

You must set up URLs for the email template to use for businesses and for persons.

1. In the extensible lookup **C1-URLNavigation**, configure these two delivered values:
   - **C1-FA-EMAIL-PER** for Person.
   - **C1-FA-EMAIL-BUS** for Business.
2. For each of the above, specify these values, which are combined to construct the target URL:
   - Base URL
   - Action
   - Detail Path
   - Description
3. Set the Status to **Active**.

Create a Contact Message Template

A person with the Sales Administrator, or other authorized role, must complete this task. For additional information, see Create Email Templates on Oracle Help Center.

1. Sign in as a **Sales Administrator** or authorized setup user, and verify that you are not in a sandbox.
2. Select **Configuration** and then select **Application Composer**.
3. Select **Message Templates**.

> **Note**
>
> Do not select **Email Templates** as that is a separate feature.

4. On **Manage Message Templates**, select **Actions** and then select **Create**.
5. Complete these fields:
   - Template Name
   - Description
6. Set the object to **Contact**.
7. If the template is ready for use, select **Active**. Only active Contact templates appear when users send email from a Contact record.
8. Enter the template text in the subject and message body.
9. Use the **Attributes** list to insert Contact merge fields, such as contact name and account details.

Copyright © 2026, Oracle and/or its affiliates.

10. Save the template.

Set Up Privileges

For a user to access the message templates, their custom Sales role must include the appropriate privilege:

Purpose	Privilege Required
View or select message templates in the **Send Email** flow.	**View Sales Notifications** (`ORA_ZCA_VIEW_SALES_NOTIFICATIONS_PRIV`)

> **Note**
>
> This privilege enables the user to view or select a message template in **Send Email**. Template creation should remain limited to Sales Administrator or setup users.

Complete these steps to assign the privilege:

1. Log into Oracle Fusion as a Sales Administrator or setup user.
2. Select **Tools** and then select **Security Console**.
3. Search for the custom role assigned to the user and select **Edit Role**.
4. Open **Functional Security Policies** and select **Add Function Security Policy**.
5. Search for and add **View Sales Notifications** (`ORA_ZCA_VIEW_SALES_NOTIFICATIONS_PRIV`).
6. Save the role and ensure it is assigned to the user.
7. Sign out and back in.

Test Your Setup

After you complete the above steps, you can test your setup by completing the following steps:

1. Sign into Customer Cloud Service as a user that has been assigned to a role that includes the View Sales Notifications privilege.
2. Open the Customer Intelligence Hub and select **Send Email** from the Action menu.
3. Verify that your message template is available for selection, and select it.
4. Verify that the email appears as you expect and that the merge fields are populated correctly.

Setting Up Customer Profile Options

This section describes how to set up the system for customer profiles.

- Understanding Customer Profiles
- Setting Up Customer Profile Attribute Types
- Setting Up Customer Profile Types

Understanding Customer Profiles

Copyright © 2026, Oracle and/or its affiliates.

Customer Profiles store customer insights that can help support more personalized and engaging customer and user experiences.

The Customer Profile is linked to an Account and can capture insights received from external systems as well as information maintained by authorized users.

Customer Profile Type Controls Insight Processing

A Customer Profile Type defines the kind of insight being captured. It can be specific, such as an affordability-related insight; Or it can broader in scope, such as a Payment Profile or Credit & Collections Risk Profile. Customer Profile Type defines the types of attributes that can be stored in Customer Profiles of that type.

A Customer Profile Attribute Type defines a distinct piece of information that a Customer Profile can capture. It is linked to a Field, which allows the Customer Profile's attribute values to be validated and processed using existing rules that apply to the associated Field metadata. The Customer Profile Attribute Type also controls attribute value processing, based on the Occurrence setting:

- `Single`: only one instance of the attribute can be stored. This applies to attributes that are specific to the Account.
- `Single - Per Related Object`: one value for each related object. This applies to attributes that can vary across the objects that are related to the Account - e.g., Service Agreements, Service Points, Persons, etc..
- `Multiple`: multiple values can be stored. This applies to attributes that are lists/ collections, e.g., a list of reasons that explain a related attribute's value.

For example, a Payment Profile can capture the following:

- A Low-to-Moderate Income (LMI) indicator that can be set differently across an Account's Service Agreements. In this case, the Customer Profile Attribute Type is set up with a `Single - Per Related Object` Occurrence and with Service Agreement as the Related Object Type.
- One or more LMI Selection Reasons that are applicable when the LMI switch is set to `Y`. In this case, the Customer Profile Attribute Type is set up with a `Multiple` Occurrence and with Service Agreement as the Related Object Type.

Customer Profile Maintenance

Authorized users can add and maintain Customer Profiles and their attributes.

A Customer Profile is added for an Account. Only one Customer Profile can be active for a given type. Updates to an inactive Customer Profile (including it's attributes) reactivates it.

The Customer Profile Log includes the history of Customer Profile changes, including attribute changes (adds, updates and deletions).

Uploading Customer Profiles and Attributes

Customer Profile information can be received from external systems through the Customer Profile Business Flag Upload process (`C1-CPRUP`) that stages each incoming record as a business flag before processing it into the associated Customer Profile. During processing, the system creates a Customer Profile when one does not yet exist, or updates the existing Customer Profile. It also validates the account, attribute values, and related objects. Records with processing errors are placed in an error state for review and follow-up, while valid records are completed.

The base product provides the following business objects for uploading customer profiles:

Copyright © 2026, Oracle and/or its affiliates.

- Customer Profile Business Flag Type (`C1-CustomerProfBusFlagType`)
- Customer Profile Business Flag (`C1-CustomerProfBusinessFlag`)
- Customer Profile Type `C1-CustomerProfileType`
- Customer Profile `C1-CustomerProfile`

Refer to Uploading Customer Insights for more information.

Customer Profile Use Cases

Customer Profiles can be used to segment customers and enhance customer understanding, which can help optimize business operations when the user and customer experiences are more personalized, efficient, and engaging.

Examples of Customer Profile uses:

- Customer Profile information can be shown (or drive the display) of certain types of customer insights on the user interface. For example, if the Account's Payment Profile has an LMI indicator set to `true` for at least one of its Service Agreements, the Customer sidebar zone can show a 'Likely LMI' badge next to the customer's name. Refer to the `LMI-BADGE` Insight Type for information on the badge insight that is included with the base product.
- Customer Profile information could also drive other processes, such as Credit & Collections. For example. the Account Debt Monitor process could create a Collection Process that is tailored for customers who are in the low-to-moderate income segment. Refer to the 'Check if Account is Likely Low-to-Moderate Income (LMI)' Collection Class Control - Collection Condition algorithm (`COLL COND LM`) that is provided for this purpose.

Setting Up Customer Profile Attribute Types

A Customer Profile Attribute Type is a distinct piece of information that a customer profile can capture. For example, a Low-to-Moderate Income (LMI) indicator can be stored on a customer's Payment and/or Credit & Collection customer profiles

Refer to Understanding Customer Profiles for more information on customer profiles.

To set up customer profile attribute types, open **Admin Menu**, then **Customer Information**, then **Customer Profile Attribute Type**. You are brought to a query portal with options for searching for customer profile attribute types.

Once a customer profile attribute type has been selected, you are brought to the maintenance portal.

The **Customer Profile Attribute Type** zone shows the customer profile attribute type information and any applicable related objects.

Refer to the embedded help for more information.

Setting Up Customer Profile Types

A Customer Profile Type defines the attributes that can be stored on a particular type of customer profile.

Refer to Understanding Customer Profiles for more information on customer profiles.

Copyright © 2026, Oracle and/or its affiliates.

To set up customer profile types, open **Admin Menu**, then **Customer Information**, then **Customer Profile Type**. You are brought to a query portal with options for searching for customer profile type.

Once a customer profile type has been selected, you are brought to the maintenance portal.

The **Customer Profile Type** zone shows the customer profile type information and valid attribute types.

Refer to the embedded help for more information.

Defining Service Order Options

Service orders manage the activities that are required to enable service, disable service, and perform other operations. These activities can range from simple events (such as, a meter read) to complex events (such as, installing a power line and a new meter). They can also include both fieldwork and smart commands.

- Service Orders
- Defining Disconnect Locations
- Defining Field Activity Profiles and Types
- Setting Up Field Activity Types
- Setting Up Field Activity Type Profiles
- Setting Up Fieldwork Cancellation Reasons
- Setting Up Field Activity Remarks
- Setting Up Outage Call Types
- Setting Up Service Call Communication Types
- Setting Up Service Categories
- Setting Up Service Codes
- Service Order Management External Applications
- Setting Up Service Order Activities
- Setting Up Service Order Field Activities
- Setting Up Service Order Field Activity Types
- Configuring Service Order Field Activity Reschedule Reasons
- Field Task Types

Service Orders

This section provides an overview of service order activities.

- Service Order Overview
- Understanding Service Order Management
- Supported Service Order Processes

Copyright © 2026, Oracle and/or its affiliates.

Service Order Overview

Service Order Management provides functionality that orchestrates the field activities and smart meter commands that are necessary to change the state of service at a service point. This can involve complicated business logic due to numerous, often continually-changing variables, such as:

- The type of service order request; such as, enable service, disable service, cut, reconnect, and back-to back.
- Whether the service point is connected to the source of power, or not.
- If there is a device installed. If a device is installed, is it a smart meter, non-smart meter, or an item?
- The device's state at the service point; for example, pre-commissioned, connected, or disconnected.
- If it's a smart meters, what is the head-end system that manages the device at the service point/ If it's a non-smart meter (scalar), how close is the scheduled read date to the start date?

Additionally:

- Every implementation performs field work requests differently; for example, a cut service for non-payment is performed differently in every locale.
- Every head-end vendor requires different messages, in terms of the number and type of messages, and they change over releases of *their* software
- Field workers may perform different work than was requested by the system; for example, the system might request a simple reading, and the worker might decide to exchange the meter because the meter is old and broken..
- A service order request may be cancelled after it is scheduled or has been dispatched

Service orders are designed to resolve all of the above concerns, and provide a flexible and configurable solution to meet the challenge of managing customer service requests.

Understanding Service Order Management

At a high level, service order handles requests for service as follows:

Receive / Create Service Order Request

A service order request can be as a result of a customer requesting a change to their service such as enabling or disabling service when moving into or out of a residence, but can also be the result of other business processes, such as a request to cut service due to non-payment. Within Oracle Utilities Customer To Meter actions such as start service, stop service, and cut for non-payment result in the creation of a field activity request for each effected service point. These requests are sent to Service Order Management to be performed. In addition to requests that are created with Oracle Utilities Customer To Meter, service order requests can be received from other applications

Create Service Order Activity

When Service Order Management receives the service order request, it creates a service order activity. This activity will manage and orchestrate the other activities needed to fulfill the original service request.

Copyright © 2026, Oracle and/or its affiliates.

Evaluate Service Point and/or Meter

The service order activity evaluates the current state of the service point, meter, or item, and determines the appropriate action(s) to take to fulfill the service request.

Create Activities as Needed

Based on the evaluation of the service point, meter, or item, the service order activity creates one or more activities perform the required actions.

- Service order field activities involve sending workers into the field to perform service. This can include meter installation, meter replacement, and other activities.
- Command activities are smart meter commands used to remotely change the state of the meter. This can include connect, disconnect, checking the device status (ping), or requesting an on-demand reading.

Following each of activities, the orchestration activity re-evaluates the state of the service point, meter, or item to determine the next appropriate action(s) required to fulfill the original service request.

For example, when enabling service at a service point with a smart scalar meter, a typical scenario might involve the following:

1. Service Order Field Activity - Install Meter
2. Command Activity: Commission Device
3. Command Activity: Remote Connect
4. Command Activity: On Demand Read - Scalar

Update Field Activity Request or Requesting System

When the orchestration activity determines that everything necessary to satisfy the service order request is ready, the orchestration activity is completed. If the orchestration activity was created by a field activity request, the field activity request is also completed. If, however, the orchestration activity was created due to external system rather than a field activity request the orchestration activity will inform the requester.

Send Notification to Subscribing Systems

The service order activity can also be configured to send notifications to other subscribing systems regarding the status of the service point, meter, or item.

Supported Service Order Processes

Service Order Field Activity supports the following service order processes as part of the base package.

- Enable Service
- Disable Service
- Cut for Non-Payment
- Reconnect for Payment
- Meter Exchange
- Back-to-Back
- Cancellation Service Order Requests

Copyright © 2026, Oracle and/or its affiliates.

- Update Service Order Requests

Enable Service

Enable Service is a request to start service for a particular service point. It is typically initiated by a *Start Service* field activity request, but can also be initiated by another CIS application.

A Start Service field activity request results in the creation of an Enable Service orchestration activity. The Enable Service orchestration activity then initiates the necessary field activities and smart meter commands to change the state of service at a service point to Enabled and confirm that an initial measurement exists; for example, an enable service orchestrating activity could involve the following service order field activities and smart meter commands based on the state of the service point:

- If there is no meter at the service point and the service point is not connected at the source, a service order field activity will be created to connect the service point to its source and install a meter.
- If a smart meter exists at the service point and it is in a disconnected state, a smart meter command will be sent to the appropriate head-end system to connect the meter.
- If there is a non-smart (manual) meter at the service point and it's On, a service order field activity will be created to request a read on or near the start date (depending when the next read is scheduled).

Disable Service

Disable Service is a request to stop service for a particular service point. It is typically initiated by a *Stop Service* field activity request, but can also be initiated by another CIS application.

A Stop Service field activity request will result in the creation of a Disable Service orchestration activity in Service Orders. The Disable Service orchestration activity will initiate the necessary field activities and smart meter commands to change the state of service at a service point to Disabled and confirm that a final measurement exists; for example, a disable service orchestration activity could invoke the following service order field activities and smart meter commands based on the state of the service point:

- If there is a non-smart (manual) meter at the service point and it's on, a service order field activity will be created to turn off the meter.
- If a smart meter exists at the service point and its state is connected or commissioned, a smart meter command will be sent to the appropriate head-end system to perform a remote disconnect of the meter.

Cut for Non-Payment

Disconnect for non-payment is a type of field work that is most often initiated from a collections process that has been created from within the Credit & Collection module. A collections process is created when the customer’s current balance has been determined to be past due. One of the methods for eliciting payment is to issue a disconnect. A *Disconnect for Non-Payment* field activity request will result in the creation of a Cut Service for Non-Payment orchestration activity in Service Orders. The Cut Service for Non-Payment orchestration activity will initiate the field activities and/or smart meter commands necessary to change the state of service at a service point to the disabled state.

For a Cut for Non-Payment service order, the orchestrations steps will include determining if a smart meter command can be executed, or if a request to a mobile field work application (such as Oracle Utilities Mobile Workforce Management) to dispatch a truck needs to be executed. Another example might be to determine if a disconnect can be executed for a service point. These rules might include determining if there is life support equipment at the premise, or if the

Copyright © 2026, Oracle and/or its affiliates.

service point can be disconnected at the current time of year or business hours based on utility defined rules taking into account predominant weather conditions (for instance, the utility cannot disconnect someone in the dead of winter or the service point resides at a postal code that is affected by extreme cold weather).

Reconnect for Payment

Reconnect Service for Payment is a request to restart service for a particular service point after customer has paid off their past due balance. It is typically initiated from the Credit & Collection's Severance Process.

A Reconnect for Payment field activity request will result in the creation of a Reconnect Service for Payment orchestrator activity in Service Orders. The Reconnect for Payment orchestration activity will initiate the field activities and/or smart meter commands necessary to change the state of service at a service point to the enabled state; for example, a service order request to reconnect service could involve the following service order field activities and smart meter commands based on the status of the service point:

- If there is a non-smart (manual) meter at the service point and it's turned off, a service order field activity will be created to turn on the meter.
- If a smart meter exists at the service point and it is in a disconnected state, a smart meter command will be sent to the appropriate head-end system to connect the meter.

Meter Exchange

Meter Exchange service is basically an exchange of an old meter with a new meter. A meter exchange request may be initiated as the result of:

- A smart meter rollout.
- A customer opting-out of having a smart meter.
- A meter no longer working correctly.
- A meter reaching the end of its life expectancy.
- A customer with a manual meter enrolling in a program that requires a smart meter.
- A need to upgrade measurement capabilities.

Meter exchanges can be initiated in a variety of ways, including:

- Within service order field activities (internally)
- By an external CIS or asset management system such as Oracle Utilities Operational Device Management
- A pick-up order from a field work application such as Oracle Utilities Mobile Workforce Management

Meter Exchanges involving smart meters will include smart meter commands in addition to field activities. A request can optionally include information on the type of meter needed, depending on the requesting system; this information could be specific (the type of configuration) or general (smart or manual).

Back-to-Back

Back to Back service is service disablement and enablement in single step; for example, one customer moving out while another moves in. The main purpose of this service is to minimize dispatching field work to obtain readings for both a start and stop event.

Another example of a back-to-back scenario would involve an tenant moving out and another moving into the same premise. As long as the gap between tenants (vacancy) is within the

Copyright © 2026, Oracle and/or its affiliates.

utility company's threshold, the change would be considered as a back-to-back event. In this situation, a single field task is used for this event to take the meter reading.

A Back-to-Back Service field activity request will result in the creation of a Back-to-Back Service orchestration activity in Service Orders. The Back-to-Back Service orchestration activity will initiate the field activities and smart meter commands necessary to change the state of the service of both customers at a service point to the desired state and confirm that measurements exists for that period.

Canceling Service Order Requests

A cancellation request basically cancels any orchestration and/or specific activities that are in progress. If the cancellation request is for an orchestration activity then that orchestration activity along with corresponding activities will be discarded. If the cancellation request is for a specific activity then only that specific activity will be discarded.

Depending on the status of a specific activity, a Cancel orchestration activity will either discard the activity or an outbound communication will be sent to the field work system to request a cancel the specific activity. Once the cancellation sent to the field work system is confirmed as successful, the Cancel orchestration activity will also cancel the parent orchestration activity and/or specific activities. If the cancellation sent to the field work system is not successful then the Cancel orchestration activity will be discarded. When it is discarded, it will send a response message to the requester indicating the cancellation was unsuccessful. A cancellation request can be initiated in a variety of ways, including by cancelling a field activity request , within Service Orders itself (internally), and by an asset management system such as Oracle Utilities Operational Device Management. Cancellation requests can also be created as pick-up orders from a field work application such as Oracle Utilities Mobile Workforce Management.

Update Service Order Requests

An update request handles requests for updates to an existing enablement, disablement, cut for nonpayment, and reconnect for payment orchestration activity and their related specific field activities.

The types of data available for update include the following:

- Instructions
- Comments
- Start Date/Time
- Contact Name
- Main Phone

Additionally, an update request can check to see if appointment information is different between the specific activity and the update orchestration activity. Depending on the status of the specific activity, direct updates will be made to the activities and communications within Service Orders or an outbound communication will be sent to the field work system to request an update of the specific activity fields that have been changed. Once that update to the field work system is confirmed as successful the Update orchestration activity will update the effected orchestration activity. If the updates are not successful the Update orchestration activity will be discarded. When it is discarded, it will send a response message to the requester indicating the update was unsuccessful.

Update service requests can be initiated by a field activity request or within Service Orders itself (internally).

Copyright © 2026, Oracle and/or its affiliates.

Defining Disconnect Locations

When a service point is disconnected from the supply source, a disconnect location must be specified. This location defines where service was severed.

To define disconnect location codes, open **Disconnect Location** page.

Description of Page

Enter a **Disconnect Location** and **Description** for every disconnect location.

Defining Field Activity Profiles and Types

A field activity request initiates a service order to perform a task at a service point. Every field activity request references a field activity type.

Most field activity requests are created:

- When a customer service representative starts or stops service at a premise. When this happens, the system automatically creates field activity requests for each service point.
- When service is cut due to lack of payment.

If you set the system up correctly, your CSRs won't have to create these field activity requests. Rather, the system creates field activity requests based on the information you set up in your field activity type profiles.

- Refer to Starting Service & Field Activities for a description of how field activity requests are created to start service.
- Refer to Stopping Service & Field Activities for a description of how field activity requests are created to stop service.
- Field activities used to cut service due to non-payment are created by two modules:
  - If you use severance process to stop a service agreement due to nonpayment, refer to Field Events And Their Activities for a description of how these field activity requests are created.
  - If you use cut processes to stop a service agreement due to nonpayment, refer to Field Activities To Cut and Reconnect Service for a description of how these field activity requests are created.

The topics in this section describe how to design your field activity profiles and field activity types.

- How Does A Field Activity Type Profile Get Used?
- Designing Field Activity Type Profiles
- Designing Field Activity Types

How Does A Field Activity Type Profile Get Used?

A field activity type profile contains:

Copyright © 2026, Oracle and/or its affiliates.

- A list of the types of field activity requests that can be made for service points.
- A matrix defining the specific activity type(s) to generate in order to start service, stop service, or disconnect due to nonpayment.

You may wonder how field activity type profiles get related to your service points. It's a little indirect, but the indirection provides a great deal of flexibility:

- Every service point references an SP type.
- Every SP type references the specific field activity type profile used by the start/stop and credit and collections process to generate field activity requests.

An example will help illustrate how this works:

- When you start service at a specific service point, the system extracts the service point's SP type.
- Then, the system determines the field activity type profile that is to be used on this SP type.
- And finally, it generates the appropriate field activity request.

Designing Field Activity Type Profiles

With Service Order Management typically only a single field activity type profile will be needed for the meter and item related service points. This profile should have a template for each of the following with a service point condition of *External Fieldwork Orchestration* to indicate that Service Order Management is orchestrating the field work:

- Start Service
- Stop Service
- Start/Stop
- Cut for Non-Payment
- Reconnect for Payment

Designing Field Activity Types

A field activity type is required for the following events:

- Start service
- Stop service
- Cut for non-payment
- Reconnect for payment
- Back-to-back

Setting Up Field Activity Types

The topics in this section describe how to set up field activity types.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about field activity types, refer to Defining Field Activity Profiles and Types.

- Field Activity Type - Main
- Field Activity Type - FA Characteristics
- Field Activity Type - FA Completion Control
- Field Activity Type - SP Type Review

Field Activity Type - Main

You begin to define a field activity type by opening **Admin**, and then **Field Work**, and then **Field Activity Type**, and then **Add.**

Description of Page

Enter an easily recognizable **Activity Type** and **Description** for each field activity type.

Use **Field Activity Priority** to define the priority associated with field activity requests of a given type.

Note

The values for this field are customizable using the Lookup table. This field name is FA_PRIORITY_FLG.

Turn on **Display as Alert** if Control Central should display an alert if its premise has a completed field activity request of this type. If this switch is on,

- Use **Nbr Days Alert Active** to define the number of days the alert should appear on Control Central. The field activity request's scheduled date is used as the start date for the alert period.
- Enter the **Alert Information** to appear on Control Central.

Note

Recommendation. We recommend only using this feature on unusual field activity types (e.g., cut for non-payment) so that a CSR is not presented with an alert for every field activity request.

Enter the **Business Object** for this FA Type if the FA type will need access to the BO plug-in spots. If the FA type has a business object specified, the system applies the business object's rules when the FA is added, changed, or deleted. This includes the business object's validation algorithms (only executed after the "core validation" specified on the maintenance object is done), post-processing algorithms, and audit algorithms. The maintenance object for the FA

Copyright © 2026, Oracle and/or its affiliates.

should specify an algorithm that finds its business object using its FA Type. Refer to the base package Determine Standard Business Object algorithm for an example of this type of algorithm.

Field Activity Step is not applicable when Service Order Management is used.

Use the **Characteristic** collection to define **Characteristic Type**s and their respective **Characteristic Values** to describe characteristics that are common to all field activity requests of this type.

> **Note**
>
> You can only choose characteristic types defined as permissible on the field activity type record.

Field Activity Type - FA Characteristics

Description of Page

Use the **Characteristics** collection to define characteristics that can be defined for field activity requests of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on field activity requests of a given type. Turn on the **Default** switch to default the **Characteristic Type** when field activity requests of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

> **Note**
>
> **Field activity requests created by the system**. When setting the Required switch, remember that most field activities are created by the system. Only turn on the required switch for these types of activities if a default characteristic value can also be indicated

Field Activity Type - FA Completion Control

Description of Page

> **Note**
>
> **These algorithms are optional.** The use of completion algorithms on a field activity type is optional. You would only use them if you have special functions that should be executed when a given field activity type is completed.

The **Field Activity Completion Controls** tab is used when an algorithm should be executed when a field activity request is completed. The type of algorithm may differ based on the CIS division in which the service point's premise is located. The following fields must be defined:

Copyright © 2026, Oracle and/or its affiliates.

CIS Division Defines the division associated for which the algorithm will be executed. The system will only execute the algorithm when a field activity request is performed at a service point whose premise is governed by the division.

FA Completion Algorithm Defines the algorithm that will be executed when a field activity request is performed at a service point whose premise is governed by the associated division.

Field Activity Type - SP Type Review

Not applicable when Service Order Management is being used.

Setting Up Field Activity Type Profiles

The topics in this section describe the pages used to define a field activity type profile.

> **Note**
>
> **When a new profile is added.** When you introduce a new field activity type profile, you must define the SP types that use it. Refer to Understanding Service Point Types for more information.

> **NOT_SUPPORTED**
>
> **For more information** about field activity type profiles, refer to Designing Your Field Activity Profiles & Types.

- Field Activity Type Profile - Main
- Field Activity Type Profile - Template
- Field Activity Requests Initiated To Start Service
- Field Activity Requests Initiated To Stop Service
- Field Activity Requests Initiated For Back-to-Back Service
- Field Activity Requests Initiated To Cut Service Due To Non-Payment
- Field Activity Requests Initiated To Place A Disconnect Warning At A Service Point
- Field Activity Requests Initiated To Reconnect Service At A Service Point
- Field Activity Requests Initiated To Reread A Meter At A Service Point
- Defining A Profile's Valid Field Activity Types

Field Activity Type Profile - Main

Description of Page

Enter a unique **Field Activity Type Profile** and **Description** for the activity type profile.

Copyright © 2026, Oracle and/or its affiliates.

The **FA Type Profile Templates** indicate the templates that exist for this field activity type profile. Use the drill down button to go directly to the desired template. Alternatively, you can go to the **Template** tab and scroll until you find the correct template.

Field Activity Type Profile - Template

Description of Page

The information in the **Field Activity Profile Template** collection defines the field activity request(s) created for each situation identified by the **Customer Event**. The possible customer events are Cut for Non-payment, Disconnect Warning, Reconnect for Payment, Reread, Stop Service, Start Service, and Start/Stop.

> **Note**
>
> Field activity requests do not need to be defined for Reread when Service Order Management is being used.

Other customer events can be defined on the Look Up page (search for the CUST_EVT_FLG field name). Refer to the following sections for more detail about customer events included with the base product.

The fields defined for each event are common. You define the field activity request(s) to be generated given the condition of the service point. The following fields display:

SP Field Condition Define the condition of the service point associated with the field activity. Define *External Fieldwork Orchestration* when Service Order Management is being used.

Sequence You will typically have a single field activity request for any specific combination of SP Field Condition and Disconnect Location. Therefore you'll just have a single sequence (say 10) for each combination. If you need to generate multiple field activities based on a given combination, use a unique sequence number for each activity.

No Activity Turn on this switch if no field activity should be generated for the condition.

Disconnect Location this is not applicable when Service Order Management is being used.

Activity Type Define the type of field activity request to be generated. You should take care to ensure that the activity type is defined as valid for the SP type (on the last page).

Where Used

Refer to the following sections for information about where each template is used.

Field Activity Requests Initiated To Start Service

Use the customer event Start to define the field activity request(s) used to start service at a service point whose SP type references this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Copyright © 2026, Oracle and/or its affiliates.

The start service process uses this information to determine the type of field activity requests to create to start service at a service point.

> **Warning**
>
> Warning. Field activity requests will only be created for starts if you have defined the appropriate field work creation algorithm on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

Field Activity Requests Initiated To Stop Service

Use the customer event Stop to define the field activity request(s) used to stop service at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

The stop service process uses this information to determine the type of field activity requests to create to stop service at a service point.

> **Warning**
>
> Warning. Field activity requests will only be created for stops if you have defined the appropriate field work creation algorithm on the service agreement's SA type. Refer to SA Type - Algorithms for more information.

Field Activity Requests Initiated For Back-to-Back Service

Use the customer event Start/Stop to define the field activity request(s) used to stop service for one customer and start service for another at a service point whose SP type uses this profile.

> **Note**
>
> **Terminology.** We use the term **back-to-back** to describe the situation when a single field activity request supports both the stop and start service requests. The system sets up a back-to-back situation by default when it is aware of both the start and stop customers at a premise.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Copyright © 2026, Oracle and/or its affiliates.

The start/stop service process uses this information to determine the type of field activity requests to create to start service at a service point.

Field Activity Requests Initiated To Cut Service Due To Non-Payment

Use the customer event Cut for Non-Payment to define the field activity request(s) used to cut service at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to cut service at a service point.

Field Activity Requests Initiated To Place A Disconnect Warning At A Service Point

Use the customer event Disconnect Warning to define the field activity request(s) used to place a disconnect warning at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to leave a disconnect warning at a service point.

Field Activity Requests Initiated To Reconnect Service At A Service Point

Use the customer event Reconnect for Payment to define the field activity request(s) used to reconnect service (after being cut) at a service point whose SP type uses this profile.

Description of Page

See Field Activity Type Profile - Template for a description of the fields.

Where Used

Severance and cut events use this information to determine the type of field activity requests to create to reconnect service at a service point.

Field Activity Requests Initiated To Reread A Meter At A Service Point

This template is not applicable when Service Order Management is being used.

Copyright © 2026, Oracle and/or its affiliates.

Defining A Profile's Valid Field Activity Types

Description of Page

The **Field Activity Type** collection shows the field activity requests that may be created for at service points whose SP type references the field activity type profile.

Where Used

This information is used to control the types of field activity requests that may be created for at a service point.

Setting Up Fieldwork Cancellation Reasons

Description of Page

Enter a **Cancel Reason** and **Description** for every field activity request cancellation reason.

Only use **System Default** on those reason codes that are placed on field activity requests that are automatically canceled by the system. The following table lists the valid values and the condition where this cancel reason is used.

System Default	System Condition
Cut Process canceled	Placed on field activity requests that are canceled when a Cut Process is canceled.
SA Start/Stop canceled	Placed on field activity requests that are canceled when a pending start/stop is canceled.
Severance Process canceled	Placed on field activity requests that are canceled when a severance process is cancelesd.

> **Note**
>
> **Required values.** You must have one reason code defined for each of the System Default values that corresponds to an event that may occur in your implementation.

Setting Up Field Activity Remarks

> **Note**
>
> Field work remarks are typically processed by Service Order Management; therefore, field activity request remarks are not required.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Enter a unique **Field Activity Remark** and a **Description** for every field activity request remark.

Turn on **Eligible for Processing** if field activity requests marked with a given remark code should cause one or more algorithm to execute.

The grid contains **Algorithms** associated with the field activity request remark. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Field Activity Remark Activation	Optional	These algorithms are executed when there are pending FA remarks linked to a field activity request and the FACT (Field activity remark activation) background process runs. Refer to Field Activity - Characteristics/Remarks for more information.

Setting Up Outage Call Types

When you create an outage call, you must supply an outage call type. Outage call types contain a great deal of information that is defaulted onto the outage call, including the outage category group codes or trouble codes. To set up outage call types, open **Admin**, and then **Field Work**, and then **Outage Call Type**, and then **Add**.

NOT_SUPPORTED

Refer to The Big Picture of Outage Care Integration for more information about how trouble calls are created and sent to NMS

The topics in this section describe the base-package zones that appear on the Outage Call Type portal.

- Outage Call Type List
- Outage Call Type

Outage Call Type List

Copyright © 2026, Oracle and/or its affiliates.

The Outage Call Type List zone lists every outage call type, i.e. every service task type that has a service task type class of Outage Call. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent outage call type.
- Click the Add link in the zone's title bar to add a new outage call type.

Outage Call Type

The Outage Call Type zone contains display-only information about an outage call type. This zone appears when an outage call type has been broadcast from the Outage Call Type List zone or if this portal is opened via a drill down from another page. The following functions are available:

- **Edit**: to start a business process that updates the outage call type.
- **Delete**: start a business process that deletes the outage call type.
- **Duplicate**: to start a business process that duplicates the outage call type.

Please see the zone's help text for information about this zone's fields.

Setting Up Service Call Communication Types

A Service Call Communication Type defines the rules and controls for creating service call communications.

Refer to Asset Management System Integration for more information.

To set up service call communication types, open **Admin**, and then **Field Work**, and then **Service Call Communication Type**, and then **Search**.

This is a standard All-In-One Portal.

The information captured on the service call communication type depends on the business objects that your implementation supports. Refer to the embedded help text for more information.

Setting Up Service Categories

A service category represents a broad category of services.

Refer to Asset Management System Integration for more information.

To set up service categories, open **Admin**, and then **Field Work**, and then **Service Category — CCB**, and then **Search**.

This is a standard All-In-One Portal.

The information captured on the service category depends on the business objects that your implementation supports. Refer to the embedded help text for more information.

Setting Up Service Codes

Service Codes are used to create the appropriate work record from the service call

Copyright © 2026, Oracle and/or its affiliates.

Refer to Asset Management System Integration for more information.

To set up service codes, open **Admin**, and then **Field Work**, and then **Service Code — CCB**, and then **Search**.

This is a standard All-In-One Portal.

The information captured on the service code depends on the business objects that your implementation supports. Refer to the embedded help text for more information.

Service Order Management External Applications

The external systems used with Service Order Management must be defined as External Applications using the "External Application" (D1-ExternalApplication) business object. Examples of external systems can include:

- A customer information system (such as a separate installation of Oracle Utilities Customer Care and Billing)
- A field work system (such as Oracle Utilities Mobile Workforce Management)
- An asset management system (such as Oracle Utilities Operational Device Management or Oracle Utilities Work and Asset Management)

Information defined for external system service providers used by Service Order Management include:

- **Our Name/ID in Their System**: This is the value that the field work system uses to identify our system.
- **Utility Device ID Type**: This is the Device ID Type that will be used when communicating with the external application and it will be the assumed Device ID Type for any device identifiers sent from the external application.
- **Utility Service Point ID Type**: This is the Service Point ID Type that will be used when communicating with the external application and it will be the assumed Service Point ID Type for any service point identifiers sent from the external application.

Refer to Understanding External Applications and Configuring External Applications for more information about external applications.

Processing Roles

The external application’s processing roles define how data relevant to the field work system is sent and/or created.

Field work service providers can use the following processing roles:

- **Activity Notification**: Used to send notifications to subscribing and/or requesting systems about the status of orchestration and/or service order field activities.
- **Appointment Request**: Used to send a request for an appointment to the field work system.
- **Cancelation Activity**: Used to send notifications to requesting systems when canceling orchestration and/or service order field activities.
- **Collection Details**: Used to retrieve details about collections processing (used with "Cut Service for Non-Payment" and "Restore Service for Payment" orchestration activities).
- **Customer Contact**: Used to send a contact to a customer regarding a service request
- **Field Activity**: Used to send a service order field activity to the field work system.

Copyright © 2026, Oracle and/or its affiliates.

- **Field Activity Completion**: Used to send a notification regarding completion of a service order field activity.
- **Interim Status Update**: Used to send updates regarding the status of orchestration and service order field activities to requesting systems.
- **Meter Exchange Mapping**: Used to define how to define different types of meter exchanges based specific roles and device configurations. This can provide context to field crews to help ensure they install the correct type of device and device configuration when exchanging a meter.
- **Response - Appointment**: Used to send a request for an appointment to the field work system.
- **Response - Fail**: Used to send a response to an external system when Service Order Management fails to respond.
- **Response - Missed Appointment**: Used to send a response to the field work system when notification of a missed appointment is received.
- **Response - Negative Acknowledgement**: Used to send a negative acknowledgement response to an external system in the event that a request is rejected.
- **Response - Received**: Used to send a response to an external system to acknowledge receipt of a request.
- **Response - Success**: Used to send a response to an external system when Service Order Management successfully processes a request.
- **Send Field Activity Remark**: Used to send a service order field activity remark to a subscribing system
- **Update Activity**: Used to send notifications to requesting systems when updating orchestration and/or service order field activities.

Setting Up Service Order Activities

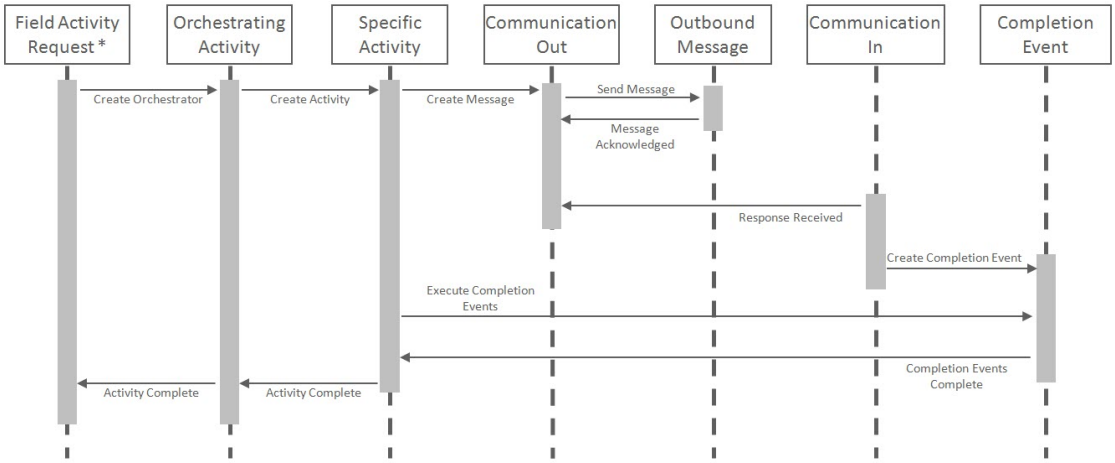
- Understanding Service Order Activities
- Service Order Activity Processing
- Service Order Activity Algorithm Types
- Understanding Service Order Field Activity Types
- Configuring Service Order Activity Types

Understanding Service Order Activities

This section describes service order activities and how they manage the service order process.

Service order activities coordinate a large number of child transactions that represent the communication to and from an external application. The below diagram depicts a service order activity:

Copyright © 2026, Oracle and/or its affiliates.



Each object in the sequence diagram has a distinct set of duties within the context of the communication:

- **Orchestrating Activity**: controls the overall intent of the communication. For example an "Enable Service" could include initiating one-to-many specific activities to install the meter and begin the flow of the metered commodity to the service point.
- **Specific Activity**: can be initiated from an orchestrating activity or directly. These activities represent a single task to be carried out such as installing a meter or a remote disconnect smart meter command.
- **Communication Out**: orchestrates the communication to the external application and provides robust handling for any errors that might occur during that communication.
- **Outbound Message**: represents the message payload sent to the external system and the synchronous response.
- **Communication In**: orchestrates the handling of an asynchronous or unsolicited response from an external system.
- **Completion Event**: carries out the results of the communication. For example, in the case of a remote connect it would create the appropriate on off history entry for the device's installation event.

The base package provides the following types of service order activities:

- **Enable Service**: Used to enable service at a service point.
- **Disable Service**: Used to disable service at a service point.
- **Cut for Non-Payment**: Used to cut off service at a service point due to non-payment of past due amounts.
- **Reconnect Service for Payment**: Used to restore service at a service point after receipt of past due payment.
- **Exchange Meter**: Used to orchestrate the exchange of meters at service point, such as in the event that a customer upgrades their meter.
- **Back to Back Service**: Used to orchestrate a change of service when the customer at a service point changes (such as when owners/tenants change).

Service Order Activity Processing

Copyright © 2026, Oracle and/or its affiliates.

To understand how service order activities manage the service order process, it’s important understand the lifecycle of orchestration activity business objects.

Service Order Orchestration Activity Lifecycle

All service order orchestration activity business objects share a common parent business object that defines their lifecycles. This is the Service Point Activity Orchestration business object (D1-SPActivityOrchestration). The table below outlines the lifecycle for this business object.

State	Description
Pending	The initial state for orchestration activities. An Enter algorithm sends an acknowledgement to the requesting system. The activity is transitioned to the next state via a monitor process.
Validate	Enter algorithms perform the following: Validate Activity Type Validate Service Point Check for a non-final duplicate service order request activity for the same service point.
Validation Error	If the business object fails any of the validations in the Pending state, it enters this state. Activities in this state can be corrected and retried.
Discarded	Activities discarded in other states enter this state. Enter algorithms perform the following: Validate that non-final child activities can be discarded without the need for a cancel activity Cancel non-final child activities Send a failure notification to the requesting system
Waiting for Effective Date	If an orchestration activity has a future effective date, it remains in this state until the effective date is reached. A Monitor algorithm transitions the activity to the next state when the activity's effective date time is reached (process date time >= effective date time).
Are SP and Device Ready?	Each type of orchestration activity business object has a unique set of Enter algorithms that perform operations as appropriate for the type of service order request. See Service Order Activity Algorithm Types for more information about these algorithms.
Activity in Progress	Orchestration activities remain in this state while their child activities are processed. A Monitor algorithm transitions the activity to the "Are the SP and Device Ready?” state if there are no non-final child activities related to the current activity. A Monitor algorithm validates that the orchestration activity has not been in its current state for too long, based on the Expiration Days parameter on the orchestration activity’s type and the Expiration Date/Time on the orchestration activity An Exit algorithm resets the Expiration Date/Time on the orchestration activity such that each time the activity exits this state its Expiration Date/Time is updated.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Activity Error	If one or more child activities enters an Error state, the orchestration activity enters this state. Activities in this state can be corrected and retried.
Retry	When an orchestration activity is retried after correction of an error condition, it enters this state. Enter algorithms perform the following: Check to determine if there are child service order field activities in progress that have outbound communications awaiting a response. Transition any non-final child activity to the "Reject" state (the state defined as "Reject" on the child activity business object lifecycle. This is most often the "Discarded" state).
Completed	Orchestration activities enter this state when all child activities have successfully completed. An Enter algorithm send a success notification to the requesting system.

Use the Business Object and Algorithm portals to view additional details about this business object and its lifecycle algorithms.

Cancel / Update Orchestration Lifecycle

The Cancel Orchestration (D1-CancelOrchestration) and Update Orchestration (D1-UpdateOrchestration) business objects have a similar lifecycle, with the following exceptions:

- There is no "Waiting for Effective Date" state.
- In place of the "Are SP and Device Ready?" state, they have "Cancel Specific Activity" / "Update Specific Activity" states. Enter algorithms on these states attempt to cancel or update a specific child activity.
- In the place of "Activity In Progress" and "Activity Error" states, they have "Communication in Progress" and "Communication in Error" states.

Use the Business Object and Algorithm portals to view additional details about this business object and its lifecycle algorithms.

Service Order Activity Algorithm Types

When an orchestration activity enters the "Are SP and Device Ready?" state, a set of Enter algorithms are used to evaluate the state of the service point / meter / item to determine which actions are required to complete the service request. These algorithms are based on the following algorithm types.

Customer-Device Compatibility Check (D1-DVCOMCHK)

Algorithms of this type execute the "Customer-Device Compatibility Algorithm" defined on the orchestration activity's activity type. Algorithms of this type uses the following parameters:

- **Activity BO To Create If Compatibility Detected**: Specifies the activity business object to instantiate if the algorithm detects an in compatibility between the customer/service point and device.

Note: The base package does not include algorithm types for the "Activity Type - Customer Device Compatibility" algorithm entity.

Copyright © 2026, Oracle and/or its affiliates.

Connect Only If Previously Connected (D1-CONPRECON)

Algorithms of this type check if the "Connect New Device" flag has a value. If the flag is not populated, the algorithm sets the value of the flag based on the connection status of the device prior to the meter exchange (used only with Meter Exchange requests).

For additional information, refer to the D1-CONPRECON Algorithm Type.

Create Meter Exchange Field Activity (D1-CREMTREXC)

Algorithms of this type create a service order field activity based on details provided in the algorithm's parameters. Algorithms of this type uses the following parameters:

- **Activity and Specific Field Task to Create**: Specifies the type of activity business object and field task type to create for meter exchange service order field activities, as defined by the following mnemonics:

Mnemonic	Description
activityBOToCreate	Specifies the activity business object to create.
specificFieldTask	Specifies the Field Task Type when creating a service order field activity. This value comes from the Field Task Type extendable lookup.

For example, to create a service order field activity based on the D1-FieldActivity business object and the Exchange Meter field task type, these mnemonics would be configured as follows:

```
activityBOToCreate=D1-FieldActivity specificFieldTask=D1-ExchangeMeter
```

For additional information, refer to the D1-CREMTREXC Algorithm Type.

Decommission Removed Meter (D1-DCRMMTR)

Algorithms of this type create a decommissioning command for a removed meter (used only with Meter Exchange requests). Algorithms of this type uses the following parameters:

- **Decommission Activity BO to be created**: Specifies the type of activity business object to create when decommissioning a meter. The specific activity is created, as defined by the following mnemonics:

Mnemonic	Description
activityBOToCreate	Specifies the activity business object to create.

For example, to create a smart meter command activity based on the D1-DeviceDecommission business object, this parameter would be configured as follows:

```
activityBOToCreate=D1-DeviceDecommission
```

For additional information, refer to the D1-DCRMMTR Algorithm Type.

Copyright © 2026, Oracle and/or its affiliates.

Create Specific Activity (D1-CRSPACT)

Algorithms of this type determine if a specific activity needs to be created based on the state of the service point. Algorithms of this type use the following parameters to specify the conditions and activity to be created:

- **Field Activity BO**: Specifies the activity business object to instantiate if the algorithm creates a field task type (see next parameter).
- **SP State and Activity BO to Create:** Specifies the type of activity business object to create based on the state of the service point. This parameter can be repeated up to 20 times. Instances of the parameter are evaluated one at a time and the first condition matching the state of the service point is used. Parameters should be ordered from the most restrictive condition to the least restrictive. This parameters uses the following mnemonics to indicate the state (any combination of the following) of the service point:

Mnemonic	Description
servicePointConnected	Specifies if the service point is currently connected Valid values are “true" and "false".
disconnectLocation	Specifies the “Disconnect Location” for the service point. Valid values are “D1SR" (source) and "D1DV" (device).
deviceInstalledAtSP	Specifies if there is a device currently installed at the service point. Valid values are “true" and "false".
installationEventStatusOverride	Specifies the value of the "Installation Status" option type of the Install Event's Status ("Pending", "Conn-PreComm", "ManualOff", etc.)

Based on the unique combination of these mnemonics, a specific activity is created, as defined by the following mnemonics:

Mnemonic	Description
activityBOToCreate	Specifies the activity business object to create (used most often to specify a command business object)
specificFieldTask	Specifies the field task type when creating a service order field activity. This value comes from the Field Task Type extendable lookup. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the activity business object to create.
spTypeCategory	Specifies a service point type category. Valid values in include "D1MT" (meter), "D1IT" (item), "D1MI" (multi-item), from the SP_CATEGORY_FLG lookup. Specifying this mnemonic indicates that a service order field activity should be created only if the service point’s category match the one specified by this mnemonic.
executeOverrideAlgorithm	Specifies whether or not to execute the algorithm specified for the **Override Device/Task Algorithm** on the service order orchestration activity type. This allows the activity business object to be dynamically determined based on an algorithm instead of the "activityBOToCreate" or "specificFieldTask" mnemonics. Valid values are “true" and "false".

Copyright © 2026, Oracle and/or its affiliates.

For example, the following parameter configuration would create a "Connect Service Point and Install Meter" service order field activity given the following conditions:

- Service Point Connected: False
- Disconnect Location: Source
- Device Installed at Service Point: False
- Service Point Category: Meter

```
servicePointConnected=false disconnectLocation=D1SR deviceInstalledAtSP=false
specificFieldTask=D1-ConnSPAtSrceAndInstMtr spTypeCategory=D1MT
```

For additional information, refer to the D1-CRSPACT Algorithm Type.

Update Device (D1-UPDDVC)

Algorithms of this type determine if an activity needs to be created to update the device based on the state of the service point and device installed at the service point. Algorithms of this type use the following parameters to specify the conditions and activity to be created.

- **Error if SP Not Connected or no Device Installed (Default is Yes)**: Indicates if the algorithm should return an error if the service point is not connected or if a device is not currently installed. Valid values are "Yes" and "Con" (continue)
- **SP State and Activity BO to Create**: Specifies the type of activity business object to create based on the state of the service point. This parameter can be repeated up to 20 times. Instances of the parameter are evaluated one at a time and the first condition matching the state of the service point is used. Parameters should be ordered from the most restrictive condition to the least restrictive. These parameters use the following mnemonic to indicate the state (any combination of the following) of the service point:

Mnemonic	Description
installationEventStatusOverride	Specifies the value of the "Installation Status" option type of the Install Event's Status ("Pending", "Conn-PreComm", "ManualOff", etc.)

Based on the value of this mnemonic, a specific activity is created, as defined by the following mnemonics:

Mnemonic	Description
activityBOToCreate	Specifies the activity business object to create (used most often to specify a command business object)
specificFieldTask	Specifies the Field Task Type when creating a service order field activity. This value comes from the Field Task Type extendable lookup. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the activity business object to create.
spTypeCategory	Specifies a service point type category. Valid values in include "D1MT" (meter), "D1IT" (item), "D1MI" (multi-item), from the SP_CATEGORY_FLG lookup. Specifying this mnemonic indicates that a service order field activity should be created only if the service point's category match the one specified by this mnemonic.

Copyright © 2026, Oracle and/or its affiliates.

Mnemonic	Description
alternativeFieldTask	Specifies an alternative Field Task Type to use when creating a service order field activity in the event that the device does not support the command indicated by the "activityBOToCreate" mnemonic. Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the activity business object to create. A value of 'skip' will continue the evaluation of the algorithm's next parameter

For example, the following parameter configuration would create a "Turn On Meter" service order field activity given the following conditions:

- Installation Status: Manual Off
- Service Point Category: Meter

```
installEventStatusOverride=ManualOff specificFieldTask=D1-TurnOnMeter
spTypeCategory=D1MT
```

Other parameters used by algorithms of this type include:

- **Field Activity BO**: Specifies the activity business object to instantiate if the algorithm creates a field task type (see above parameter).
- **XPath of Activity Element controlling Activity creation**: Defines an element within the activity business object schema that can be used to control whether or not this algorithm should create an activity. For example, to specify that the value of the "Connect New Device" flag be used to determine whether or not the algorithm should create an activity, this parameter could be set to "connectNewDevice".
- **Element value indication that Activity creation should not proceed**: Specifies a value for the element defined in the "XPath of Activity Element controlling Activity creation" parameter that would indicate that the algorithm should not create an activity. Valid values are based on the element defined for the "XPath of Activity Element controlling Activity creation" parameter. For example, to specify that an activity should not be created if the "Connect New Device" flag is set to "Do Not Connect / Turn On", this parameter should be set to "D1NC" (from the D1_CONNECT_NEW_DEVICE_FLG lookup).

For additional information, refer to the D1-UPDDVC Algorithm Type.

Remote Turn Off Turn On (D1-REMONOFF)

Algorithms of this type remotely turn a device off and on for a Back to Back service request. Algorithms of this type use the following parameters:

- **Device Incompatibility Detected Activity BO**: Specifies the activity business object the algorithm will look for. If the algorithm finds an activity of this business object, the algorithm terminates.
- **Remote Connect BO**: Specifies the activity business object to instantiate when creating a remote connect command.
- **Remote Disconnect BO**: Specifies the activity business object to instantiate when creating a remote disconnect command.
- **Installation Event Status Override for Connect**: The override status to which the Installation Event Status is set after performing a remote connect command.

Copyright © 2026, Oracle and/or its affiliates.

- **Installation Event Status Override for Disconnect**: The override status to which the Installation Event Status is set after performing a remote disconnect command.

For additional information, refer to the D1-REMONOFF Algorithm Type.

Check for Measurement (D1-CHKMSMT)

Algorithms of this type determine if measurements exist on activity's service point as of the service date/time. If no measurement is found, algorithms of this type create an activity to either obtain or wait for a measurement. The specific type of activity is based on the type and configuration of the device and service point. Algorithms of this type use the following parameters:

- **Activity BO To Wait For Measurement:** Specifies the activity business object to instantiate when the algorithm logic indicates it should wait for a measurement for the service point.
- **Activity BO For Field Read**: Specifies the field activity business object to instantiate when the algorithm logic indicates it should request a meter reading from the field.
- **Specific Field Task**: Specifies the field task type when creating a service order field activity for a meter reading from the field.
- **Activity BO To Wait For Scheduled Read**: Specifies the activity business object to instantiate when the algorithm logic indicates it should wait for a scheduled read for the service point.
- **Activity BO For On Demand Read - Scalar**: Specifies the activity business object to instantiate when the algorithm logic indicates it should issue an on-demand read (scalar) smart meter command.
- **Start Range for Normal Measurement Condition**: The start of the range of conditions that indicate "normal" measurements when the algorithm is searching for measurements for the service point.
- **End Range for Normal Measurement Condition**: The end of the range of conditions that indicate "normal" measurements when the algorithm is searching for measurements for the service point.
- **Minimum Range for bottom Measurement condition**: The minimum measurement condition used when searching for measurements for the service point. Used only when no measurements are found in the "normal" range defined by the "Start/End Range Normal Measurement Condition" parameters.

The following parameters on the orchestration activity type are also used by algorithms of this type when searching for measurements for the service point:

- **Look for Measurement within the Day**: Limits the search to the reference date (the service date).
- **Minimum and Maximum Offset Number of Days**: Numbers of days added to /subtracted from the reference date to expand the search period.

For additional information, refer to the D1-CHKMSMT Algorithm Type.

Algorithm Types and Orchestration Activity Business Objects

Each of the orchestration activity business objects uses a different set of these algorithm types. The table below lists which of these algorithm types are defined for each of the service order orchestration activity business objects.

Copyright © 2026, Oracle and/or its affiliates.

	Enable Service	Disable Service	Cut for Non-Payment	Reconnect for Payment	Meter Exchange	Back-to-Back
Customer-Device Compatibility Check	X					X
Connect Only If Previously Connected					X	
Create Specific Activity	X					
Create Meter Exchange Field Activity					X	
Update Device	X	X	X	X	X	X
Remote Turn Off Turn On						X
Decommission Removed Meter					X	
Check For Measurement:	X	X	X			X

Cancel / Update Orchestration - Algorithm Types

Enter algorithms on the "Cancel Specific Activity" and "Update Specific Activity" states attempt to cancel or update a specific child activity. These algorithms are based on the following algorithm types.

- **Cancel Specific Activity**: Algorithms of this type cancel the specific activity (either a service order field activity or a smart meter command) that is associated to the Cancel or Update orchestration activity, based on the current status of the specific activity.
- **Update Specific Activity**: Algorithms of this type update the specific activity (either a service order field activity or a smart meter command) that is associated to the Cancel or Update orchestration activity, based on the current status of the specific activity.

Algorithm Type	Algorithm(s)
Cancel Specific Activity (D1-CANSPACT)	Cancel Specific Activity (D1-CANSPACT)
Update Specific Activity (D1-UPDSPAC)	Update Specific Activity (D1-UPDSPAC)

Use the Algorithm Type and Algorithm portals to view additional details about these algorithms.

Understanding Service Order Field Activity Types

Service order activity types must be configured for each type of service order activity.

Service order activity types are assigned to the following Activity Type Categories:

Service Order Activity	Activity Type Category
Back-to-Back	Service Request Orchestration

Copyright © 2026, Oracle and/or its affiliates.

Cancel Orchestration	Orchestration Maintenance
Cut Service for Non-Payment	Request Orchestration
Disable Service	Request Orchestration
Enable Service	Request Orchestration
Exchange Meter	Request Orchestration
Reconnect Service for Payment	Request Orchestration
Update Orchestration	Orchestration Maintenance

Refer to Understanding Activity Types for more information about activity types.

Configuring Service Order Activity Types

You use the **SOM Activity Type** portal is to view and maintain service order activity types.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order activity types as well as service order activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List:** This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type:** This zone provides information about the selected Activity Type
- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the “.Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

Service Order Management Configuration

The following table outlines the activity types that must be configured for the service order activity types supported by Service Order Management:

Service Order Activity	Activity Type (Business Object)
Enable Service	Enable Service (D1–EnableServiceType)
Disable Service	Disable Service (D1–DisableServiceType)
Cut Service for Non-Payment	Cut Service for Non-Payment (D1-CutServiceForNonPaymentType)
Reconnect Service for Payment	Reconnect for Payment (D1–ReconnectForPaymentType)
Meter Exchange	Exchange Meter (D1–ExchangeMeterType)
Back-to-Back Service	Back-to-Back Service (D1–BackToBackServiceType)
Cancel Orchestration	Cancel Orchestration (D1–CancelOrchestrationType)
Update Orchestration	Update Orchestration (D1–UpdateOrchestrationType)

The demonstration database contains examples of each of these service order activity types.

Setting Up Service Order Field Activities

- Understanding Service Order Field Activities
- How Do Service Order Field Activities Work?
- Service Order Field Activity Processing
- Service Order Field Activity Communication
- Unrelated Pickup Orders

Copyright © 2026, Oracle and/or its affiliates.

Understanding Service Order Field Activities

This section describes service order field activities and how they communicate with field work management systems.

Service order field activities are activities that involve sending workers into the field to perform service. This can include meter installation, meter replacement, and other activities.

Service order field activities send messages to a field work system, which in turn assigns them to crews to be completed in the field.

Field Activity Information

All service order field activities are based on the Field Activity (D1-FieldActivity) business object, and include the following user-accessible information:

- **Status**: The current status of the service order field activity.
- **Service Date/Time**: The date and time the service order field activity was created.
- **Service Point**: The service point associated with the service order field activity.
- **Field Task Type**: The field task type for the service order field activity. This defines the type of task and other processing details regarding how Service Order Management processes the service order field activity. See Field Task Types for more details about field task types.
- **Recipient**: The field work system service provider to which the service order field activity is sent for scheduling and assignment.
- **Device ID**: The device related to the service order field activity (if applicable).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.
- **Address Information**: The address of the service point associated with the service order field activity.

The field activity business object also contains other information that is populated by algorithms and scripts as the service order field activity is processed by the system.

How Do Service Order Field Activities Work?

At a high level, service order field activities work as follows:

Create Field Activity

A service order orchestration activity creates a service order field activity based on the current state of the service point/meter/item.

Retrieve Required Data

The service order field activity uses a set of pre-processing algorithms to derive and populate data needed by the activity, such as the device, service point, address, effective date, and others.

Copyright © 2026, Oracle and/or its affiliates.

Request Appointment (Optional)

If the service order field activity task type specifies that field tasks of this type require an appointment, the service order field activity checks for available appointment slots in the field work system and sends a notification to the appointment handling system.

Create Outbound Communication

The service order field activity creates an outbound communication to send the service order field activity to the field work system. The outbound communication gathers the information required by the field work system before being sent. This information is retrieved by a set of processing scripts defined on the field task type.

Receive Inbound Communication

When the service order field activity has been completed, the field work system sends an inbound communication back to Service Order Management.

Inbound communications can contain Field Activity Remarks (entered by field resources when they perform and complete their field work. If the Field Activity Remarks reference completion events, they are executed.

The inbound communications create completion events as defined on the field task type. If the service order field activity was successfully completed, it creates the "Completion Events When Successful" completion events. If the service order field activity was canceled, it creates the "Completion Events When Canceled" completion events.

Execute Completion Events

After receiving the inbound communication, a service order field activity algorithm transitions any active completion events into their executed state.

Complete Processing

The service order field activity completes its processing by doing the following:

- Updating the parent orchestration activity
- Sending a success response to the requester
- Transitioning the parent orchestration activity to the next state in its lifecycle
- Sending a service order field activity completion outbound communication to subscribing systems.

Service Order Field Activity Processing

This section outlines how service order field activities are processed.

Pre-Processing, Validation, and Post-Processing Algorithms

When service order field activities are first instantiated, a set of pre-processing algorithms populate and derive information needed for the activity, such as the activity type, service point, device, address, effective date, and other information.

Validation algorithms validate this information when first retrieved and when updated.

When service order field activities are completed, a post-processing algorithm populates the activity end date/time.

Copyright © 2026, Oracle and/or its affiliates.

Service Order Field Activity Lifecycle

As a service order field activity moves through its lifecycle, it triggers various business processes based on the type of service order field activity. The table below outlines the lifecycle for the Field Activity (D1-FieldActivity) business object.

State	Description
Pending	The initial state for service order field activities. An Enter algorithm sends an acknowledgement to the requesting system. The activity is transitioned to the next state via a monitor process.
Validate	Enter algorithms perform the following: Validate Activity Type (and transition to error if invalid) Derive and validate service order field activity recipient Validate duplicate and conflict service order field activities Derive and validate service order field activity service point Derive and validate service order field activity device Validate address constituents Check for any existing cut service restrictions The activity is transitioned to the next state via a monitor process.
Validation Error	If the business object fails any of the validations in the Pending state, it enters this state. Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. Activities in this state can be corrected and retried.
Waiting to Request	If a service order field activity has a future effective date, it remains in this state until the effective date is reached. A Monitor algorithm transitions the activity to the next state when the activity's effective date time is reached (process date time >= effective date time). An Enter algorithm sets the "Allow Child to Transition Parent Activity Based On Effective Date" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Waiting for Appointment	If the service order field activity passes its validations and the effective date has been reached, the activity enters this state. Enter algorithms perform the following: Evaluate if an appointment is required for field tasks of this type. If not, the activity transitions to the "Communication in Progress" state. Create a To Do if an appointment is necessary but the system is not able to send an appointment request. Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. Send a notification to the appointment handling system. Monitor algorithms perform the following: Verify if an appointment has been supplied. Send a notification to the appointment handling system. The activity is transitioned to the next state via a monitor process. See Waiting for Appointment for more information about this state.
Communication in Progress	Service order field activities enter this state following the "Waiting for Appointment" or "Retry" states. Enter algorithms perform the following: Create an outbound communication for the service order field activity (see Communication in Progress for more information). Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. Monitor algorithms perform the following: Check for existing child communications. Check that the activity hasn’t timed out.
Discarded	Activities discarded in other states enter this state. Enter algorithms perform the following: Cancel outstanding outbound communications. Cancel outstanding completion events. Populate the cancel reason. Send a failure notification to the requesting system. Transition the parent activity to the "Activity Error" state (see Service Order Orchestration Activity Lifecycle for more information). Check if a Cancel Orchestration activity is required.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Communication Error	If an outbound or inbound communication an Error state, the service order field activity enters this state. Monitor algorithms perform the following: Check that the activity hasn’t timed out. Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role. Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. Activities in this state can be corrected and retried.
Retry	When a service order field activity is retried after correction of an error condition, it enters this state. Enter algorithms perform the following: Check to determine if there are associated outbound communications in progress. Cancel any outstanding outbound communications.
Execute Completion Events	After an inbound communication is received, it enters this state. Enter algorithms perform the following: Executes completion events defined on the field task type (these completion events were initially created by the inbound communication). Evaluates the "Field Activity Completed" flag on the service order field activity. If this is set to "No", the service order field activity is transitioned to the "Canceled In Field" state. The activity is transitioned to the next state via a monitor process. See Execute Completion Events for more information about this state.
Completion Event Error	If an error occurs during completion event processing, the service order field activity enters this state. Monitor algorithms perform the following: Check that the activity hasn’t timed out. Enter algorithms perform the following: Create a To Do based on specified To Do Type and To Do Role. Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. Activities in this state can be corrected and retried.
Completed	Service order field activities enter this state when all completion events have successfully completed. Enter algorithms perform the following: Update the parent orchestration activity. Send a success response to the requester. Transition the parent orchestration activity to the next state in its lifecycle. Send a service order field activity completion outbound communication to subscribing systems.

Copyright © 2026, Oracle and/or its affiliates.

State	Description
Canceled in Field	If the "Field Activity Completed" flag on the field activity is set to "No", the service order field activity enter this state. Enter algorithms perform the following: Send a failed response to the requester. Transition the parent orchestration activity to the "Activity Error" state. Create a To Do to notify users that the service order field activity has been canceled.

Waiting for Appointment

When a service order field activity enters the "Wait for Appointment" state, it first determines if an appointment is necessary for the service order field activity. If not, the activity moves on to the "Communication in Progress" state (see below).

If an appointment request cannot be sent for some reason, the service order field activity creates a To Do item to alert a user to attempt to manually request an appointment. Otherwise, the service order field activity sends an outbound message to the field work system requesting an appointment. based on the appropriate processing role defined on the."Send Notification to Appointment Handling System - Enter" algorithm.

Processing Role	Outbound Communication Business Object
Appointment Response (default) Used if: An appointment is required and needs to be scheduled Appointment has been set	Send Appointment Response Outbound Message (D1-SendApptRespOutboundMsg) Note: An outbound message must be created based on this business object.

The response from the field work system can be received by creating an Inbound Web Service that references the "Book selected appointment to Field Activity" (D1-BookAppt) service script.

While in this state, monitor algorithms verify if an appointment has been supplied and send notifications to the field work system.

Communication in Progress

Service order field activity communications are records of messages sent between Service Order Management and an external field work system. Communications can flow both outbound and inbound.

See Service Order Field Activity Communication for more information about service order field activity communication.

Manually Completing Service Order Field Activities

Service order field activities in the "Communication in Progress" state can be manually completed or canceled by clicking **Complete** in the **Record Actions** section of the **Activity** zone. This opens the **Field Activity Manual Completion** page where the user can enter appropriate details to manually complete the field activity (such as completion information, service point information, device information, and so on). When submitted, this page creates an inbound communication and transitions the field activity to the next state as appropriate.

Copyright © 2026, Oracle and/or its affiliates.

Note: Implementations can define custom service scripts for each field task type in the **Manual FA Completion Controls** section on the Field Task Type extendable lookup to customize the sections displayed on the **Field Activity Manual Completion** page.

Execute Completion Events

After receiving the inbound communication, the service order field activity enters the "Execute Completion Events" state.

The inbound communication will have previously created completion events for the service order field activity, based on those defined on the field task type or those referenced by field activity remarks. These creation events begin in the "Pending" state.

An Enter algorithm transitions completion events associated with the service order field activity into their "Executed" state.

Service Order Field Activity Communication

This section outlines how service order field activities communicate with field work systems.

When a service order field activity enters the "Communication in Progress" state, it sends an outbound communication to the field work system, and waits for an inbound communication response.

See **Understanding the Service Order Field Activity Communication Process** below for more information about the role of communications in the service order field activity communication process.

Outbound Communications

Outbound Communications represent messages sent from Service Order Management to. an external field work system Outbound communications use the following types of objects:

Outbound Communication Business Objects

An outbound communication business object exists for each type of message to be sent to an external system. For service order field activities, the following base package outbound communication objects can be used.

Type of Outbound Communication	Outbound Communication Business Object
Initial service order field activity outbound communication	Field Activity Outbound Communication (D1-FieldActivityOBComm)
Modify outbound communication Used to send an update to a service order field activity previously sent to the field work system.	Field Activity Outbound Communication (D1-ActivityModifyOBComm)

Outbound Message Types

A outbound message type must also be created for each type of message to be sent to an external system. Again, this is based on the types of messages the system is designed to accept. For service order field activities, the following outbound message types are needed:

Copyright © 2026, Oracle and/or its affiliates.

Type of Outbound Communication	Outbound Message Type
Initial Service Order Field Activity Message	Field Activity Outbound Message
Modify Existing Service Order Field Activity	Modify Field Activity Outbound Message

Refer to the Oracle Utilities Application Framework documentation for more information about outbound message types.

External Systems

You must also create an External System for each external system to which Service Order Management will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch, XAI, or Real-time)
- Message Sender (if Processing Method is set to Real-time or XAI)
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

To continue the example above, you might create the following external system:

External Application		
Outbound Message Type	**Processing Method**	**Batch Control**
Field Activity Outbound Message	Batch	Sync Request Monitor (F1-SYNRQ)
Modify Field Activity Outbound Message	Batch	Sync Request Monitor (F1-SYNRQ)

Refer to the Oracle Utilities Application Framework documentation for more information about external systems.

Inbound Communications

Inbound Communications represent messages sent from an external field work system to Service Order Management. Inbound communications are typically sent in response to a service order field activity. Inbound communications use the following types of objects:

Inbound Communication Business Objects

An inbound communication business object must be created for each type of message to be received from an external system. For service order field activities, the following base package inbound communication object can be used.

Inbound Communication Business Object
Field Activity Inbound Communication (D1-FieldActivityIBComm)

Inbound Web Service

Copyright © 2026, Oracle and/or its affiliates.

You must also create an Inbound Web Service for each type of message to be received from an external system. Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send. To continue the example above, you might create the following inbound web services:

Inbound Web Service	Schema (Inbound Communication Business Object)
Field Activity Inbound Communication	Field Activity Inbound Communication D1-FieldActivityIBComm

Refer to the Oracle Utilities Application Framework documentation for more information about Inbound Web Services.

Field Activity Remarks

Inbound communications can contain activity remarks, which represent notes entered by the field worker as they perform and complete their field work. These can be solely informational, or can reference completion events via the "Remark Processing" section of the Field Activity Remark Type extendable lookup. This allows information sent with the inbound communication to initiate business processing if necessary.

Completion events specified on this extendable lookup are created by the inbound communication, and then executed when the service order field activity enters the "Execute Completion Events" state.

Understanding the Service Order Field Activity Communication Process

This section provides an overview of the communication process that takes place when a service order field activity is initiated. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific base package objects used by Service Order Management

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Base Package Objects
1.	An orchestration activity creates a service order field activity as part of its processing. A service order field activity business object is instantiated for the command.	Field Activity Business Object: Field Activity (D1-FieldActivity)
2.	When the service order field activity enters the Communication in Progress tate, it creates an outbound communication.	Outbound Communication Business Object: Field Activity Outbound Communication (D1-FieldActivityOBComm)
3.	A Enter algorithm on the "Awaiting Response" state of the outbound communication retrieves information needed by the outbound message to be sent to the field work system based on processing scripts specified on the field task type.	Enter Algorithm: Populate Send Detail for Field Activity (D1-POPSNDDTL)

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Base Package Objects
4.	A Enter algorithm on the "Awaiting Response" state of the outbound communication creates an outbound message.	Enter Algorithm: Create Outbound Message (D1-COUTMSG) Note: An outbound message type for this message is not included in the base package.
5.	The outbound message is sent to middleware components via an External System and Batch Control. Middleware components utilize Business Process Execution Language (BPEL).	External System: MWM Batch Control: Sync Request Monitor (F1-SYNRQ)
6.	The middleware converts the outbound message from SOM format into the format used by the field work system, and sends the message to the field work system.	
7.	When the field work system sends a response, the middleware receives the response message from the field work system, and converts it from the format used by the field work system to SOM format and invokes an Inbound Web Service.	Inbound Web Service: D1-FieldActivityIBComm
8.	The Inbound Web Service picks up the message, and creates a corresponding inbound communication. The specific type of inbound communication business object created is determined by the Inbound Web Service.	Inbound Web Service: D1-FieldActivityIBComm Inbound Communication Business Object: Field Activity Inbound Communication (D1-FieldActivityIBComm)
9.	The inbound communication identifies the parent outbound communication.	Outbound Communication Business Object: Field Activity Outbound Communication (D1-FieldActivityOBComm)
10.	The inbound communication creates the completion events defined on the field activity field task type (Successful or Canceled, as appropriate) in the "Pending" state. If the inbound communication contains field activity remarks, it also executes any field activity remark completion events.	Inbound Communication Business Object: Field Activity Inbound Communication D1-FieldActivityIBComm
11.	The inbound communication updates the outbound communication. This update is performed by an Enter algorithm on the “Completed” Status of the inbound communication business object’s lifecycle.	Inbound Communication Business Object: Field Activity Inbound Communication D1-FieldActivityIBComm Outbound Communication Business Object: Field Activity Outbound Communication (D1-FieldActivityOBComm)

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Base Package Objects
12.	The outbound communication updates the “Completion Flag” and the original service order field activity business object. This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect) Field Activity Business Object: Field Activity (D1-FieldActivity)

Unrelated Pickup Orders

When field work crews are out performing field work, it’s possible that they will encounter other work unrelated to their current task that needs to be done. This type of work can be as simple as trimming a tree whose branches are too close to power lines, or the replacement of a meter for a different customer or service point. These types of task are referred to as "unrelated pickup activities." Crews can either work the field activity or leave it to be assigned to another crew at a later date.

When the crew creates an unrelated pickup activity in the field work system, it is sent to Service Order Management, and a corresponding service order field activity is created in the system.

Unrelated pickup activities can be created via one of the following Inbound Web Services:

- Field Activity Asynchronous Req Inbound (D1-FARequestAsynchronous)
- Field Activity Synchronous Req Inbound (D1-FARequestSynchronous)

Once created, the are processed like any other service order field activity. If the pickup activity was completed in the field before being sent to Service Order Management, it will quickly move through its lifecycle (as now further action is needed) until it reaches the "Completed" state.

Retrieving Service Point Information

If the unrelated pickup activity is customer-related it will require service point information to be created. This information can be queried by the field work crew via the "Field Work Service Point Query" (D1-FieldWorkSPQuery) Inbound web service.

This service uses set of service point criteria to allow the field crew to search for a service based upon either service point or device information. The service returns a list of service points that is configurable in length. If the number of results is larger than the configured maximum length the service indicates that additional records exist and the crew can request another set of results allowing them to identify the proper service point to associate to the activity.

There are times when an unrelated pick-up activity is identified but the field crew is out-of-coverage (i.e. no network connection) and they will not be able to immediately verify service point information. In this type of situation, the crew can input the service point criteria fields and create the activity, which, when imported into Service Order Management, will attempt to identify the service point based upon the information provided. If the service point can be uniquely identified everything should operate as normal. If the service point cannot be uniquely identified then the service order field activity is set to the error state.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Service Order Field Activity Types

- Understanding Service Order Field Activity Types
- Configuring Service Order Field Activity Types

Understanding Service Order Field Activity Types

A single service order field activity type must be configured to support communication with external field work systems such as Oracle Utilities Module Workforce Management. Service order activity types are assigned to the “Field Activity” Activity Type Category.

Refer to Understanding Activity Types for more information about activity types.

Service Order Field Activity Types and Divisions

Service order field activity types can specify a Division, to represent the specific operating company associated with activities of that type. If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", users can only access service order activities or service order field activities with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the**Miscellaneous** tab of the **User** portal.

See Understanding Divisions for more information about divisions.

Configuring Service Order Field Activity Types

You use the **SOM Activity Type** portal is to view and maintain service order field activity types.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order field activity types as well as service order field activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List:** This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type:** This zone provides information about the selected Activity Type
- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types

Copyright © 2026, Oracle and/or its affiliates.

that have yet to be configured. Broadcast a record to display the details of the selected record.

- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the “Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

Service Order Management Configuration

The following table outlines the activity types that must be configured for the service order field activity type supported by Service Order Management:

Field Activity	Activity Type (Business Object)
Field Activity	Field Activity (D1–FieldActivityType)

The demonstration database contains examples of each of these service order activity types.

Configuring Service Order Field Activity Reschedule Reasons

The Reschedule Reason (D1-RescheduleReasonLookup) extendable lookup can be used to define reasons for rescheduling field activities. Reschedule reasons are captured on field activities, as well as their parent service order activity and an update orchestration activity.

Use the following steps to define service order field activity reschedule reasons:

Copyright © 2026, Oracle and/or its affiliates.

1. Select **Admin**, **General**, **Extendable Lookup** to access the **Extendable Lookup Query** portal.
2. Enter “D1-RescheduleReasonLookup” in the **Business Object** field (or enter “Reschedule Reason” in the **Description** field) and click **Search**.
3. Click the “Reschedule Reason” link in the **Description** column of the search results.
4. Click **Add**, and enter the following details for the reschedule reason (fields not listed below are optional):
    - **Reschedule Reason**: A code used to designate the reschedule reason.
    - **Status**: The status of the reschedule reason.
    - **Description**: A description of the reschedule reason. This is what will appear in **Reschedule Reason** drop-down list on the service order field activity.
5. Click **Save**.

Field Task Types

- Understanding Field Task Types
- Configuring Field Task Types

Understanding Field Task Types

A service order field activity’s field task type defines details about the type of task to be performed and how the system will process the activity.

Field Task Type Information

Field task types are values for the Field Task Type (D1-FieldTaskTypeLookup) extendable lookup. Each field task type value includes the following information:

- **Routing**: Indicates if field tasks of this type can only be performed at a service point. Valid values are "SP Required" and "Pass-Through".
- **Appointment Option**: Indicates if an appointment (via a mobile workforce application) is required or applicable to field tasks of this type. Valid values are "Not Applicable", "Optional", and "Required".
- **Completion Events When Successful**: One or more completion events that are executed upon successful completion of field tasks of this type.
- **Completion Events When Canceled**: One or more completion events that are executed upon cancellation of field tasks of this type.
- **Duplicate Task Type Information**: Defines processing rules for handling potential duplicate field tasks, including:
    - **Allow Duplicates**: Specifies whether or not duplicate field tasks are allowed
    - **Duplicate Threshold**: A number of hours used to determine if a newly instantiated field task type should be considered a duplicate.
    - **Field Task Types**: A list of one or more field task types that are considered to be duplicates of the field task type
- **Conflict Task Type Information**: Defines processing rules for handling potentially conflicting field tasks, including:

Copyright © 2026, Oracle and/or its affiliates.

  - **Allow Conflicts**: Specifies whether or not conflicting field tasks are allowed
  - **Conflict Threshold**: A number of hours used to determine if a newly instantiated field task type should be considered a conflict.
  - **Field Task Types**: A list of one or more field task types that are considered to conflict with the field task type
- **Processing Scripts**: Defines one or more processing scripts to extract supplemental information needed by the mobile workforce application to schedule field tasks of this

Configuring Field Task Types

Field task types are configured using the extendable lookup portal.

You can access the extendable lookup portal from the **Admin > General > Extendable Lookup**.

Use the **Extendable Lookup Search** zone to search for and select the Field Task Type (D1-FieldTaskTypeLookup) extendable lookup.

The following zones may appear as part of the portal's **Main** tab page:

- **Extendable Lookup List:** This zone displays a list of values for the Field Task Type extendable lookup.
- **Extendable Lookup List:** This zone provides information about the selected value.

Service Order Management Configuration

The following table outlines the field task types that must be configured to support each of the service order activity types supported by Service Order Management:

Service Order Activity	Field Task Types
Enable Service — Meters	Connect SP at Source: D1-ConnectSPAtSource Connect SP at Meter: D1-ConnectSPAtMeter Connect SP at Source and Turn On: D1-ConnectSPAtSourceAndTurnOn Connect SP at Meter and Turn On: D1-ConnectSPAtMeterAndTurnOn Connect SP at Source and Install Meter: D1-ConnSPAtSrceAndInstMtr Connect SP at Meter and Install Meter: D1-ConnSPAtMtrAndInstMtr Install Meter: D1-InstallMeter Turn On Meter: D1-TurnOnMeter

Copyright © 2026, Oracle and/or its affiliates.

Service Order Activity	Field Task Types
Enable Service — Items	Connect SP at Source: D1-ConnectSPAtSource Item - Connect SP at Device: D1-ConnectSPAtDevice Item - Connect SP at Source and Turn On: D1-ConnSPAtSrceAndTurnOnDvc Item - Connect SP at Device and Turn On: D1-ConnectSPAtDvcAndTurnOn Connect SP at Source and Install Device: D1-ConnSPAtSrceAndInstDvc Connect SP at Device and Install Device: D1-ConnSPAtDvcAndInstDvc Item - Install Device: D1-InstallDevice Turn On Item: D1-TurnOnItem
Disable Service — Meters	Turn Off Meter: D1-TurnOffMeter
Disable Service — Items	Turn Off Item: D1-TurnOffItem
Cut Service for Non-Payment — Meters	Cut for Non-Payment: D1-CutForNonPayment
Cut Service for Non-Payment — Items	Item - Cut for Non-Payment: D1-CutItemForNonPayment
Reconnect Service for Payment — Meters	Reconnect for Payment: D1-ReconnectForPayment
Reconnect Service for Payment — Items	Item - Reconnect for Payment: D1-ReconnectItemForPayment
Meter Exchange	Exchange Meter: D1-ExchangeMeter
Item Exchange	Exchange Device: D1-ExchangeDevice
Back-to-Back — Meters	Read Meter: D1-ReadMeter

Defining Credit & Collections Options

> **Note**
>
> The functionality described in this section is meant to handle the collection of unpaid balances. If your organization practices open-item accounting and collects on unpaid bills, you will not use this functionality. Rather, you will use the functionality described under Defining Overdue Processing Options.

The system periodically monitors how much your customers owe to ensure they haven't violated your collection criteria. When a violation is detected, the system generates the appropriate responses (e.g., letters, disconnect notices, collection agency referrals, and eventually write off). This section describes how to set up the tables that control your credit & collections processing.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Setting up the credit & collections control tables is as challenging as your organization's collection rules. If you have simple rules then your setup process will be straightforward. If your collection rules are complicated (e.g., they differ based on the type of customer, the type of debt, the age of debt, the type of service, etc.), then your setup process will be more challenging.

- The Big Picture Of Credit & Collections
- Creating Collection, Severance & Write-Off Procedures
- How To

The Big Picture Of Credit & Collections

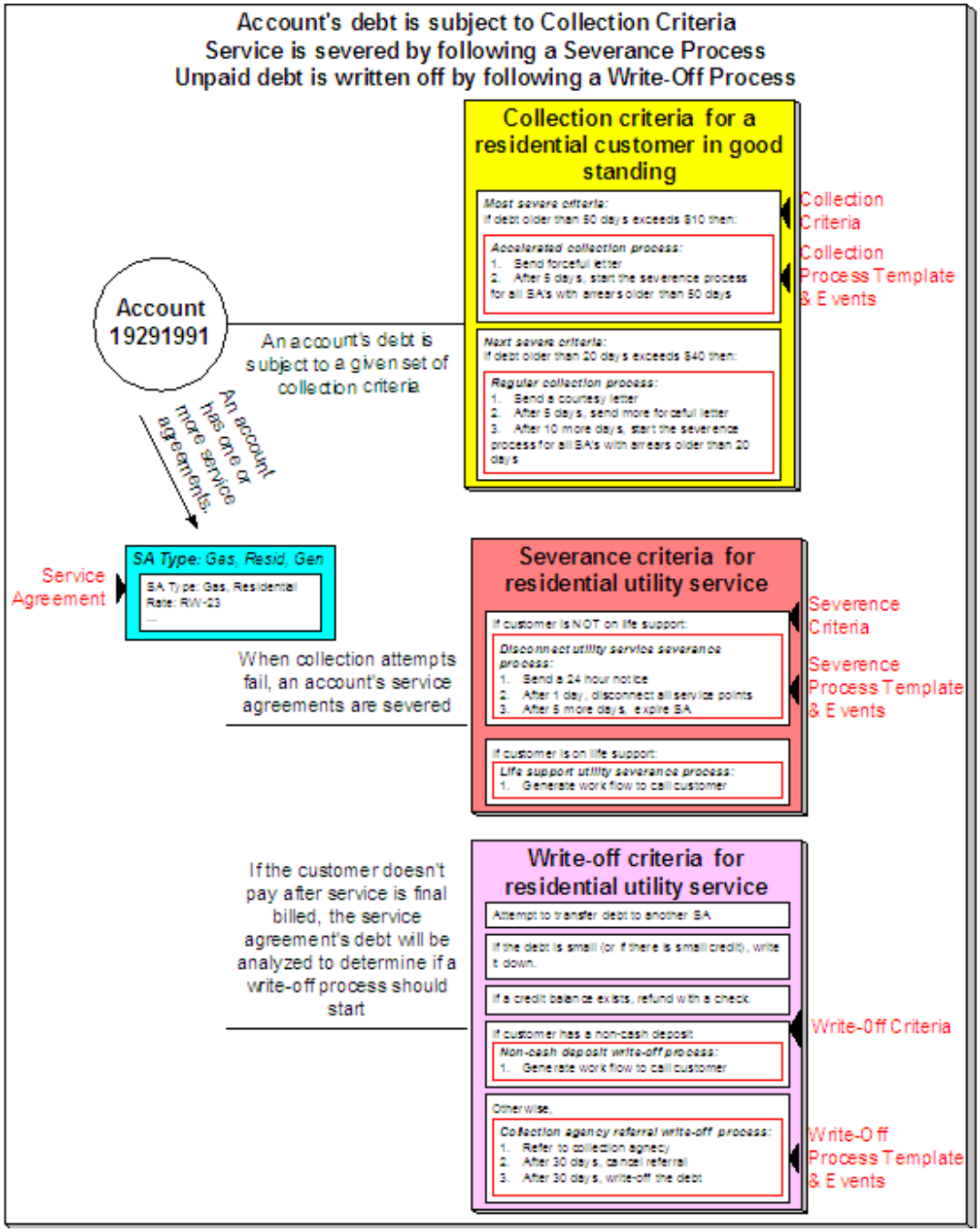
This section provides an overview of important C&C concepts with which you should be familiar before you set up your C&C control tables.

- Collection Criteria versus Severance Criteria Versus Write Off Criteria
- The C&C Monitors
- The Big Picture Of Collection Processes
- The Big Picture Of Collection Events
- The Big Picture Of Severance Process Cancellation
- The Big Picture Of Severance Events
- The Big Picture Of Write Off Processing
- The Big Picture Of Write-off Events
- Calendar Versus Work Days
- The Big Picture Of Payment Arrangements and Pay Plans

Collection Criteria versus Severance Criteria Versus Write Off Criteria

The following diagram introduces important concepts related to the C&C processes:

Copyright © 2026, Oracle and/or its affiliates.



There are many important concepts illustrated above:

An account's debt comes from its service agreements	An account's debt is managed at the service agreement level, i.e., the system keeps track of how much a customer owes in respect of each service agreement. In order to determine an account's balance, the system must add up the debt on each of the account's service agreements.
Collection criteria define intolerable debt	Collection criteria are control data that define intolerable debt. Most criteria are defined using a combination of number of days in arrears and a dollar amount.

Copyright © 2026, Oracle and/or its affiliates.

Collection criteria may be compared to an account's total debt or to subsets of debt	If your organization has simple collection procedures, you will probably target collection criteria at an account's total debt. However, you have the option of segregating an account's debt into debt classes and targeting the collection criteria at each class. For more information about debt classes, see Different Collection Criteria For Different Customers And Different Debt.
Collection criteria also define what to do when the level of intolerable debt is exceeded	When you define collection criteria, you also define how the system should respond if an account violates your criteria. These collection events are defined in respect of a "collection process template".
There are usually several collection events that take place when criteria are violated	A collection process template usually has several collection events. Each event is meant to prod the customer to pay. The initial collection events are typically letters. If payment is not received after several such attempts, the last collection event typically starts a severance process for each service agreement in arrears.
A severance process template defines how to sever a service agreement	A "severance process template" defines how to sever a given type of service agreement. A severance process template usually contains several severance events. These events are a series of letters and / or disconnection field activities that eventually result in the expiration of a service agreement if payment is not received.
Severance criteria define how to sever service agreements	Severance criteria define the severance process to be executed for service agreements of a given SA type. The severance process may differ depending on some attribute of the customer (or premise). For example, you may have a different severance process if the customer has life support equipment.
After a service agreement is severed, it will be final billed	When the last active service agreement linked to an account is stopped, the system changes the account's bill cycle to bill that evening. If only one of many SAs is stopped, the SA will only be final billed as per the account's original bill cycle schedule.

Copyright © 2026, Oracle and/or its affiliates.

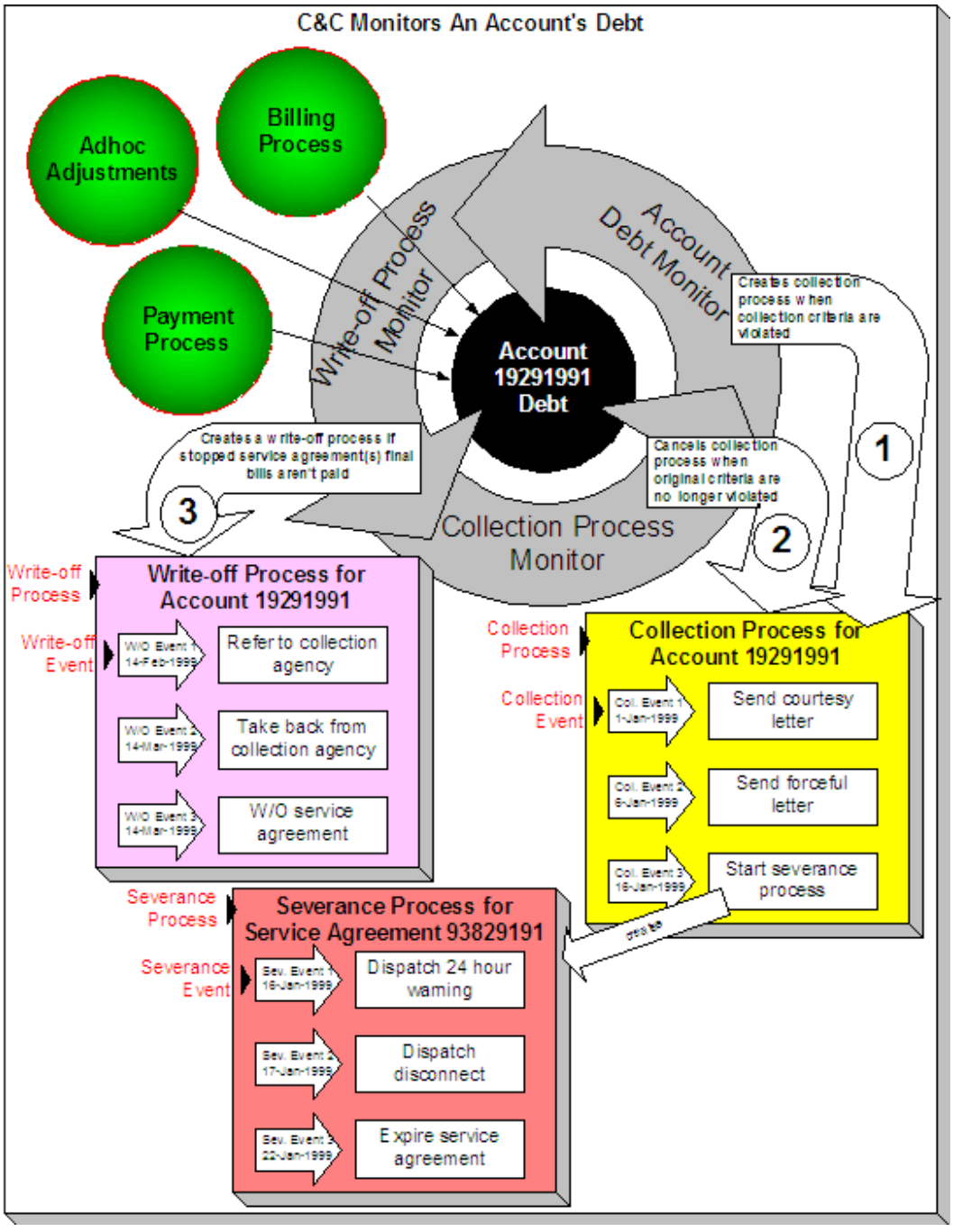
If a customer doesn't pay their final bill, the account's debt will be analyzed to determine if the system can reduce the debt to zero using a variety of mechanisms	The system will look at an account's finaled debt on its next scheduled credit review date (typically a few days after the bill's due date). The system will attempt to reduce the service agreement's debt to zero using all of the following methods: • If the account has active service agreements, it will transfer the finaled debt to an active service agreement. – If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit). * If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer).
If a customer's finaled debt cannot be reduced via any of the previous methods, the system creates a write-off process	A write-off process contains one or more write-off events. These events can generate a letter, send a To Do entry to a CSR, send a referral to a collection agency, etc. When you set up the system, you define the type of write-off process to use for every collection class / write-off debt class combination. In addition, you can also indicate when the type of write-off process should differ depending on some attribute of the customer (or premise). For example, you may have a different write-off process if the customer has a non-cash deposit.
The last write-off event typically causes the debt to be written off	Ultimately, if the write-off events fail, the debt will have to be written off. When debt is written-off, the system creates a write-off service agreement and transfers the outstanding debt to it. This means the debt stays with the account for life and will have to be paid off if the customer ever returns.

> **Note**
>
> **Checkpoint.** At this point, you should be familiar with the concept that an account's debt is compared to user-defined collection criteria. If the account violates the criteria, a series of events will ensue that prod the customer to pay. If the customer doesn't respond, every service agreement in arrears will be severed (i.e., disconnected). If lack of service doesn't inspire payment, the service agreement will be expired and a write-off process will be created to manage the write-off activities.

The C&C Monitors

Copyright © 2026, Oracle and/or its affiliates.

Your collection, severance and write-off criteria described in the previous section exist to support the processes that manage the collection activities. The following diagram illustrates, at a high level, the major processes that manage the collection of overdue debt:



There are many important concepts illustrated above:

Bills, payments and adjustments affect an account's debt	An account's debt is the accumulation of all bills, payments and adjustments.

Copyright © 2026, Oracle and/or its affiliates.

The Account Debt Monitor creates a collection process when an account violates collection criteria	Periodically, a background process referred to as the Account Debt Monitor determines if an account's debt violates your collection criteria. If so, a collection process is created using the violated criteria's collection process template. Refer to When Is An Account's Debt Monitored? for a description of when an account's debt is compared against collection criteria.
A collection process contains one or more collection events	The collection process contains a series of collection events. These events correspond with the collection event types associated with the collection process template. The initial collection events are typically letters. If payment is not received after several such communications, the last collection event typically starts a severance process for each service agreement in arrears.
The Collection Process Monitor cancels a collection process when warranted	The Collection Process Monitor cancels a collection process when its service agreements satisfy your cancellation criteria (e.g., when the service agreements have less than $10 of debt older than 20 days). Refer to How Are Collection Processes Cancelled for more information about the cancellation process.
The last collection event starts one or more severance processes	The last collection event typically starts one or more severance processes. A severance process contains the activities necessary to sever a service agreement. The service agreement(s) that are severed may be all SAs that are associated with the collection process. Alternatively, you can nominate a service agreement to act as the primary service to cut (you'd do this if you cut electricity when the customer doesn't pay for their gas). The algorithm on the collection event that starts severance will control which service agreement(s) are severed. Refer to How To Nominate A Single Service Agreement To Sever for more information.
Each service agreement has its own severance process	Every service agreement that is severed has a severance process. The type of process is dependent on the severance criteria linked to the service agreement's SA type.
A severance process contains one or more severance events	The severance process contains a series of severance events. The events correspond with the severance process template's severance events.

Copyright © 2026, Oracle and/or its affiliates.

The system cancels a severance process when warranted	The system cancels a severance process when its service agreement satisfies your cancellation criteria (note, it is possible to set up the system so that all service agreements in the debt class must satisfy your cancellation criteria before a severance process is cancelled). It's important to note that the cancellation is real time (as opposed to the cancellation of collection processes, which happens in a background process). Refer to How Are Severance Events Canceled? for more information.
The last severance event should expire the service agreement	The last severance event typically expires its service agreement. When the last service agreement linked to an account is expired, the system will schedule the account for billing (outside of its normal bill cycle schedule).
If you nominate a single SA to sever when multiple SAs are in arrears...	Earlier we indicated that you can nominate a service agreement to act as the primary service to cut (you'd do this if you cut electricity when the customer doesn't pay for their gas). If you do this, you also need a severance event that will sever all other service agreements in the debt class if the severance of the nominated service agreement doesn't inspire payment. A severance event algorithm to do such is supplied with the base package. Refer to How To Nominate A Single Service Agreement To Sever for more information.
The Write-Off Monitor creates a write off process to collect stopped, unpaid debt	The Write-Off Monitor reviews stopped and reactivated service agreements after their closing bill's due date (plus grace period). The Write-Off Monitor attempts to reduce the service agreement's debt to zero using all of the following methods: • If the account has active service agreements, it will transfer the finaled debt to an active service agreement. – If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit). * If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer). If the system is unsuccessful in reducing the account's debt to zero, a write-off process will be created using the appropriate write-off process template. Refer to The Big Picture Of Write Off Processing for more information about the write-off process.

Copyright © 2026, Oracle and/or its affiliates.

A write-off process contains one or more write-off events	The write-off process contains a series of write-off events. These events correspond with the write-off event types associated with the write-off process template. The initial write-off events are typically collection agency referrals and/or letters. If payment is not received as a result of such efforts, the last write-off event typically writes off the customer's debt.
The system cancels a write-off process when warranted	The system cancels a write-off process when its service agreements no longer have debt (i.e., they become closed).
Another write-off process will be created if a closed service agreement ever reactivates	If a service agreement becomes reactivated (e.g., because the final payment bounces), the service agreement will be processed by the Write-Off Monitor and the whole write-off process starts again.

Note

Checkpoint. At this point, you should be familiar with the concept that a collection process will be created for an account that violates collection criteria. The collection process consists of a series of events that typically generate letters and / or To Do entries. If the customer doesn't respond, a severance process will be started for one or more service agreements. A severance process consists of a series of events that typically generate letters and/or disconnection field activities. If lack of service doesn't inspire payment, the last severance event expires the service agreement (and a final bill will be scheduled when the last service agreement is expired). If the customer doesn't pay the final bill, a write-off process will be created for each type of unpaid debt. The write-off process consists of a series of events that ultimately result in the write-off of the customer's debt. When debt is written-off, the system creates a write-off service agreement and transfers the outstanding debt to it. This means the debt stays with the account for life (because the write-off service agreement is linked to the account) and will have to be paid off if the customer ever returns.

The Big Picture Of Collection Processes

The topics in this section describe how collection processes are created and cancelled.

NOT_SUPPORTED

For more information refer to The Lifecycle Of A Collection Process And Its Events.

- How Does The Account Debt Monitor Work?
- How Are Collection Processes Cancelled?

Copyright © 2026, Oracle and/or its affiliates.

How Does The Account Debt Monitor Work?

This section describes how the Account Debt Monitor uses your collection criteria and collection process templates to collect overdue debt.

- Different Collection Criteria For Different Customers And Different Debt
- Override Conditions
- When Is An Account's Debt Monitored?
- How Is An Account's Debt Monitored?
- What Happens When A Collection Process Is Started?
- Experimenting With Alternative Collection Process Templates

Different Collection Criteria For Different Customers And Different Debt

Consider the following:

- You probably have different collection criteria for different jurisdictions (i.e., CIS Divisions). For example, if you have customers in different states / provinces, you may have different regulator-imposed criteria applied to each state's debt. You differentiate your debt in respect of the collection process via the **CIS division code on each customer's account**.
- You probably have different collection criteria for different customer segments. For example, customers with large bills probably have strict criteria, whereas you're probably more lenient with small customers (or vice versa). You differentiate your customers in respect of the collection process via a **collection class code on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You probably have different collection criteria for different classes of debt. For example, if a single customer has both regulated and unregulated debt, you probably have commission-imposed criteria to be applied to the regulated debt, but you have the freedom to apply stricter criteria to the unregulated debt. You differentiate your debt in respect of the collection process via a **debt class code on the customers' service agreements** (note: the debt class is actually defined on the service agreement's SA type).
- You will have different criteria for every currency in which you work because the monitoring process always compares a customer's debt against some value and this value must be denominated in the customer's currency. A customer's currency is defined using a **currency code on the account**.

Given the above, you should understand that different collection criteria will exist for every combination of CIS division, collection class, debt class, and currency code. If you're confused, consider the following matrix (where we assume you have a single currency and division and therefore avoid the third and fourth dimensions):

SA's Debt Class	Account's Collection Class: Commercial Customer	Account's Collection Class: Residential Customer

Copyright © 2026, Oracle and/or its affiliates.

Regulated	N/A - there is no regulated, commercial customer debt.	Highest Priority: If > $5 in arrears by more than 50 days, create the accelerated collection process for residential customers. Lower Priority: If > $25 in arrears by more than 25 days, create the courtesy reminder collection process for residential customers.
Unregulated	Highest Priority: If > $10 in arrears by more than 50 days, create the accelerated collection process for commercial customers. Lower Priority: If > $1000 in arrears by more than 25 days, create the normal collection process for commercial customers.	Highest Priority: If > $10 in arrears by more than 25 days, create the normal collection process for residential customers.
Charitable Contribution	Highest Priority: If > $10 in arrears by more than 50 days, create the charitable collection process.	Highest Priority: If > $10 in arrears by more than 50 days, create the charitable collection process.

Also, notice that there can be multiple criteria for each cell in the matrix. What differentiates one collection criteria from another is its priority. The higher priority criteria will be compared first. If the debt meets the criteria, the collection process is initiated and no further comparisons are performed.

NOT_SUPPORTED

For more information about maintaining this matrix, refer to Setting Up Collection Class Controls. For more information about how the system handles an element in this matrix that has multiple criteria, see How Is An Account's Debt Monitored?.

Override Conditions

Warning

Your credit & collection requirements may not require any overrides and therefore this section may not be relevant for your organization.

The matrix presented in the previous section showed:

- You can have different collection criteria for different categories of debt and customers.
- When a collection criteria is violated, the system generates a specific collection process.

Copyright © 2026, Oracle and/or its affiliates.

This works great for many organizations, but if your organization has other factors that affect either the collection criteria OR the collection process that is initiated when the criteria is violated, you may need to use override collection criteria. For example,

- If you have a different collection process for regulated, residential debt during the winter months, you'll need to use override collection criteria (where the override criteria is "if it's winter").
- If you have different collection criteria for customers with a poor credit score, you'll need to use override collection criteria (where the override criteria is "if the customer's credit rating is poor").

NOT_SUPPORTED

Refer to Designing Your Collection Class Control Overrides for more information.

This section describes how and when the Account Debt Monitor analyzes an account's debt.

When Is An Account's Debt Monitored?

The account debt monitor (ADM) analyzes an account's debt at least every X days, where X is defined on the customer class control associated with the account's customer class and division (in the field Min Credit Review Freq (Days)).

In addition, an account's debt will also be monitored as follows:

- The ADM looks at an account's debt X days after an account's bill due date (X is defined on the account's customer class in the field Collection Grace Days).
- The ADM looks at an account's debt after a payment is canceled when the cancellation reason indicates NSF (non-sufficient funds).
- The ADM looks at an account's debt after a payment arrangement is broken (assuming you use the base package's break payment arrangement plug-in). Refer to Monitoring Payment Arrangements for more information.
- The ADM looks at an account's debt after a pay plan is broken. Refer to The Pay Plan Monitor for more information.

How Is An Account's Debt Monitored?

Assume the following collection control matrix exists for your organization:

SA's Debt Class	Account's Collection Class: Commercial Customer	Account's Collection Class: Residential Customer

Copyright © 2026, Oracle and/or its affiliates.

Regulated	N/A - there is no regulated, large customer debt	Highest Priority: If > $5 in arrears by more than 50 days, create the accelerated collection process for residential customers. Lower Priority: If > $25 in arrears by more than 25 days, create the courtesy reminder collection process for residential customers.
Unregulated	Highest Priority: If > $10 in arrears by more than 50 days, create the accelerated collection process for commercial customers. Lower Priority: If > $1000 in arrears by more than 25 days, create the normal collection process for commercial customers.	Highest Priority: If > $10 in arrears by more than 25 days, create the normal collection process for residential customers.

This matrix contains the information used by the Account Debt Monitor.

NOT_SUPPORTED

For more information about the information in this matrix, refer to Different Collection Criteria For Different Customers And Different Debt.

This matrix can be overwhelming when viewed as a whole. So let's consider how to use it for a specific account's debt and things will become clearer.

First, because an account belongs to a unique collection class, we only have to worry about a single column in the matrix when monitoring an account's debt.

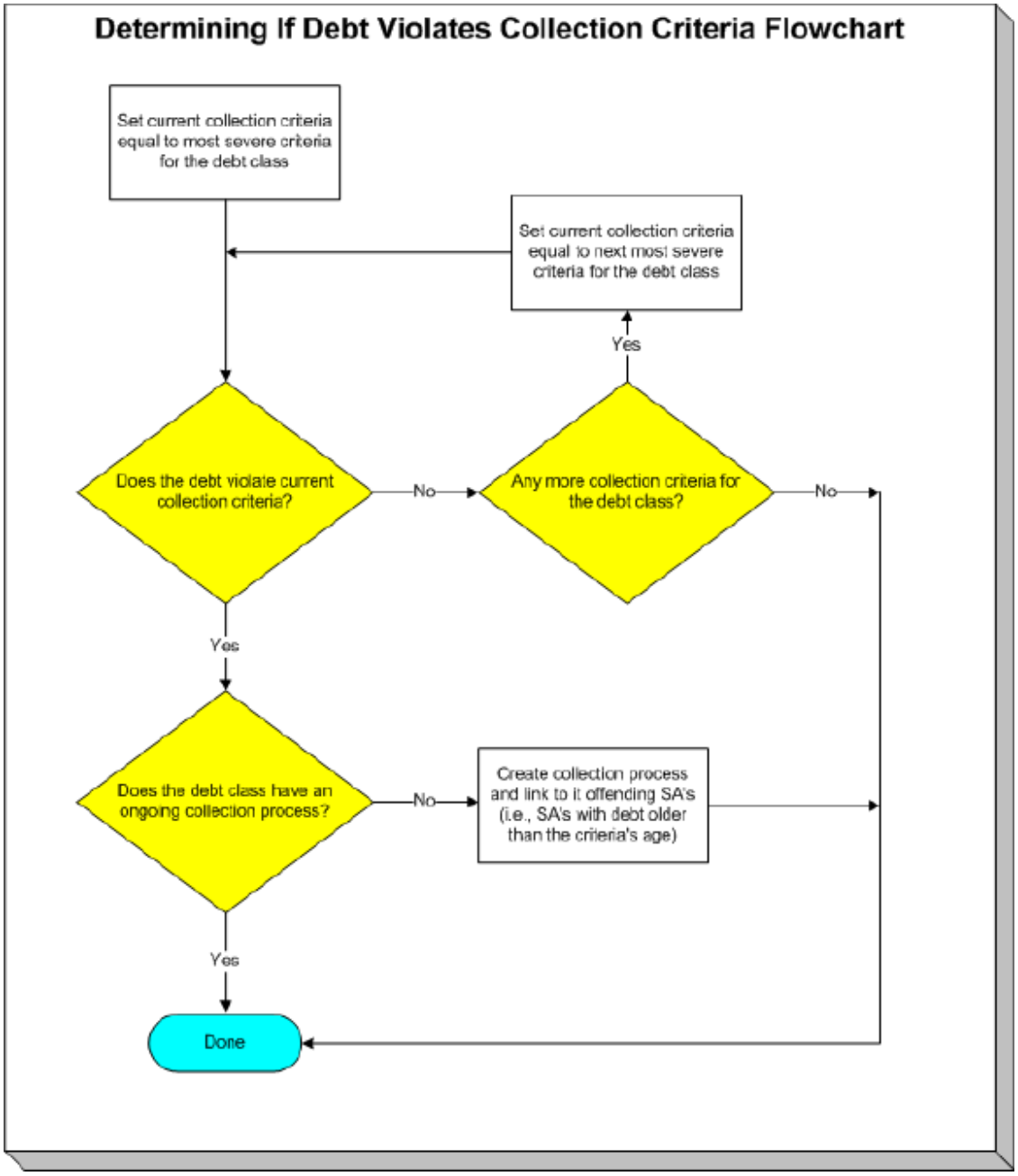
Next, we accumulate the total amount of aged debt for each unique debt class associated with the account's service agreements.

Next, we subject the accumulated aged debt to the override aged debt algorithm (plugged in on the debt class). This algorithm can cause aged debt to be reduced. This is an optional algorithm and is only used if you set up pay plans for customers. Refer to How Pay Plans Affect The Account Debt Monitor for more information.

Next, we determine if the debt for the debt class violates the collection criteria in the respective matrix element. If so, we kick off a collection process and link the offending service agreements to it. The logic associated with the determination of whether to kick off a collection process is rather sophisticated. The following flowchart explains the exact details.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Important.** If a service agreement is part of an ongoing severance process, it will NOT be considered by the Account Debt Monitor (it's already being severed). If a service agreement is stopped, closed, or reactivated, it also will NOT be considered by the Account Debt Monitor (it's already severed).



Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Multiple collection processes may be kicked off.** It's important to be aware that if an account's service agreements reference multiple debt classes, a collection process will be started for each offending debt class.
>
> **One collection process per debt class.** A given debt class for an account may only have one ongoing collection process at any point in time.

What Happens When A Collection Process Is Started?

When you define collection criteria, you must define the collection process template to use if the criteria are violated. The system uses this template to create the account-specific collection process.

Every service agreement that is part of the offending debt class that has debt older than X days will be linked to the collection process (where X is the debt age on the collection criteria).

Also linked to the collection process will be one or more collection events. These events are typically a series of letters meant to prod the customer (you can also create an event that sends a To Do entry to a user to highlight the offensive debt). You define exactly which letters are generated and when they are generated when you set up the events on your collection process templates.

It's important to note that all of the collection events will be created when the collection process is created. Each of these collection events contains a trigger date. The trigger date of the first event(s) will typically be the current date. The trigger date of the other events will be in the future. Refer to Calendar vs Work Days for information that describes how the trigger date is set.

A separate process, Activate Collection Events, is responsible for activating collection events whose date is on or before the current date. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, start a severance processes, etc.)

If adequate payments / credits are recorded in the system, the collection process will be cancelled.

> **NOT_SUPPORTED**
>
> For more information about collection process templates, see Setting Up Collection Process Templates. For more information about collection events, see The Big Picture Of Collection Events. For more information about how a collection process is cancelled, see How Are Collection Processes Cancelled.

Experimenting With Alternative Collection Process Templates

The system allows you to determine the efficacy of proposed collection process templates using a small subset of customers before implementing the templates on the entire customer base. We use the term "champion / challenger" to reference this functionality.

Copyright © 2026, Oracle and/or its affiliates.

We'll use an example to explain. Let's assume your prevailing collection process template for residential customers starts with a "gentle reminder" letter followed 10 days later by a letter threatening collection agency referral if payment is not received. You may want to experiment with the impact of a change to this template. For example, you may want to change the "gentle reminder" to something more assertive and follow this up 5 days later with an even sterner warning. You can use the "champion / challenger" functionality to perform this experiment.

The following points describe how to implement "champion / challenger" functionality:

- Set up a "challenger" collection process template for each template that you want to experiment with.
- Insert a new **Champion/Challenger** option on the Collection Processing Feature Configuration for every champion template. Each option's value defines:
  - the "champion" collection process template code
  - the "challenger" collection process template code
  - the percentage of the time the system should use the "challenger" template
- Keep in mind that you can only experiment with one challenger template per champion template. For example, let's assume you have two prevailing collection process templates - one for residential customers and another for commercial customers. You can experiment with different challenger templates for the residential and commercial templates. However, you cannot experiment with two different challenger templates for the residential champion template (i.e., a champion template can have 0 or 1 challenger template).

After setting up the above, the Account Debt Monitor will use the challenger template X% of the time rather than the champion template.

If you are using the Oracle Utilities Analytics product, you can configure analytic zones in innumerable ways to compare the efficacy of the champion versus the challenger. For example,

- You can set up a graph to show the average duration of each type of process.
- You can set up a graph to show the average dollars that were successfully collected.
- You can set up a dimensional scorecard to show how each template performed in different regions (or customer classes or ...).
- Etc (the list is limited by your imagination.

How Are Collection Processes Cancelled?

A collection process may be cancelled via the mechanisms described in this section.

- The Collection Process Monitor Can Cancel A Collection Process
- A New Payment Plan Can Cancel A Collection Process
- A User May Cancel A Collection Process At Their Discretion
- Stopping A Service Agreement May Cancel A Collection Process

The Collection Process Monitor Can Cancel A Collection Process

Copyright © 2026, Oracle and/or its affiliates.

The Collection Process Monitor (CPM or C1-CPM)) is a background process that reviews a collection process when the debt associated with one of its service agreements is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment.
- The creation of a payment segment.
- The creation of an adjustment that credits a service agreement.

The review performed by the CPM occurs as follows:

- **Debt class cancel criteria.** In general, the sum of all debt associated with the collection process's debt class must be less than a given threshold amount for a collection process to be cancelled. If so, the collection process is cancelled.
- Please be aware that, if a Pay Plan exists for the account and debt class, the customer's debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount. Please be aware that this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

> **Note**
>
> **The above logic is not "hard coded".** The CPM calls the Collection Process Cancel Criteria Algorithm defined on the debt class that is associated with the collection process. This algorithm will cancel a collection process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. However, because it's an algorithm, you can introduce whatever cancellation criteria you please.

- **Service agreement cancel criteria.** You can optionally introduce a special quirk to the cancellation logic. This quirk is a bit difficult to understand. To understand it, you should recall:
  - All service agreements that are in arrears in a given debt class are linked to the collection process.
  - The collection event called Start Severance creates a severance process for every service agreement that is in arrears on the collection process (the alternative is to Nominate A Single Service Agreement To Sever).
  - If you use the Start Severance collection event, you would want to remove a service agreement from a collection process when it no longer has intolerable debt (regardless of the state of the debt class's entire debt). You'd want to do this because, if you don't, the system would start a severance process for the paid up service agreement and if it's paid up, you wouldn't want a severance process created for it.
- To "remove" service agreements from a collection process when they no longer have intolerable debt, you should plug-in a Service Agreement-Oriented Cancel Criteria Algorithm on your collection process templates. The CPM will call this algorithm if you've plugged it in.

> **Note**
>
> When all service agreements are "removed" from a collection process, the CPM cancels all pending collection events and cancels the collection process.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Checking if individual service agreements should be removed from a collection process is optional (meaning that you don't have to plug one in on the collection process template).

A New Payment Plan Can Cancel A Collection Process

Refer to Collection Process / Severance Process Cancellation When A Pay Plan Is Created for the details.

> **Note**
>
> **Real time cancellation.** Please be aware that the system will cancel a collection process real time when a pay plan is created (if the pay plan's scheduled payments are enough to pay-off the customer's outstanding debt).

A User May Cancel A Collection Process At Their Discretion

A user may cancel a collection process at their discretion.

Stopping A Service Agreement May Cancel A Collection Process

The system will "remove" a service agreement from a collection process when it is stopped (i.e., when the service agreement's status becomes Stopped). When the last service agreement is "removed" from the collection process, the collection process will be cancelled.

The Big Picture Of Collection Events

This section describes the various types of collection events and their lifecycle.

- How Are Collection Events Created?
- Types Of Collection Events
- Collection Event Lifecycle
- Collection Event Trigger Date
- How Are Collection Events Completed
- The Last Collection Event Should Kick Off Severance Process(es)
- How Are Collection Events Canceled?

Copyright © 2026, Oracle and/or its affiliates.

How Are Collection Events Created?

Collection events may be created as follows:

- The Account Debt Monitor creates a collection process when an account violates collection criteria. The collection process has one or more collection event(s). The number and type of events is controlled by the collection process template associated with the collection process.
- Collection events are created when a user creates an ad hoc collection process. The number and type of events is controlled by the collection process template defined when the collection process is created.
- An ad hoc collection event may be created and linked to an existing collection process by a user at their discretion.

> **Note**
>
> **Bottom line.** Most collection events are created by the system when it creates a collection process for delinquent accounts. If you need to create an ad hoc collection event, you can either create a collection process whose template contains the desired event OR link the desired event to an existing collection process.

> **NOT_SUPPORTED**
>
> For more information about the creation of events by the Account Debt Monitor refer to What Happens When A Collection Process Is Started?. For more information about creating ad hoc collection processes, refer to Collection Process Maintenance. For more information about creating ad hoc events, refer to Collection Process - Events.

Types Of Collection Events

The following table describes the various types of collection events and what happens when they are completed:

Type Of Collection Event	What Happens In The System
Send Letter	A customer contact is created for every financially responsible person linked to the account. The customer contact causes a letter to be produced. The type of letter is defined on the customer contact's contact type. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).

Copyright © 2026, Oracle and/or its affiliates.

Create To Do Entry	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the collection event type.
Cancel Budget	Every service agreement linked to the account that is on a budget is "removed" from the budget (i.e., the recurring charge amount for the service agreement is set to zero). In addition, a syncing adjustment is issued to cause each SA's current balance to be set equal to their payoff balance (the adjustment type is defined on the SA's SA type). Also, the Budget Plan field on the Account - Budget page is cleared.
Generic Algorithm	The algorithm defined on the event's event type is executed.
Start Severance Process	A severance process is created for every service agreement linked to the collection process. The type of severance process is defined on the SA type table (every service agreement references an SA type).

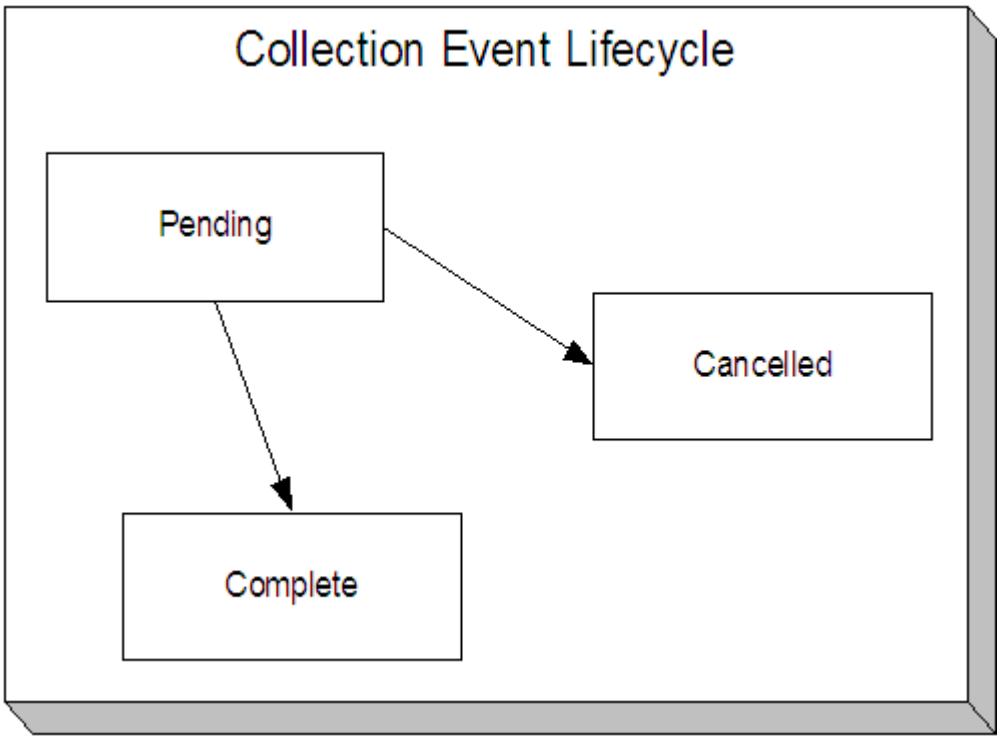
NOT_SUPPORTED

Refer to Setting Up Collection Event Types for more information.

Collection Event Lifecycle

The following diagram shows the possible lifecycle of a collection event:

Copyright © 2026, Oracle and/or its affiliates.



Collection events are initially created in the pending state.

When the system sees a pending event with a trigger date on or before the current date, the system executes the event's activity and completes the event.

NOT_SUPPORTED

For more information about a collection event's trigger date, see Collection Event Trigger Date.

A pending event will be cancelled automatically by the system when the account's debt no longer violates the collection criteria that sparked the event's collection process. A pending event may also be cancelled by a user at their discretion. Refer to How Are Collection Processes Cancelled for more information about how the system will cancel a collection process (and its events).

Collection Event Trigger Date

When a collection event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

How Are Collection Events Completed

A background process runs periodically (at least daily) that looks for collection events with a trigger date on or before the current date. For each triggered event, the system executes its activity and then completes it. Refer to Collection Event Activator for more information.

Copyright © 2026, Oracle and/or its affiliates.

The Last Collection Event Should Kick Off Severance Process(es)

The last collection event will typically kick off the severance process for every service agreement linked to the collection process. This will only happen if you set up the collection process template accordingly (i.e., the last event type in the process template is the kind that starts a severance process for every service agreement linked to the collection process).

> **Note**
>
> **Nominating a service agreement to sever.** Many organizations that have multiple services in arrears will NOT sever every service agreement that's in arrears. Rather, they will nominate one service agreement and use it to encourage the customer to pay for the other services. If your organization works this way, then your last collection event should call the Nominate A Service Agreement To Sever Algorithm.

How Are Collection Events Canceled?

Users can cancel a collection event at their discretion. In addition, the system can cancel a collection event when it automatically cancels a collection process. Refer to How Are Collection Processes Cancelled for the details.

The Big Picture Of Severance Process Cancellation

The topics in this section provide high level information about the cancellation of severance processes.

> **NOT_SUPPORTED**
>
> For more information refer to The Lifecycle Of A Severance Process And Its Events.

- How Are Severance Processes Cancelled?
- What Happens When A Severance Process Is Cancelled?

How Are Severance Processes Cancelled?

A severance process may be cancelled via the mechanisms described in this section.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to What Happens When A Severance Process Is Cancelled? for what happens when a severance process is cancelled.

- The Freezing Of Certain Financial Transactions Can Cancel A Severance Process
- A New Payment Plan Can Cancel A Severance Process
- A User May Cancel A Severance Process At Their Discretion
- Stopping A Service Agreement Will Cancel A Severance Process

The Freezing Of Certain Financial Transactions Can Cancel A Severance Process

Note

The system will only cancel a severance process if its severance process template indicates that **Auto Cancel** is allowed. Typically, this switch is set on all severance process templates except for the odd ones that are used to reconnect service.

The system reviews a severance process real-time whenever its service agreement's debt is reduced. Financial events that can cause service agreement debt to be reduced are:

- The cancellation of a bill segment.
- The creation of a payment segment.
- The creation of an adjustment that credits a service agreement.

Note

Real time cancellation. Unlike collection processes, the system cancels severance processes real time (i.e., there is no background process that monitors severance processes). Why are severance processes canceled real time? Because a severance process may have events that create field activities to sever service. These events need to be canceled the moment the FT is frozen, we can't wait until a background process runs. This means that if a customer pays in person for a service agreement that is pending severance, the system will cancel the process and its field activities (if any) the moment the payment is entered.

The review takes place as follows:

- **Debt class cancel criteria.** In general, the sum of all debt associated with the severance process's debt class must be less than or equal to a given threshold amount for a severance process to be cancelled. If so, the severance process is cancelled.
- Please be aware that, if a Pay Plan exists for the account and debt class, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it

Copyright © 2026, Oracle and/or its affiliates.

is compared to the threshold amount. Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

> **Note**
>
> **The above logic is not "hard coded".** The system calls the Severance Process Cancel Criteria Algorithm defined on the debt class that is associated with the severance process. This algorithm cancels a severance process if the sum of ALL service agreements in the debt class have debt less than or equal to a given threshold amount. However, because it's an algorithm, you can introduce whatever cancellation criteria you please.

- **Service agreement cancel criteria.** You can optionally introduce a special quirk to the cancellation logic. This quirk is a bit difficult to understand. To understand it, you should recall:
  - The collection event called Start Severance creates a severance process for every service agreement that is in arrears. Note: you would only use this type of collection event if you do not Nominate A Single Service Agreement To Sever.
  - If you use the Start Severance collection event, then you would want to cancel a severance process when its service agreement no longer has intolerable debt (regardless of the state of the debt class's entire debt).
- To cancel a severance process when its related service agreement no longer has intolerable debt, you should plug-in a Cancel Criteria Algorithm on your severance process templates. The system will call this algorithm if you've plugged it in.

> **Note**
>
> **Manual Creation**. A user can create a severance process for an account that does not qualify to be on severance according to the cancel criteria algorithm. For example, perhaps your cancel criteria algorithm cancels a severance process when the account's debt falls below a threshold amount. A user can create a severance process for an account whose debt is already below this threshold. Because cancellation is real time, there is no action that will cause this severance process to be canceled. When a manual severance process is created, the system executes the appropriate cancellation criteria algorithm. If the algorithm indicates that the system would have canceled this severance process, a warning is issued.

A New Payment Plan Can Cancel A Severance Process

Refer to Collection Process / Severance Process Cancellation When A Pay Plan Is Created for the details.

> **Note**
>
> **Real time cancellation.** Please be aware that the system will cancel a severance process real time when a pay plan is created that pays off enough debt.

Copyright © 2026, Oracle and/or its affiliates.

A User May Cancel A Severance Process At Their Discretion

A user may cancel a severance process at their discretion.

Stopping A Service Agreement Will Cancel A Severance Process

The system will cancel a severance process if its service agreement is stopped (i.e., when the service agreement's status becomes Stopped).

What Happens When A Severance Process Is Cancelled?

The following takes place when a severance process is canceled by the system:

- The system cancels all pending severance events and deactivates the severance process.
- If there are any field activities linked to the severance process, an optional plug-in spot defined on the installation record allows you to plug in an algorithm to cancel these field activities.
  - The base package Severance Process Cancellation Algorithm will cancel all pending field activities that were created as a result of the severance process that are not linked to a dispatched field order.
- If there are any pending field activities left associated with the severance process, it is marked to trigger the creation of a To Do entry to highlight that field activities exist for a canceled severance process. (This happens if you have not plugged in an algorithm to perform the cancellation or if the algorithm detected a condition that prevented cancellation.) To create the To Do entry, you must run the background process TD-SPRO.
- There is an optional plug-in spot defined on the severance process' template. If an algorithm is plugged-in, it is called. The base package algorithm will create a reconnect process if there are completed field activities for a cut for nonpayment severance event associated with the severance process. Refer to Severance Post Cancellation Algorithm for more information about this algorithm.

The Big Picture Of Severance Events

This section describes the various types of severance events and their lifecycle:

- How Are Severance Events Created?
- Types Of Severance Events
- Field Events Versus Office Events
- Severance Event Lifecycle
- Severance Event Dependencies and Trigger Date
- Field Events And Their Activities
- Severance Event Activation & Completion
- How Are Severance Events Canceled?

Copyright © 2026, Oracle and/or its affiliates.

How Are Severance Events Created?

Severance events may be created as follows:

- The process that completes (i.e., executes) collection events creates a severance process when it completes a "start severance process" collection event. The severance process has one or more severance event(s). The number and type of events is controlled by the severance process template associated with the severance process. Refer to The Collection Event Activator for more information about this process.
- Severance events will be created when a user creates an ad hoc severance process. The number and type of severance events is controlled by the severance process template associated with the severance process.
- An ad hoc severance event may be created and linked to an existing severance process by a user at their discretion.

> **Note**
>
> **Bottom line.** Most severance events are created by the system when it creates a severance process for delinquent service agreements. If you need to create an ad hoc severance event, you can either create a severance process whose template contains the desired event OR link the desired event to an existing severance process.

> **NOT_SUPPORTED**
>
> For more information about creating ad hoc severance processes and events, refer to How To Perform Common Severance Process Functions.

Types Of Severance Events

The following table describes the various types of severance events and what happens when they are completed:

Type Of Severance Event	What Happens In The System
Send Letter	A customer contact is created for every financially responsible person linked to the service agreement's account. It is the customer contact that causes a letter to be produced. The type of letter is defined on the customer contact's contact type. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).
Create To Do Entry	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.

Copyright © 2026, Oracle and/or its affiliates.

Create Field Activities	A field activity is created for each service point associated with the service agreement being severed. The type of activity is defined on the service point's SP type's field activity type profile.
Generic Algorithm	The algorithm defined on the event's event type is executed.
Expire Service Agreement	The service agreement is expired and, if earlier severance events created "cut for non-payment" field activities, these field activities will be used as the basis for stopping service. Refer to Finalizing Pending Stops for how the system use the meter reads on these field activities as the "stop reads" on the service agreement. Note, you can see the field activities that are used to "cut" and "stop" service by viewing the Field Activities grid on Service Agreement - Service Point.
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the collection event type.

NOT_SUPPORTED

Refer to Setting Up Severance Event Types for more information.

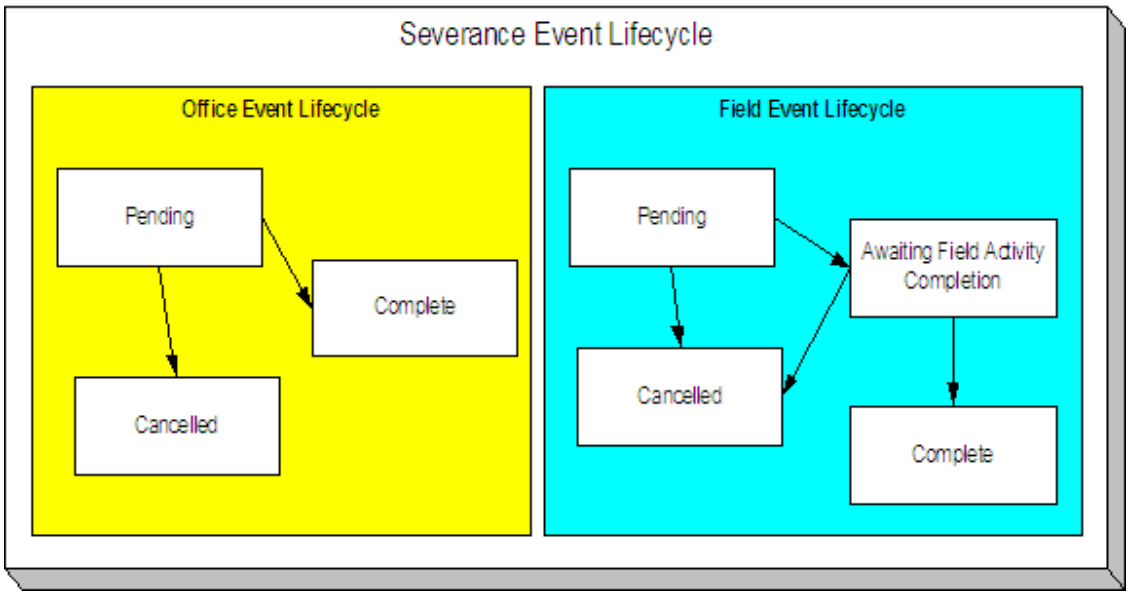
Field Events Versus Office Events

Severance events are considered either field or office events. Office events are just like collection events in that they don't involve any type of field activity in their completion. Field events, however, are not like collection events because they create one or more field activities. These activities cause these events to have a different lifecycle. Refer to Severance Event Lifecycle for more information.

Severance Event Lifecycle

The following diagram shows the possible lifecycle of a severance event:

Copyright © 2026, Oracle and/or its affiliates.



The following points explain the lifecycle of severance events of the office variety:

- Office events are initially created in the pending state.
- A pending office event becomes complete when the system sees that its trigger date is on or before the current date. At this time, the system executes the event's activity (e.g., create a letter, create To Do entry).

NOT_SUPPORTED

For more information about a severance event's trigger date, see Severance Event Dependencies & Trigger Date.

- A pending office event will be cancelled automatically by the system when the debt associated with the severance process's service agreement is sufficiently reduced. A pending office event may also be cancelled by a user at their discretion. Refer to How Are Severance Processes Cancelled for more information about how the system will cancel a severance process (and its events).

The following points explain the lifecycle of severance events of the field variety:

- Field events are initially created in the pending state.
- A pending field event becomes awaiting field activity completion when the system sees that its trigger date is on or before the current date. At this time, the system creates the field activities associated with the given event (e.g., disconnection warning, disconnect for nonpayment, etc.).

NOT_SUPPORTED

For more information about a severance event's trigger date, refer to Severance Event Dependencies & Trigger Date. For more information about the field activities that are created for a field severance event, refer to Field Events And Their Activities.

Copyright © 2026, Oracle and/or its affiliates.

- An awaiting field activity completion field event becomes complete when the system sees that its field activities are all complete or cancelled.
- A pending field event will be cancelled automatically by the system when the service agreement associated with the severance event's severance process has sufficient credits. A pending field event may also be cancelled by a user at their discretion.
- An awaiting field activity completion field event will be cancelled automatically by the system when the service agreement associated with the severance event's severance process has sufficient credits if the field activity has not been dispatched (refer to Designing Your Reconnection Procedures for information on how the system handles the situation if the field activity is completed or dispatched). An awaiting field activity completion event may also be cancelled by a user at their discretion.

Severance Event Dependencies and Trigger Date

When a severance event is created by the system, its trigger date cannot be set. This is because, unlike collection events, the trigger date on severance events can only be set when ALL of the preceding severance events on which it depends are complete. An example will help explain why this design is necessary. Consider the following example that shows a standard severance process and its events:

Event Number	Severance Event	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Preceding Events
10	Field activity - Disconnect for non-payment warning	N/A - first event	0
20	Field activity - Disconnect for non-payment	10	2
30	Create To Do entry	20	0
40	Send letter to customer	20	0
50	Expire service agreement	20	10

This severance process is meant to execute as follows:

- On the first day, generate a 48-hour warning of impending disconnection. This is a field event as most organizations deliver this warning in person.
- After 2 days (i.e., 48 hours) have passed, create a field activity to disconnect for non-payment.
- When this is completed, generate a To Do entry to let a CSR know about the cutoff. Also, send a letter to the customer.
- If after 10 days from the cutoff, we still don't have payment, expire the service agreement.

As can be seen from the above example, the later events are dependent on the completion of the field activities in the earlier events. This means that when you set up a severance event, you must indicate the events on which it depends (and the number of days after their completion that the event should be triggered).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bottom line.** The system sets the trigger date on severance events when it detects that all of its dependent events are complete (this is the responsibility of the SED background process). Refer to Calendar vs Work Days for a description of your choices in respect of how the trigger date is calculated.

Field Events And Their Activities

When the system is told to start a severance event that creates field activities (e.g., disconnect for non-payment), it will generate a field activity for every service point linked to the service agreement.

The question is, where does it get the field activity type associated with the field activities? The answer is explained below:

- Every service point has an SP type.
- Every SP type references a field activity type profile.
- A field activity type profile contains a matrix defining activity types to generate under various situations. Some of these situations are those associated with severance.

> **NOT_SUPPORTED**
>
> For more information, refer to Setting Up Field Activity Type Profiles.

Severance Event Activation & Completion

A background process runs periodically (at least daily) that looks for severance events with a trigger date on or before the current date. This process executes the activity associated with each event. If the event is an office event, the event then becomes complete. If the event is a field event, the event becomes "awaiting field activity completion" until the field activities are complete. At this time, the severance event is completed. Refer to Severance Event Activator for more information.

How Are Severance Events Canceled?

Users can cancel a severance event at their discretion. In addition, the system can cancel a severance event when it automatically cancels a severance process. Refer to How Are Severance Processes Cancelled for the details.

The Big Picture Of Write Off Processing

Copyright © 2026, Oracle and/or its affiliates.

Before you financially write-off debt, most companies go to some effort to collect the past due funds. You control exactly what happens by setting up the various write-off control tables. The topics in this section provide background information that will help you understand how the information in these control tables is used.

- How Is Debt Financially Written-Off?
- The Ramifications of Write Offs in the General Ledger
- Automated versus Manual Write Offs
- How Does The Write-Off Monitor Work?
- How Does A Write-Off Process Get Cancelled?
- How Do Collection Agency Referrals Work?

How Is Debt Financially Written-Off?

Before debt can be written-off, a write-off service agreement must exist for the account. Why? Because when you write-off a normal service agreement's debt, you are actually transferring its debt to a write-off service agreement.

A write-off service agreement is just like other service agreements in that:

- It holds debt.
- When a payment is received, the service agreement's debt is reduced.

Debt is transferred to a write-off service agreement (WO SA) from the customer's uncollectable service agreements (SAs). The following points highlight important characteristics about the uncollectable SAs and the WO SA:

- The WO SA and the uncollectable SAs should be linked to the same account (note: this isn't a strict rule, it just makes sense because an account's written off funds should be linked to the account).
- Debt may be transferred to a WO SA from any type of service agreement regardless of debt class, i.e., a WO SA can contain debt that originated in any debt class.
- When you transfer debt from the uncollectable SAs to the WO SA, the debt is removed from the uncollectable SAs (and their status becomes closed - assuming their balance becomes zero).
- If you use the system's automated write-off processing, the system will create WO SAs for you. The system's automated write-off processing can write-off revenue in a different manner than is used to write-off liabilities. Refer to The Ramifications of Write Offs in the General Ledger for more information.
- WO SAs are immune from the account debt monitor (assuming their debt class is marked as not being subject to collection activities).
- WO SAs are not billed (assuming their SA type is marked as being not billable).
- WO SAs start their life with a non-zero payoff and current balances (i.e., they have debt when first started). This debt is transferred from the normal service agreement(s) whose uncollectable debt necessitated the creation of the WO SA.
- If the customer pays off the write-off debt, the WO SA remains active in case you ever need to write-off debt in the future. If you don't like the WO SA remaining active after it's paid off, you can indicate on the WO SA's SA Type that it is a "one time charge", this will cause the WO SA to be automatically closed when it's paid off.

Copyright © 2026, Oracle and/or its affiliates.

- You can transfer additional uncollectable debt to the WO SA.

> **Note**
>
> **Bankruptcy write-offs.** If you have to write-off debt because a customer declares bankruptcy, everything stated above is true. The only thing you have to do is use a different SA type for bankruptcy write-offs as compared to "normal" write-offs. On the bankruptcy write-off SA type, simply leave the payment segment type blank - this way the system will never distribute a payment to the bankrupt debt (because bankrupt debt is legally uncollectable).

The Ramifications of Write Offs in the General Ledger

> **Warning**
>
> If you practice cash accounting, refer to Cash Accounting and Write-Offs.

When you write-off unpaid debt, you shouldn't book it all to a write-off expense account. Why? Because the debt that you're writing off typically contains both revenue and liabilities. At write-off time, you typically want to:

- Book the written off revenue to a write-off expense account, and
- Reduce the liabilities (you don't owe the liability if you don't get paid).

Consider the following example of a simple electric service agreement with two financial transactions:

Event	GL Accounting
Customer is billed	A/R 1000 Revenue <900> State Tax Payable - Taxing State - California <80> City Tax Payable - Taxing City - San Francisco <20>
Customer is levied a late payment charge	A/R 50 Late Payment Revenue <50>

After these two financial transactions are booked, the customer has debt of $1050. Of this $1,050; $950 is revenue and $100 is liability (money you owe the taxing authorities).

If the customer doesn't pay, you will eventually have to write-off this debt. Most organizations would issue the following types of financial transactions to do this:

Event	GL Accounting

Copyright © 2026, Oracle and/or its affiliates.

Write-off the bill	Write-off Expense 900 State Tax Payable - Taxing State - California 80 City Tax Payable - Taxing City - San Francisco 20 A/R <1000>
Write-off the late payment charge	Write-off Expense 50 A/R <50>

Notice in the above transactions, the two separate revenue accounts are written off by booking to an expense account. However, the liability accounts are reversed. Why is revenue treated differently from liabilities at write-off time? There's a good reason for it (if you're an accountant), for the time being, just accept that this is how it works.

And finally, we need to worry about what happens if the customer eventually pays off their written off debt. If this happens, most organizations would pay off the write-off first, and, if there was still money left, they'd reimburse the taxing authorities. If we assume the customer pays off the entire written off debt, the following financial transactions would be issued:

Event	GL Accounting
Pay off the written off debt	Cash 900 Write-off Expense <900>
Reinstate the liabilities	Cash 100 State Tax Payable - Taxing State - California <80> City Tax Payable - Taxing City - San Francisco <20>

While the reinstatement of liabilities at payment time is possible in the system, the ramifications of doing such make this approach impracticable (the ramifications are a) if the check bounces, we would not be able to reduce the liabilities, and b) if there was a partial payment of the liabilities, the remaining unpaid amount could get written down). Therefore, when a write-off is paid the following financial transactions should be issued:

Event	GL Accounting
Pay off the written off debt	Cash 900 Write-off Expense <900>
Reinstate the liabilities	Cash 100 Reinstated liabilities <100>

Notice that rather than reinstating the individual liabilities, we simply reinstate all liabilities into a single account. This means your accountants will have to distribute this money to the appropriate liabilities manually.

So, how do we achieve the above in the system? This explanation is a little complicated, but it'll make sense if you keep the above financial transactions in mind:

- First of all, you'll need two different SA types - one to hold the written off revenue and another to hold the reduced liabilities.
  - On the SA type that holds written off revenue, indicate that it is not billable, indicate that it cannot have excess credits, and give it a high payment distribution priority. The distribution code on this SA type should reference your Write-off Expense account.

Copyright © 2026, Oracle and/or its affiliates.

- On the SA type that holds the reduced liabilities, indicate that it is not billable, indicate that it cannot have excess credits, and give it a high payment distribution priority. The distribution code on this SA type should reference a the "reinstated liabilities" GL account.

Next, you need to understand how the system's standard write-off logic works:

- The system accumulates the distribution codes from GL details associated with recent financial transactions linked to the service agreement being written-off.
- When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreement(s). The number and type of service agreements to which the bad debt is transferred is defined on the distribution codes. Refer to Setting Up Distribution Codes for how to define the type of write-off service agreement associated with a distribution code. In our example, we'd need the two SA types described above - one for the revenue accounts, the other for the liability accounts.
- At write-off time, for those distribution codes associated with revenue, the system will create a transfer adjustment from the normal service agreement to the write-off revenue service agreement. This will reduce (credit) the receivable on the normal service agreement and increase (debit) the expense account defined on the write-off revenue service agreement.
- However, if we do the above for the distribution codes associated with liabilities, we have a problem. The problem is a bit hard to explain unless you understand tax accounting, but it basically comes down to this - if we simply transfer the portion of the receivable balance associated with the liabilities to the write-off liability SA, we will always be debiting the distribution code defined on the SA type. This isn't correct because we really want to debit the liability account (and reference the characteristic type and value from the original credit) when we reduce the liability. So how do we do this? For those distribution codes associated with liabilities, you need to indicate that you want to override the distribution code on the "transfer to" side of the transfer adjustment with the distribution code / characteristic type / characteristic value that was originally booked. Refer to Setting Up Distribution Codes for how to indicate you want to override the distribution code at write-off time. If you do the above, then at write-off time the transfer adjustment will reduce (credit) the receivable on the normal service agreement and increase (debit) the original liability accounts from the original financial transactions.

If you followed the above, you'll see that we now have everything debited and credited appropriately. And, if a payment materializes for the written off debt, we will simply debit cash and credit the distribution code on the respective SA (either Write Off Expense or Reinstated Liabilities).

> **Note**
>
> **Batch and real-time write-offs may use the above processing.** The above logic is executed real time when a user writes off debt using the write-off transaction (assuming the base package write off algorithm is plugged into the account's customer class). The above logic is executed in batch when a write-off event that references a Write Off Using Distribution Codes event type is executed. Write-off events are described in detail below.

Copyright © 2026, Oracle and/or its affiliates.

Automated versus Manual Write Offs

The system will automatically create write-off SAs and transfer uncollectable debt to them during the automated write-off processing described below.

If necessary, you can write-off debt outside of the automated write-off process using either of the following methods:

- You can transfer bad debt from any service agreement to a write-off service agreement using a transfer adjustment.
- You can use the write-off transaction to write-off debt real-time. When this transaction is used, the system executes the logic embedded in the Write Off Method algorithm that's plugged in on the account's customer class.

How Does The Write-Off Monitor Work?

This section describes how the Write Off Monitor uses your write-off criteria and write-off process templates to collect overdue debt.

- Different Write-Off Criteria For Different Customers And Different Debt
- When Is Debt Monitored For Write Off Purposes?
- Attempt To Close The SA Before Creating A Write Off Process
- What Happens When A Write-Off Process Is Started?

Different Write-Off Criteria For Different Customers And Different Debt

Consider the following:

- You probably have different write-off criteria for different customer segments. For example, customers with large bills probably have strict criteria, whereas you're probably more lenient with small customers (or vice versa). You differentiate your customers in respect of the collection process via a **collection class code on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You probably have different write-off criteria for different classes of debt. For example, if a customer has both regulated and unregulated debt, you probably have commission-imposed criteria to be applied to the regulated debt, but you have control over how to write-off unregulated debt. You differentiate your debt in respect of the collection process via a write-off **debt class on the customers' service agreements** (note the write-off debt class is actually defined on the SA type and every service agreement has a SA type).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Write Off Debt Class vs. Regular Debt Class.** It's important to be aware that a SA type references both a regular debt class and a write-off debt class. The regular debt class controls the collection criteria applied against an account's service agreements. The regular debt class is also used to segregate an account's outstanding balance on several queries in the system. The write-off debt class controls the write-off criteria applied against an account's stopped service agreements. The reason the system supports two different debt classes is because you may categorize your service agreements differently when you try to collect overdue debt versus when you write-off debt.

Given the above, you should understand that different write-off criteria will exist for every combination of collection class and write-off debt class. If you're confused, then consider the following matrix:

SA's Write-Off Debt Class	**Account's Collection Class: Commercial Customer**	**Account's Collection Class: Residential Customer**
Regulated	N/A - there is no regulated, commercial customer debt.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains, create the default write-off process for regulated debt.

Copyright © 2026, Oracle and/or its affiliates.

Unregulated	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. Attempt to transfer debt to another active service agreement linked to the account. If the debt / credit is < $10 and > $-10, write down the debt using a write-down adjustment. If the debt / credit is <= $-10, create an A/P adjustment to refund the credit to the customer. If debt still remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for unregulated commercial debt.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains, create the default write-off process for unregulated residential debt.

Notice that each cell in the matrix has the same pattern:

- The system first attempts to reduce the SA's current and payoff balances to zero using the following methods (assuming you have set up the write-off control appropriately):
  - Sync the current balance with the payoff balance. If the SA's payoff balance is zero, this will cause the current balance to become zero and therefore close the SA.
  - If there's a debit balance, transfer the debt to any pending start or active SA in the same write-off debt class.
  - If there's a credit balance, transfer the debt to any non-closed / non-cancelled SA in the same write-off debt class
  - If the remaining debit / credit balance is within a user-defined tolerance (this is defined on the respective algorithm on the write-off control), create an adjustment to write-down the small balance.
  - If a credit balance remains, create an A/P adjustment to refund the balance with a check (the adjustment type is defined on the respective algorithm on the write-off control).
- All of the above points will cause the SA to close. If debt remains, the system starts some type of write-off process. The type of process is dependent on the respective criteria. What differentiates one write-off criteria from another is its priority. The higher priority criteria will be compared first. If the customer / debt meets the criteria, the write-off process is initiated; no further comparisons are performed.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about maintaining this matrix, refer to Setting Up Write-off Control.

When Is Debt Monitored For Write Off Purposes?

The write-off monitor only reviews a service agreement when the following conditions are true:

- The service agreement is stopped and reactivated.
- If the service agreement is a "billable charge" SA (as identified on its SA type), all of its billable charges must appear on a bill segment AND the bill segment's bill's due date plus grace period must be on or before the business date.
- If the service agreement is not a "billable charge" SA AND it is billable (as identified on its SA type), the SA must have a closing bill segment (i.e., it must be final billed) and the bill segment's bill's due date plus grace period must be on or before the business date.
- If the service agreement is a sub SA, its master SA must abide by the above conditions.
- If the service agreement is not billable, it is possible that adjustments, which affect the SA's debt, exist. The write-off monitor will only review a non-billable SA if all FTs for this SA that have been marked to include on a bill have been swept onto a bill and the bill for any of these FTs has a bill due date plus grace period on or before the business date.

Note

Postponing write-off processing. You can prevent the write-off process from processing an eligible service agreement by populating the account's C&C Postpone Date with a future date.

Attempt To Close The SA Before Creating A Write Off Process

Before the write-off monitor creates a write-off process for a stopped and reactivated service agreement, it attempts to reduce the service agreement's debt to zero using all of the following methods:

- If the account has active service agreements, it will transfer the finaled debt to a pending start or active service agreement.
- If the debt or credit amount on the service agreement is small, the system will generate an adjustment to 'write it down' (or up in the case of a small credit).
- If the service agreement has a large credit amount, the system will generate an A/P adjustment (resulting in a check being sent to the customer).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Plug-in algorithms do the work.** Algorithms that are plugged-in on the write-off control responsible for managing the service agreement's debt actually perform the above effort. You can customize these algorithms to behave exactly how your collections staff desires.

If the algorithms responsible for the above effort are successful in reducing the service agreement's debt to zero, then the service agreement closes and will not be subject to write-off processing. If the above algorithms don't result in the service agreement's debt being reduced to zero, a write-off process will be started (as describe below).

What Happens When A Write-Off Process Is Started?

When you define write-off criteria, you must define the write-off process template to use if the criteria are violated. The system uses this template to create the account-specific write-off process.

Every stopped or reactivated service agreement that is part of the offending write-off debt class will be linked to the write-off process.

Also linked to the write-off process will be one or more write-off events. These events are meant to prod the customer. You define the types of events and when they are generated when you set up your write-off process templates.

It's important to note that all of the write-off events will be created when the write-off process is created. Each of these write-off events contains a trigger date. The trigger date of the first event(s) will typically be the current date. The trigger date of the other events will be in the future. Refer to Calendar vs Work Days for a description of how the trigger date is calculated.

A separate process, Activate Write-off events, is responsible for activating write-off events whose date is on or before the current date. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do to an operator, refer debt to a collection agency, etc.)

> **Note**
>
> **Multiple write-off processes may be kicked off.** It's important to be aware that if an account's service agreements reference multiple write off debt classes, a write-off process will be started for each offending write off debt class that has stopped or reactivated service agreements.

> **NOT_SUPPORTED**
>
> For more information about write-off process templates, see Setting Up Write Off Process Templates. For more information about write-off events, see The Big Picture Of Write-off Events. For more information about how a write-off process is cancelled, see How Does A Write-Off Process Get Cancelled?.

Copyright © 2026, Oracle and/or its affiliates.

How Does A Write-Off Process Get Cancelled?

The system "removes" a service agreement from a write-off process when its status becomes closed (i.e., when its balance is zero). When all service agreements are removed, the system cancels all pending write-off events and deactivates the write-off process. When the write-off process is deactivated, all collection agency referrals associated with the write-off process are cancelled.

> **Note**
>
> **Removing closed service agreements from a write-off process.** Service agreements aren't actually removed from the process. Rather, they are inactivated so a proper audit exists.

How Do Collection Agency Referrals Work?

The following points describe how collection agency referrals work.

- A write-off process has one or more events. One type of event causes overdue debt to be referred to a collection agency.
- When a referral write-off event is activated, the system marks the event for processing by the event's Collection Agency Referral Algorithm (refer to Setting Up Write Off Event Types for more information).
- The next time the Collection Agency Referral process executes (the frequency is dependent on your background process schedule), it will refer the process' debt to a collection agency. The specific agency to which the debt is referred is controlled by the event type's Collection Agency Referral Algorithm. The sample algorithm supplied with the system simply refers debt to the collection agency with the least amount of referred debt. If you prefer different logic, you must write your own algorithm.
- Regardless of the manner in which a collection agency is selected for an account's debt, the referral involves the creation of a collection agency referral history record. Refer to Collection Referral for more information.
- A collection agency referral history record is linked to an account. It contains the amount of debt referred to the collection agency. It is the creation of this record that, in turn, triggers the interface of information to the collection agency. The method used to interface the information to the agency is defined on the collection agency's record. Refer to Setting Up Collection Agencies for more information.
- If the collection agency is successful in obtaining the funds, simply add a payment. If the payment causes the SA's balance to become zero, the system will automatically close the service agreement. When the system closes a service agreement, it is "removed" from the write-off process. When a write-off process no longer contains active service agreements, the system cancels the write-off process. When a write off process is cancelled, all collection agency referrals are automatically cancelled.
- Collection agency referrals get cancelled by the creation of a new collection agency referral history record (with a type of cancel). This record will be interfaced to the agency in the same manner used to interface a new referral (see above).

Copyright © 2026, Oracle and/or its affiliates.

- If the collection agency is not successful in obtaining your funds after a given amount of time, you probably want to cancel the referral and write-off the debt. The cancellation of the referral will happen automatically if you design your write-off process to generate a collection agency cancellation X days after the referral. Refer to Setting Up Write Off Process Templates for how to do this. You can cancel a referral manually by simply creating a new collection agency referral history record (with a type of cancel).

> **NOT_SUPPORTED**
>
> When you enable the Control Central alert algorithm, C1-COLL-REF, an alert displays when an account has an active collection agency referral. This algorithm is plugged-in on the installation record.

The Big Picture Of Write-off Events

This section describes the various types of write-off events and their lifecycle.

- How Are Write-off Events Created?
- Types Of Write-off Events
- Write-off Event Lifecycle
- Write-off Event Trigger Date
- How Are Write-off Events Completed
- The Last Write-off Event Should Write Off All Debt Associated With All SAs
- How Are Write-off Events Canceled?

How Are Write-off Events Created?

Write-off events may be created as follows:

- The Write-Off Monitor creates a write-off process when an account has unpaid, final billed service agreements. The write-off process has one or more write-off event(s). Refer to How Does The Write-Off Monitor Work? for more information about how the system creates write-off processes and their events.
- Write-off events are created when an operator creates an ad hoc write-off process. The number and type of events is controlled by the write-off process template defined when the write-off process is created.
- An ad hoc write-off event may be created and linked to an existing write-off process by an operator at their discretion.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bottom line.** Most write-off events are created by the system when it creates a write-off process for unpaid, finaled service agreements. If you need to create an ad hoc write-off event, you can either create a write-off process using a template that contains the desired event OR link the desired event to an existing write-off process.

> **NOT_SUPPORTED**
>
> For more information about creating ad hoc write-off processes and events, refer to How To Perform Common Write-off Maintenance Functions.

Types Of Write-off Events

The following table describes the various types of write-off events and what happens when they are completed:

Type Of Write-off Event	What Happens In The System
Affect Credit Rating/Cash-Only	An account credit rating demerit record is created. The number of demerits is defined on the write-off event type.
Cancel Agency Referral	All collection agency referrals associated with the write-off process will be cancelled.
Refer to Agency	The debt associated with the SAs linked to the write-off process will be referred to a collection agency.
Send Letter	A customer contact is created for every financially responsible person linked to the account. The customer contact causes a letter to be produced. The type of letter is defined on the customer contact type control table. The recipient of each letter is defined on Account / Person (those persons marked as **Receiving Notifications**).
Send To Do	A To Do entry is created. Refer to The Big Picture of To Do Entries for more information about To Do entries.

Copyright © 2026, Oracle and/or its affiliates.

Write Off using Distrib Code	This type of event is used to write-off bad debt in accordance with the distribution codes associated with the financial transactions that caused the debt in the first place. You'd use this method for example if you want to write-off revenue differently than you write-off liabilities. The system accumulates the distribution codes from GL details associated with recent financial transactions linked to each write-off service agreement. When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreement. The type of service agreements to which the debt is transferred is defined on the distribution codes.
Write Off using SA Type	The service agreements linked to the process will be written-off by transferring their debt to a new or existing write-off service agreement. Note: the SA type of the write-off service agreement is defined on the algorithm defined on events of this type.
Generic Algorithm	The system calls the algorithm defined on the write-off event type. This type of event is used when what you need to do isn't handled by one of the above event types.

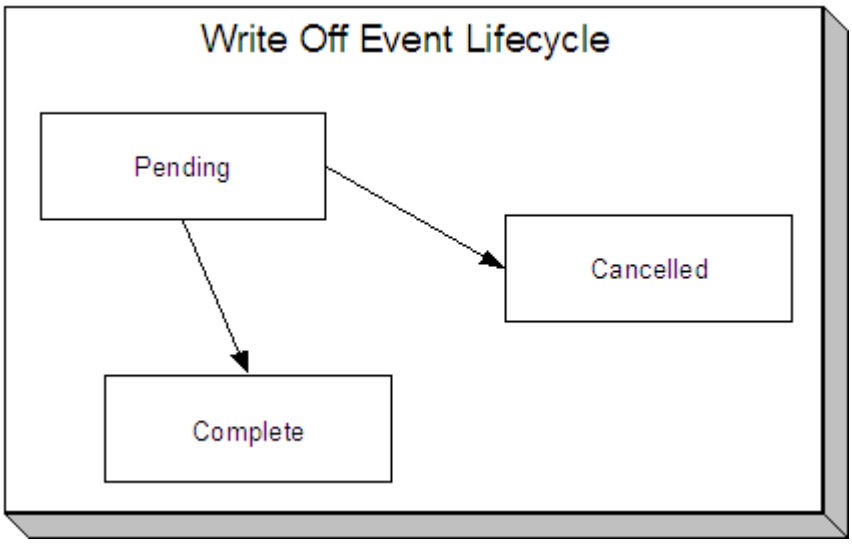
NOT_SUPPORTED

Refer to Setting Up Write Off Event Types for more information.

Write-off Event Lifecycle

The following diagram shows the possible lifecycle of a write-off event:

Copyright © 2026, Oracle and/or its affiliates.



Write-off events are initially created in the pending state.

When the system sees a pending event with a trigger date on or before the current date, the system executes the event's activity and completes the event.

NOT_SUPPORTED

For more information about a write-off event's trigger date, see Write-off Event Trigger Date.

A pending event will be cancelled automatically by the system when the service agreements linked to the process are all closed (i.e., they no longer have debt - either because it was paid or transferred to a write-off service agreement). A pending event may be cancelled by an operator at their discretion.

Write-off Event Trigger Date

When a write-off event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

How Are Write-off Events Completed

A background process runs periodically (at least daily) that looks for write-off events with a trigger date on or before the current date. For each triggered event, the system executes its activity and then completes it. Refer to Write Off Event Activator for more information.

The Last Write-off Event Should Write Off All Debt Associated With All SAs

The last write-off event will typically transfer all debt from the service agreement's linked to the process to a write off service agreement (linked to the account). This will only happen if you set up the write-off process template accordingly (i.e., the last event in the write-off process

Copyright © 2026, Oracle and/or its affiliates.

template is the kind that writes off debt for every service agreement linked to the write-off process).

How Are Write-off Events Canceled?

The system removes a service agreement from a write-off process when its status becomes closed (i.e., when its balance is zero). Be aware that any type of financial event could cause an SA's balance to fall to zero (e.g., the creation of an adjustment, the application or cancellation of a payment, the cancellation of a bill, ...). When all service agreements are removed, the system cancels all pending write-off events and deactivates the write-off process.

> **Note**
>
> **Real time cancellation.** Unlike collection processes, the system cancels write-off processes real time when the service agreement becomes closed (i.e., there is no background process that monitors write-off processes).

Besides the automated cancellation process, an operator may cancel a write-off event at will.

> **Note**
>
> **Removing closed service agreements from a write-off process.** Service agreements aren't actually removed from the process. Rather, they are inactivated so a proper audit exists.

Calendar Versus Work Days

When you set up your collection, severance and write-off process templates, you supply information that controls how the system determines the trigger date of each event in the related process. There are two different mechanisms for doing this:

- When you set up your severance process templates, you must define the number of days between each event. For example, the second event (send cutoff warning) may need to be triggered 7 days after the first event (send reminder letter).
- When you set up your collection and write-off process templates, you must define the number of days after the start of the process when each event should be triggered. For example, the second event (send cutoff warning) may need to be triggered 7 days after the start of the collection process.

The system uses this information in conjunction with the account's division's work calendar when it allocates a trigger date to the various collection, severance, and write-off events in your processes. The system offers you the following choices in respect of how it calculates an event's trigger date:

- You can indicate that the trigger date should be set to the next possible workday. For example, if you indicate that the second event is triggered 7 days after the first event, the system will add 7 days to the first event's completion date. It then checks if this is a

Copyright © 2026, Oracle and/or its affiliates.

workday (and not a holiday), if so, this is the trigger date of the event; if not, it assigns the trigger date to the next workday.

- You can indicate that the trigger date should be calculated by counting workdays. For example, if you indicate that the second event is triggered 7 days after the first event, the system will count 7 workdays (using the account's division's work calendar), and set the trigger date accordingly.

You must define which of the above methods is used in the following processes:

- Account Debt Monitor (ADM). Refer to The Account Debt Monitor for more information.
- Collection Event Trigger (CET). Refer to The Collection Event Activator for more information.
- Severance Event Set Trigger Date (SED). Refer to Set Trigger Date for more information.
- Write-off Monitor (WPM). Refer to The Write Off Monitor for more information.

The Big Picture Of Payment Arrangements and Pay Plans

The topics in this section describe two different mechanisms that allow a customer to payoff overdue debt in installments.

- The Big Picture Of Payment Arrangement Requests
- The Big Picture Of Pay Arrangements
- The Big Picture Of Pay Plans
- Setting Up Pay Plan Control Tables

The Big Picture Of Payment Arrangement Requests

A Payment Arrangement Request is used to manage the process of starting a payment arrangement. It is a step-by-step process that can include:

- Checking the account’s eligibility for payment arrangement
- Capturing the terms of the payment arrangement such as the included service agreements, down payment amount, number of installments and installment amount
- Evaluating whether or not the payment arrangement request requires approval
- User approval / rejection of the payment arrangement request
- Monitoring for the down payment, if applicable
- Creating the payment arrangement service agreement

The topics in this section describe how payment arrangement requests work.

- Lifecycle Of A Payment Arrangement Request
- Payment Arrangement Eligibility
- Payment Arrangement Request Approval
- Payment Arrangement Down Payments
- Setting Up Payment Arrangement Request Control Tables
- Automatically Cancelling Inactive Payment Arrangement Requests

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle Of A Payment Arrangement Request

The base product provides a Payment Arrangement Request (**C1–PARequest**) business object that supports the functions needed to create a payment arrangement. Its lifecycle allows for managing and monitoring of the preqrequisites of a payment arrangement:

- The first step in the payment arrangement request process is typically to check that the account is eligible for a payment arrangement. One or more eligibility criteria are executed and an overall assessment is made on the results, to determine the account's eligibility/ ineligibility. The request goes into the **Eligibility Checked** state. The results are shown on the user interface, for the user to determine the next step.
  - If the account is ineligible for a payment arrangement, the request process ends. The payment arrangement request goes into the **Denied** state.
  - If the account is eligible, the process can proceed with setting up the terms of the payment arrangement. The agreed upon terms are submitted and the request proceeds to the next step. If there are no prerequisites (approvals, down payment etc,), the request can automatically go to the **Granted** state, where the payment arrangement service agreement is created
- A payment arrangement request may be subject to prior approval if it meets certain criteria — e.g. the payment arrangement amount exceeds a threshold amount. The payment arrangement request goes into the **Approval In Progress** state, where an authorized user will need to review the request and **Approve** or **Reject** it accordingly.
  - If the user approves the request, the request will proceed to the next step — i.e. either another prerequisite (such as a down payment) or granting the request.
  - If the user rejects the request, the request will automatically go to the **Denied** state.
- If a down payment must be paid prior to starting the payment arrangement, the request will stay in the **Waiting for Down Payment** state until the payment is received or until the down payment due date.
  - If the down payment is paid (i.e. according to implementation-specific criteria for considering it paid), the payment arrangement request goes into the **Granted** state, where the payment arrangement service agreement is created. The account's SA balances to include in the payment arrangement are recalculated, based on the actual down payment that was received.
  - On the other hand, if the down payment is not received or is received but insufficient, the payment arrangement request goes into the **Unpaid Down Payment** state.
- A payment arrangement in a **Waiting for Down Payment** or **Approval In Progress** state can be **Withdrawn**. This is for cases where a customer decides to not proceed with the payment arrangement.
- Any issues encountered during the recalculation of the SA balances or during the creation of the payment arrangement service agreement can put the request into the **Error** state.

Refer to the Payment Arrangement Request business object, for more details on these lifecycle rules.

Payment Arrangement Eligibility

An account's eligibility for payment arrangement is typically checked at the start of the request process. Based on the results of these eligibility checks, a user can determine whether or not the payment arrangement request can proceed.

Copyright © 2026, Oracle and/or its affiliates.

Since the eligibility rules can vary across implementations, the base product provides plug-in spots on the Payment Arrangement Request Type (**C1–PARequestType**) business object for eligibility checking:

- The **PA Request Type – Eligibility Criteria** algorithm spot is responsible for checking specific conditions or activity on the account, to determine payment arrangement eligibility. Algorithms of this type commonly return a simple Pass/Fail result. The base product supplies a sample algorithm that checks the account's past broken payment arrangements.
- The **PA Request Type — Criteria Assessment** algorithm spot is responsible for taking the results of the execution of **PA Request Type – Eligibility Criteria** algorithms and determining the overall result, which is whether or not the account is eligible for payment arrangement. The base product supplies two sample algorithms for this spot:
  - One algorithm type checks for at least one Fail result from the **PA Request Type – Eligibility Criteria** algorithms and if it finds one, returns an *Ineligible* result
  - Another algorithm type simply returns an *Eligible* result. This is for implementations that do not need **PA Request Type – Eligibility Criteria** algorithms. This default algorithm would be the only algorithm to plug in.

.

Payment Arrangement Request Approval

A payment arrangement request may require prior approval, based on the terms specified on the request. For instance, if the total payment arrangement amount exceeds a set threshold amount, the request may have to be routed to another user for approval. That user will need to review the request and Approve or Reject accordingly.

Since the approval requirements are implementation-specific, the base product provides a **PA Request Type — Determine Approval Requirement** plug-in spot on the Payment Arrangement Request Type (**C1–PARequestType**) business object. Algorithms of this type return an indicator of whether or not approval is needed. If the algorithm indicates that approval is needed, the payment arrangement request will go into the **Approval In Progress** state until a user reviews and approves/rejects the request.

Your implementation must configure an algorithm with your specific approval criteria. The base product does not currently supply an algorithm type for this spot.

Payment Arrangement Down Payments

A down payment may be required/applicable as a prerequisite for starting the payment arrangement.

A setting on the Payment Arrangement Request Type controls whether down payments are required/optional/not applicable. If down payments are required or optional, the payment arrangement request type defines the down payment options, which are percentages of the total payment arrangement amount or a flat amount.

The down payment due date is calculated based on settings in the Payment Arrangement Request Type – e.g. number of days till due.

The down payment is monitored for the actual payment:

- If the down payment is paid, the request is granted. The SA balances are recalculated based on the down payment amount that was paid and the payment arrangement service agreement is created.

Copyright © 2026, Oracle and/or its affiliates.

- If the down payment is unpaid or is insufficient (based on the implementation's rules), the request is denied and the payment arrangement service agreement will not be created.

Refer to the Payment Arrangement Request Type (**C1–PARequestType**) business object for more details on down payment option configuration and to the Payment Arrangement Request (**C1–PARequest**) business object, to see the monitor algorithm that checks if the down payment was made.

Setting Up Payment Arrangement Request Control Tables

This section describes the control tables needed to set up payment arrangement requests.

- Setting Up Payment Arrangement Request Types
- Setting Up Payment Arrangement Request Reasons
- Setting Up Payment Arrangement Request Status Reasons

Setting Up Payment Arrangement Request Types

A payment arrangement request type defines properties that control how a request for payment arrangement is processed.

Refer to The Big Picture Of Payment Arrangement Requests for an overview of payment arrangement request functionality.

To maintain payment arrangement request types, open **Admin**, and then **Credit & Collection**, and then **Payment Arrangement Request Type**, and then **Search**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information on the fields.

Setting Up Payment Arrangement Request Reasons

A payment arrangement request can capture the specific reason why a customer is asking to go on a payment arrangement. This reason is optional if using the base-owned Payment Arrangement Request (**C1–PARequest**) business object and the Payment Arrangement Request Process Flow Type (**C1-PAREQUEST**) on your Payment Arrangement Request Types.

To use Payment Arrangement Request Reasons, configure your specific values on the **C1-PARequestReasonLookup** extendable lookup. No base values are supplied for this lookup.

Setting Up Payment Arrangement Request Status Reasons

The Payment Arrangement Request (**C1–PARequest**) base business object provides the option to specify the reason for the request going into the **Denied** or **Canceled** state.

A few base-owned status reasons are supplied. To add other reasons for payment arrangement request cancellation/denial, go to the Status Reason portal and update the **C1–PARequest** business object's status reasons for the Denied and Canceled states.

Copyright © 2026, Oracle and/or its affiliates.

Automatically Cancelling Inactive Payment Arrangement Requests

The system is delivered with a batch process, Cancel Inactive PA Requests (C1-ACPAR), that can be used to cancel payment arrangement requests that have been open for more than a specified number of days. Refer to the batch control description for more detail.

The delivered process is built using the Plug-in Driven Background Processes framework. Your implementation can introduce alternate logic, by duplicating this batch control and replacing the select and/or process records algorithms. For example, you can introduce a process record algorithm to create a To Do Entry instead of cancelling the payment arrangement request. Adjust the batch parameters according to your custom logic, if needed.

The Big Picture Of Pay Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in **billed** installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

> **Note**
>
> **Nomenclature.** Some people refer to payment arrangements as "current bill plus" agreements because the customer's bills contain charges for both their current debt plus their installment amount. After the customer has paid off their overdue debt, the customer's bill only contains charges for their current debt.

The topics in this section describe how to set up a payment arrangement and how the system monitors the ongoing arrangements.

- Creating Payment Arrangements
- Installment, Payoff and Current Amounts
- Monitoring Payment Arrangements

Creating Payment Arrangements

When you create a payment arrangement, you are actually creating a service agreement. This service agreement is just like other service agreements in that:

- It holds debt.
- It is periodically billed.
- When a payment is received, the service agreement's debt is reduced.
- If the service agreement becomes delinquent, a collection process is initiated to collect the overdue debt.

Debt is transferred to a payment arrangement service agreement (PA SA) from the customer's delinquent service agreements (SAs) at the inception of the payment arrangement.

When you transfer delinquent debt from the delinquent SAs to the PA SA, the debt is removed from the delinquent SAs. If you transfer all debt from the delinquent SAs, the customer will no

Copyright © 2026, Oracle and/or its affiliates.

longer be in arrears in a given debt class (and if the customer is no longer in arrears, active collection and severance processes will be cancelled).

> **Note**
>
> **Use the Payment Arrangement Request Transaction.** You could do the above functions by adding a new service agreement and creating transfer adjustments. However, this is tedious. Rather, use the Payment Arrangement Request process. This process is recommend when requests for payment arrangements require eligibility checking and other prerequisites such as down payments and approvals. When a payment arrangement request is granted, it creates the PA SA, transfers debt to it, and sets up the installment amount.
>
> You can also create the payment arrangement service agreement directly using the Payment Arrangement transaction if you don't require any prior eligibility checks, down payments or approvals. This transaction is also used if you need to break or cancel the payment arrangement.

Installment, Payoff and Current Amounts

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When you set up a payment arrangement service agreement (PA SA), you transfer delinquent debt to the PA SA using transfer adjustments. After moneys are transferred, the system sets the PA SA's current balance to zero. At this point, neither the original service agreements nor the PA SA have delinquent debt. If the customer neglects to pay their payment arrangement, the PA SA will fall into arrears and a collection process will ensue. If the customer neglects to pay their previously delinquent SAs, they will again fall into arrears and a collection process will ensue.

PA SAs start their life with a non-zero payoff balance (i.e., they have debt when first started). This debt is transferred from the normal service agreement(s) whose outstanding debt necessitated the creation of the PA SA.

The installment amount that the customer is billed is determined by the number of installments used to payoff the debt. For example, if the customer owes $500 on their electric and water service agreements and they want to pay this off in 10 installments, you'd set up the installment amount to be $50. The installment amount is saved on the PA SA's recurring charge amount. If the customer again falls into arrears on their normal service agreements, you can transfer additional delinquent debt to the PA SA. You can also change the installment amount as needed.

A PA SA's payoff balance typically differs from its current balance. The payoff balance is the amount of debt remaining to be paid off under the terms of the payment arrangement. The current balance is the installment amount that has been billed but not paid. For example, a customer who is paying off $500 with 10 installments of $50 would have an initial payoff balance of $500 and a current balance of $0. After the first bill, the PA SA would still have a

Copyright © 2026, Oracle and/or its affiliates.

payoff balance of $500, but its current balance would be $50. When the customer pays, the PA SA's payoff balance would fall to $450 and its current balance would return to $0.

The following table contains a financial example of a customer who sets up a payment arrangement to payoff $1,000 of debt in $10 installments.

Event	Normal SA's GL Accounting	PA SA's GL Accounting	Normal SA's Current Balance	Normal SA's Payoff Balance	PA SA's Current Balance	PA SA's Payoff Balance
Prior to creation of payment arrangement	N/A	N/A	1000	1000	N/A	N/A
Transfer debt from normal SA(s) to PA SA	Xfer 1000 A/R <1000>	PA A/R 1000 Xfer <1000>	0	0	1000	1000
Set current balance to zero on PA SA	N/A	N/A	0	0	0	1000
Customer is billed ($50 for new debt and $10 of payment arrangement debt)	A/R 50 Revenue <50>	N/A	50	50	10	1000
Customer pays $60	Cash 50 A/R <50>	Cash 10 PA A/R <10>	0	0	0	990

When the customer pays off the payment arrangement debt, the system automatically closes the PA SA after it final bills (assuming the PA SA's SA type references a bill segment type that has a bill segment creation algorithm of Recurring Charge With Auto Stop).

Monitoring Payment Arrangements

The PA SA should belong to its own debt class (let's call it Payment Arrangement Debt) so that you can have stricter collection criteria for payment arrangement debt (as compared to normal SAs). Because there will be a new debt class, there will be a unique collection class control (CCC) for payment arrangements. This CCC will have debt criteria associated with payment arrangement debt. If these criteria are violated, we will kick off a collection process that should have 1 collection event - Start Severance.

The severance process template for PA SAs will have 1 severance event that calls the Break Payment Arrangement Event algorithm. This algorithm does the following:

- Cancels ALL adjustments that were used to transfer the debt to the payment arrangement (identified by the XFER adjustment type on the PA SA's SA type). When these are cancelled, the original arrearage will be reinstated under the original SAs - this debt should be rather old by this point.
- Syncs up current balance with payoff balance on the PA SA.

Copyright © 2026, Oracle and/or its affiliates.

- Makes the PA SA pending stop (SA activation will stop the SA when it next runs).
- If there is a credit left on the PA SA (because payments were made against the arrangement), the credit will be distributed amongst the account's debt using the standard distribution algorithm. Because the payment arrangement debt that was reinstated should be rather old, it should get relieved first. This relief will occur via transfer adjustments from the PA SA to the original SAs.
- If there is a debit left (e.g., because LPC were issued or some other type of adjustment was created by an operator), the debt will be transferred back to one of the SAs from which the arrangement was originally created.
- Inserts a characteristic under the PA SA to indicate that it has been broken (we need this for the account debt monitor (ADM) a few steps down).
- Inserts a row on the account debt monitor trigger. This trigger will cause the account to be reviewed by the ADM the next time it runs.

> **Note**
>
> **The PA SA must final bill before it closes.** The PA SA will only close after the PA SA is final billed. This is OK as it won't have any money left on it.

When the ADM next runs, it will analyze the account's reinstated debt. We recommend creating a new override collection criteria for the normal debt class that will return a value of true if the account has a closed payment arrangement that has been broken in the last X days (where X is a parameter of the override collection criteria's algorithm). If this algorithm returns a true, kick off a unique collection process template (that has nasty events). A sample algorithm of this type is supplied in the base package - COLL COND PA.

To complete this discussion, we have to worry about the situation when the final bill of a payment arrangement goes unpaid. In this situation, the payment arrangement is stopped and will therefore not be processed by the ADM. In this case, the write off monitor will process the PA SA after its final bill's due date and a write-off process will start. This write off process will have a single event that calls the Break Payment Arrangement algorithm (described above). After the FT's are issued in this event, the SA will close (because it's been final billed and its balance will go to zero).

The Big Picture Of Pay Plans

A pay plan (PP) is an agreement with a customer to make payments on specific dates. Pay plans differ from payment arrangements in that pay plans have user-defined scheduled payment dates, which are independent from the customer's billing dates. In other words, payment arrangements appear on the customer's bills, pay plan scheduled payments do not.

If a customer is in arrears and you want to receive payments on specific dates (as opposed to with the customer's regular bills), you would set up a pay plan and define the dates on which you expect the payments.

The topics in this section describe how pay plans work.

- A Pay Plan Has One Or More Scheduled Payments
- Automatic Payments Can Be Created On The Scheduled Payment Dates
- A Pay Plan Insulates Overdue Debt From The Account Debt Monitor

Copyright © 2026, Oracle and/or its affiliates.

- A Pay Plan Must Reference A Pay Plan Type
- A Pay Plan May Reference A Third-Party Payor
- The Lifecycle Of A Pay Plan
- Highlighting The Existence Of Broken / Kept / Active and Denied Pay Plans
- A Pay Plan Must Reference A Payment Method
- The Pay Plan Monitor
- How Pay Plans Affect The Account Debt Monitor
- Collection Process / Severance Process Cancellation
- Interesting Pay Plan Facts

A Pay Plan Has One Or More Scheduled Payments

When you create a pay plan for an account, you must define the number of scheduled payments and their respective amounts. There is no limit to the number of scheduled payments that may be set up under a pay plan.

Automatic Payments Can Be Created On The Scheduled Payment Dates

The system will create automatic payments on a pay plan's scheduled payment dates if:

- The account is set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically), and
- The payment method defined on the pay plan indicates automatic payment is being used

The background process called PPAPAY is responsible for creating these automatic payments. It does this by calling the automatic payment creation algorithm plugged in on the installation record.

> **Note**
>
> If the **Autopay Creation Option** on the installation record is set to Create On Extract Date, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to Automatic Payment Dates for more information on how this date is calculated). Refer to Automatic Payments for more information.

A Pay Plan Insulates Overdue Debt From The Account Debt Monitor

A pay plan's scheduled payments are used by the account debt monitor as "pseudo payments" that relieve the account's debt before it is subjected to the collection criteria (refer to How Does The Account Debt Monitor Work for more information about collection criteria).

It's important to understand that a pay plan only insulates the account's debt that belongs to the pay plan's debt class. Therefore, if a customer has debt that belongs to two debt classes

Copyright © 2026, Oracle and/or its affiliates.

(e.g., normal debt and 3 rd party pass through debt), you would need to set up a separate pay plan for each debt class (assuming both types of debt are covered by a pay plan). Refer to Different Collection Criteria For Different Customers and Different Debt for more information about debt classes.

A Pay Plan Must Reference A Pay Plan Type

When you create a pay plan, you must define its pay plan type. The pay plan type controls the following functions:

- The debt class whose debt is insulated by the pay plan.
- The type of algorithm (if any) that is executed when the pay plan is broken. You might use such an algorithm to affect the customer's credit rating when the pay plan is broken.

A Pay Plan May Reference A Third-Party Payor

In addition to referencing the account whose debt is insulated by the pay plan, the pay plan must also reference the account that is responsible for making the payments. We refer to this second account as the pay plan's "payor".

While the payor's account is typically the same as the account whose debt is insulated by the pay plan, you can indicate a third-party payor (e.g., a social service agency) is responsible for making the pay plan's scheduled payments.

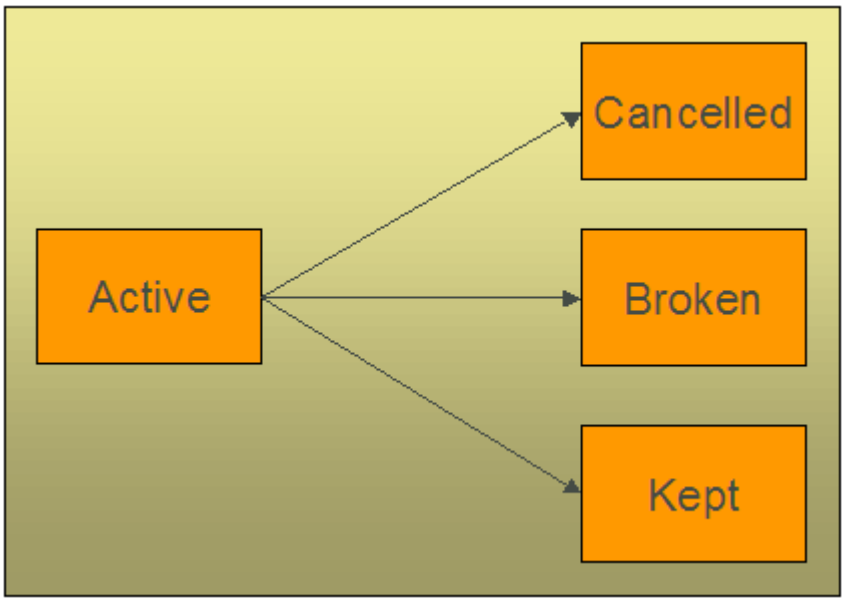
If your organization allows third-party payors, you can define each on the third-party payor control table. This control table exists to simplify the data-entry effort when you create a pay plan (as it defines the account associated with the third-party payor).

> **Note**
>
> If a pay plan does not reference a third-party payor, any non-third-party payor (i.e., any account that is not defined in the third-party payor control table) can make payments on behalf of the customer. If a pay plan references a third-party payor, only payments made by the third party on behalf of the customer are counted towards the fulfillment of the pay plan.

The Lifecycle Of A Pay Plan

The following diagram shows the possible lifecycle of a pay plan:

Copyright © 2026, Oracle and/or its affiliates.



The following points explain this lifecycle:

- Pay plans are initially created in the active state. Active pay plans are monitored for compliance by The Pay Plan Monitor.
- A pay plan may be cancelled as follows:
  - A user can cancel a pay plan at will.
  - When a SA is stopped AND there are no other active SAs in the same debt class, all **Active** pay plans associated with the account and debt class will be will be **Canceled** .
  - The activation of a collection event that calls the "cancel pay plan" algorithm will cancel all active pay plans associated with the collection process's debt class. You may want to use such a collection event if your organization cancels active pay plans when new debt causes a collection process to kick-off. Note, the base package algorithm that performs this function will not cancel the pay plan if it's associated with a 3rd party payor.
- The Pay Plan Monitor causes active pay plans to become broken if sufficient payments have not been made to satisfy the pay plan's scheduled payments.
- The Pay Plan Monitor causes active pay plans to become kept when it detects that sufficient payments have been made to satisfy the pay plan's scheduled payments.

Highlighting The Existence Of Broken / Kept / Active and Denied Pay Plans

You can define on the installation record plug-in algorithms that format alert messages. (Refer to Installation Options - Algorithms for additional information.) We recommend that you take advantage of the following algorithms to highlight pay plans:

- Highlight pay plans in a given status. This algorithm is used to highlight pay plans in a given state (broken, kept, cancelled) that were started within the last X days.
- Highlight customer contacts of a given type. This algorithm would be used to highlight customer contacts of a given type that were created within the last X days. This would be useful if you create a specific type of customer contact when you deny a pay plan. Some utilities do this to prevent customers from shopping around.

Copyright © 2026, Oracle and/or its affiliates.

In addition, you can define account-specific alerts to highlight customers that should never be allowed to have a pay plan (for whatever reason).

A Pay Plan Must Reference A Payment Method

When you create a pay plan, you must define how the customer will make the payments be referencing a payment method. Examples of payment methods include: In Person , Wire Transfer, By Post, Express Mail, etc.

The payment method is more than just documentation as it defines the number of grace days the customer has to make the pay plan's scheduled payments. For example, if you set up the payment method control table to indicate that payments made By Post have 3 grace days, then the customer has up to 3 days after each scheduled payment date to make the payment. If payment is not received by the scheduled payment date plus the grace days, the pay plan will be marked as broken (and the ADM will be triggered).

The Pay Plan Monitor

NOT_SUPPORTED

Please understand the concepts described in The Lifecycle Of A Pay Plan and The Tendering Account May Differ From The Account Whose Debt Is Relieved before reading this section.

The Pay Plan Monitor background process (referred to as PPM) is responsible for monitoring active payment plans. This process can cause a pay plan (PP) to become kept or broken (or being left as active).

Note

When is a pay plan marked as broken / kept? It's important to understand that only the PPM can cause a pay plan to become kept or broken. This means that if a customer makes a payment that satisfies a pay plan, the pay plan will only be marked as kept when the pay plan monitor next runs. Analogously, if a payment is cancelled, nothing will happen to an active pay plan until the PPM next runs. When the PPM next runs, it will see that the scheduled payment was not kept and it will break the pay plan and schedule the ADM to be executed. When the ADM next executes, it will create a collection process (because the customer's debt will no longer be insulated by the pay plan's scheduled payments).

NSF Cancellations After A Pay Plan Is Kept. If a payment is cancelled due to non-sufficient funds (NSF) after a pay plan is marked as kept, the pay plan will remain kept. But keep in mind that the pay plan's account is scheduled for review by the ADM when a payment is cancelled due to NSF. When the ADM reviews the account's debt, it will no longer have an active pay plan to insulate it and the account's debt will likely trigger a new collection process. Refer to How Pay Plans Affect The Account Debt Monitor for more information.

Copyright © 2026, Oracle and/or its affiliates.

The following points describe, at a high level, how the PPM monitors a pay plan (PP) for compliance.

- The system selects all frozen, non-cancelled payment segments associated with the PP's account and debt class where:
  - The payment date is after the start date of the pay plan, and
  - The payment's pay event has at least one tender that references the pay plan's payor.
- The system logically reduces / removes past and current scheduled payments (starting with the earliest scheduled payment) until the total amount of payment segments is exhausted (or there are no more historical / current scheduled payments).

> **Note**
>
> **Paying pay plans in advance.** Scheduled payments with a future date are not logically removed / reduced. This means that if a customer makes advance payments on a pay plan, it will not be marked as kept until all scheduled payment dates are in the past.

- If all scheduled payments have been logically removed, the pay plan is marked as kept.
- If there exist scheduled payments where the pay date + grace days (grace days are defined on the pay plan's payment method) is before the current date (i.e., a payment doesn't exist for a scheduled payment):
  - The pay plan is marked as broken.
  - The PP's break algorithm (if any) is called (note, for European / Australian pay plans, there are scenarios where the break algorithm can cause the pay plan to become unbroken - when there aren't at least two missed, historical scheduled payments).
  - An ADM trigger is stored for the PP's account. This will cause the account to be reviewed by the ADM the next time it runs. And because the pay plan is broken, its scheduled payments will no longer insulate the account's arrearage.

> **Important**
>
> It's important that you schedule the PPM to run before the ADM so that it can break unpaid payment plans prior to the ADM subjecting the account's debt to collection criteria. Refer to How Pay Plans Affect The ADM for more information.

How Pay Plans Affect The Account Debt Monitor

As described under A Pay Plan Insulates Overdue Debt From The Account Debt Monitor, a pay plan's scheduled payments insulate an account's debt from the ADM. This section describes how this is accomplished.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> You should understand the concepts in How Does The Account Debt Monitor Work and The Tendering Account May Differ From The Account Whose Debt Is Relieved before reading the following.

> **Note**
>
> **The ADM will be triggered when a pay plan is broken.** Refer to The Pay Plan Monitor for an explanation of how the ADM is triggered when a pay plan is broken.

Before the ADM subjects an account's debt to the collection criteria, it calls the debt's debt class's Override Arrears Algorithm (this is an optional plug-in spot on Debt Class). This algorithm is passed the debt class's aged debt and manipulates it as follows:

- First, a list of all past, present and future scheduled payments associated with the account and debt class's active pay plans is constructed.
  - If multiple payors are encountered (because the customer has multiple pay plans and these have different payors), a separate list of scheduled payments is maintained for each payor.
- Next, for each payor, retrieve the total amount of frozen, non-cancelled payment segments made on behalf of the pay plan's account and debt class.
  - Select all frozen, non-cancelled payment segments associated with the pay plan's account and debt class whose pay date is >= pay plan's start date and the pay segment's event has at least one tendering account associated with the pay plan's payor.
- Next, logically reduce / remove past and current scheduled payments (starting with the earliest scheduled payment) until the payor's payment amount is exhausted (or there are no more historical / current scheduled payments). Future scheduled payments cannot be remove / reduced.
- Finally, reduce the passed in aged debt with any unpaid scheduled payments.

> **Note**
>
> **This logic is not "hard coded".** Rather, the mechanism used to use a pay plan's scheduled payments to reduce debt is defined in an algorithm defined on the pay plan's debt class. The contents in this section describe how a base package algorithm works. Because it's an algorithm, you can introduce whatever logic you please.

The following is an example of how pay plans affect aged debt.

Date	Event	SA's Arrears	SA's Balances	Scheduled Payments

Copyright © 2026, Oracle and/or its affiliates.

Prior to creation of the PP 1/18/2000		$1,000 - 90 days old $1,600 - 60 days old $1,900 - 30 days old	Current: $4,500 Payoff: $4,500	
1/18/2000	Pay Plan created. The $4,500 in future scheduled payments offsets the existing $4,500 of aged debt.	$1,000 - 90 days old $1,600 - 60 days old $1,900 - 30 days old **De facto ADM debt: $0**	Current: $4,500 Payoff: $4,500	1/20/2001 $1,500 2/01/2001 $1,500 2/07/2001 $1,500
1/20/2001	The customer pays $1,500. There exists $3,000 of future scheduled payments that offset the arrears	$1,100 - 62 days old $1,900 - 32 days old **De facto ADM debt: $0**	Current: $3,000 Payoff: $3,000	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
1/20/2001	ADM runs. The $3000 in future scheduled payments offsets the Current Balance of $3000, so CC events not created.	$1,100 - 62 days old $1,900 - 32 days old **De facto ADM debt: $0**	Current: $3,000 Payoff: $3,000	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
1/24/2001	A new bill is created for $400	$1,100 - 62 days old $1,900 - 32 days old $400 - 1 day old **De facto ADM debt: $400 - 1 day old**	Current: $3,400 Payoff: $3,400	1/20/2001 $1,500 "Paid" 2/01/2001 $1,500 Future 2/07/2001 $1,500 Future
2/2/2001	Pay Plan Monitor runs. PP marked as Broken because the 2/1/2001 scheduled payment has not been paid (assuming no grace period on the pay plan's payment method)	$1,100 - 74 days old $1,900 - 44 days old $400 - 8 days old	Current: $3,400 Payoff: $3,400	1/20/2001 $1,500 "Paid" **2/01/2001 $1,500 "Late"** 2/07/2001 $1,500 Future

Copyright © 2026, Oracle and/or its affiliates.

2/2/2001	ADM runs. There are no active pay plans and therefore there is nothing to insulate the customer's debt. Therefore the aged debt will be subjected to the collection criteria and an appropriate collection process will be created.	$1,100 - 74 days old $1,900 - 44 days old $400 - 8 days old **De facto ADM debt is the same as above (i.e., rather old)**	Current: $3,400 Payoff: $3,400	**Pay plan is broken and therefore its scheduled payments cannot be used.**

Collection Process / Severance Process Cancellation

When a pay plan (PP) is created, the system determines if it can cancel active collection and severance processes associated with the pay plan's account and debt class. It does this because a pay plan's scheduled payments act as "pseudo payments" that relieve the account's debt (temporarily). The following points describe how this works:

- The system attempts to cancel collection processes by calling the Collection Process Cancel Criteria Algorithm defined on the debt class that is associated with the collection process. This algorithm is meant to cancel a collection process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. Because of the existence of the pay plan, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount (see How Pay Plans Affect The Account Debt Monitor for more information about how debt is reduced). Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.
- It attempts to cancel severance processes by calling the Severance Process Cancel Criteria Algorithm defined on the debt class that is associated with the severance process. This algorithm is meant to cancel a severance process if the sum of ALL service agreements in the debt class have debt less than a given threshold amount. Because of the existence of the pay plan, the actual debt will be temporarily reduced by the amount of the pay plan's scheduled payments before it is compared to the threshold amount (see How Pay Plans Affect The Account Debt Monitor for more information about how debt is reduced). Note: this temporary reduction will only occur if you have plugged in the appropriate pay plan debt reduction algorithm on the debt class.

If collection / severance processes still exist for the account / debt class associated with the pay plan, a warning is issued.

Interesting Pay Plan Facts

The following points describe a variety of interesting facts about pay plans (PP):

- An account may have many active pay plans. However, only 1 pay plan may be active for a given account / debt class / payor at any point in time.
- The existence of a pay plan has no impact on payment distribution.

Copyright © 2026, Oracle and/or its affiliates.

- When a SA is stopped, if the SA's debt class has an active PP AND there are no other active SAs in the same debt class, the PP will be cancelled
  - The cancel reason will be "cancelled by system" (as opposed to "cancelled by user")
- If necessary, different collection / severance processes can be triggered if a broken PP is detected (via the override algorithms on CCC and write-off control - we do NOT provide such algorithms).

Setting Up Pay Plan Control Tables

This section describes the control tables needed to set up pay plans.

- Setting Up Pay Plan Types
- Setting Up Payment Methods
- Setting Up Third Party Payors

Setting Up Pay Plan Types

Pay plan types control what is done by a given pay plan. Open **Admin**, and then **Credit & Collection**, and then **Pay Plan Type** to define your pay plan types.

> **NOT_SUPPORTED**
>
> For more information refer to The Big Picture Of Pay Plans for more information.

Description of Page

To modify a pay plan type, simply move to a field and change its value. To add a new pay plan type, press + to insert a row, then fill in the information for each field. The following fields display:

Pay Plan Type The name of the pay plan type.

Description A meaningful description of the pay plan type.

Broken Algorithm This algorithm is called when a pay plan is broken. Refer to The Pay Plan Monitor for more information about how pay plans are broken.

If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that breaks a pay plan.

Debt Class The debt class covered by pay plans of this type. Refer to Setting Up Debt Classes for more information.

Setting Up Payment Methods

Payment methods are used to describe how a customer intends to make their pay plan's scheduled payments. Open **Admin**, and then **Credit & Collection**, and then **Pay Method** to define your payment methods.

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> For more information refer to A Pay Plan Must Reference A Payment Method for more information.

Description of Page

To modify a pay method, simply move to a field and change its value. To add a new pay method, press + to insert a row, then fill in the information for each field. The following fields display:

Pay Method The name of the payment method.

Description A meaningful description of the payment method.

Grace Days The number of days added to the scheduled payment date. The ADM will consider the pay plan to be broken if payment is not made by the scheduled date plus the grace days.

Auto Pay If the pay method is marked as being for **Auto Pay**, the PPAPAY background process will automatically create an automatic payment on the pay plan's scheduled payment dates IF the account has been set up for automatic payment.

Setting Up Third Party Payors

Pay plans support optional third-party payors. Open **Admin**, and then **Credit & Collection**, and then **Third Party Payor** to define your third-party payors.

> **Note**
>
> A third-party payor refers to an account. You must set up the account before you can create a third-party payor.

> **NOT_SUPPORTED**
>
> Refer to A Pay Plan May Reference A 3rd Party Payor for more information.

Description of Page

The following fields display for each third party payor:

Third Party Payor Provide a meaning id for the third-party payor that can be easily recognized when setting up a pay plan.

Description A meaningful description of the payor.

Account ID The account that is used for this payor. It is this account that is tracked as the "payee" of any payments made towards a third party payor's pay plan.

Active Check this box if the payor is currently available to participate in pay plans.

Copyright © 2026, Oracle and/or its affiliates.

Creating Collection, Severance & Write-Off Procedures

Your collection procedures define how your organization collects overdue debt. Your severance procedures define how your organization severs service agreements when collection attempts fail. Your write-off procedures define how your organization writes off finaled debt. In this section, we describe how to set up the data that controls these procedures.

NOT_SUPPORTED

For more information about collection, severance and write-off procedures, see The Big Picture Of Credit & Collections (C&C).

Warning

There are innumerable ways to design your collection, severance and write-off procedures. Some designs will result in easy long-term maintenance, others will result in maintenance headaches. In this section, we provide information to help you understand the ramifications of the various options. Before you set up your production collection, severance and write-off procedures, we encourage you to gain an intuitive understanding of these options by using the system to prototype the alternatives.

- Designing Your Collection Procedures
- Setting Up Collection Procedures
- Designing Your Severance Procedures
- Designing Your Reconnection Procedures
- Setting Up Severance Procedures
- Designing Your Write-Off Procedures
- Setting Up Write-Off Procedures
- Setting Up Feature Configuration

Designing Your Collection Procedures

The design of your collection procedures is an iterative process. Over time, you will develop intuitive skills that will allow you to skip some iterations. However, when you're starting out, we recommend you use the following matrix as your guide. When the matrix is complete, you're ready to set up the collection process control tables.

SA's Debt Class	Account's Collection Class:	Account's Collection Class:

The topics discussed below will gradually complete this matrix using a simple case study.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about how the information in this matrix is used to monitor your customers' debt, refer to Different Collection Criteria For Different Customers And Different Debt.

- Designing Your Debt Classes
- Designing Your Collection Classes
- Designing Collection Class Controls
- Designing Your Collection Class Control Overrides
- Designing Collection Process Templates & Collection Event Types
- Defining Cancellation Process Auto Cancellation Criteria

Designing Your Debt Classes

Multiple debt classes are needed when you have different collection procedures for different types of service agreements. If all service agreement debt is collected the same way, then you'll have just one debt class (call it Generic). However, if you're like many organizations, you will have multiple debt classes. The following points will help you understand why:

- If you have both regulated and unregulated service, you probably have different debt classes for each type of service. Why? Because your local regulators control how you collect and cutoff overdue, regulated debt. For unregulated debt, your organization controls how overdue debt is collected.
- If your customers make charitable donations, you will have a charitable contribution debt class. Why? Because you probably send a different type of letter when the customer falls into arrears on their charitable contributions. You also can't cut off their water service if they don't make their charitable contributions.
- If you levy deposits, you will probably have a deposit debt class. Why? Because you probably respond differently if the customer doesn't pay their deposit (e.g., you may decide to cut off their electric service until the deposit is paid).
- If you allow customers to make payments on Non-Billed Budgets, you will probably have a Non-Billed Budget debt class. Why? Because you probably respond differently if the customer doesn't pay their Non-Billed Budget (e.g., you may decide to expire the Non-Billed Budget but not affect their other service since the Non-Billed Budget is a way for customers to prepay for upcoming bills).
- If you write-off uncollectable debt, you will need another debt class for write-off service agreements. Why? Because when you write-off debt in the system, you transfer the uncollectable debt from the original service agreement(s) to a write-off service agreement. The write-off service agreement holds this debt forever (or until it is paid). You need to use a different debt class for the write-off service agreements because they aren't subject to collection criteria.
- If you use the system to charge your organization's company usage, you'll need another debt class (we refer to it as the "N/A" debt class below). Why? Because all service agreements must have a debt class, even those that will never have debt.

SA's Debt Class	Account's Collection Class	Account's Collection Class

Copyright © 2026, Oracle and/or its affiliates.

Regulated
Unregulated
Charitable Contribution
Deposit
Non-Billed Budget
Write Off
N/A

Designing Your Collection Classes

Multiple collection classes are needed when any debt class has different collection rules depending on the type of customer. If all customers within all debt classes are collected the same way, then you'll just have a single collection class (call it Generic). However, if you're like many organizations, you will have multiple collection classes.

Consider unregulated debt. For commercial/industrial customers, you probably don't worry until they owe you more than, say, $100 after 20 days. For residential customers, you probably don't worry until they owe you more than, say, $5 after 20 days. In this situation, you will have at least two collection classes: one for large customers, the other for residential customers.

In our example, we are assuming you have two collection classes: Residential and Commercial/Industrial.

SA's Debt Class	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
Charitable Contribution		
Regulated		
Unregulated		
Write Off		
Company Usage		

Designing Collection Class Controls

At this point we have the rows and columns defined in our matrix. Now it's time to work on the individual cells.

Each cell should have a "collection class control" that defines its collection criteria and what to do if the criteria are violated. If a cell doesn't have a collection class control, this means you don't have any debt associated with that combination of collection class and debt class. So, we'll mark each cell without debt with "N/A".

SA's Debt Class	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
Regulated		N/A
Unregulated		
Charitable Contribution		N/A

Copyright © 2026, Oracle and/or its affiliates.

Deposit		
Write Off		
N/A		

Next, we'll mark each cell for debt classes whose debt isn't collectable (i.e., the write-off and N/A debt classes).

SA's Debt Class	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
Regulated		N/A
Unregulated		
Charitable Contribution		N/A
Deposit		
Write Off	N/A	N/A
N/A	N/A	N/A

> **Note**
>
> If the Account Debt Monitor encounters debt associated with a non-existent collection class control, it will issue an error.

Determining the collection criteria in each remaining cell can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For charitable debt, if the customer is more than $0 in arrears by more than 20 days, kick off the "charity reminder" collection process. We'll talk more about this collection process later.
- For regulated / residential debt, if the customer is more than $15 in arrears by more than 20 days, kick off the Normal Regulated collection process. We'll talk more about this collection process later.
- For unregulated / residential debt, if the customer is more than $5 in arrears by more than 20 days, kick off the Normal Unregulated collection process. We'll talk more about this collection process later.
- For unregulated / commercial-industrial debt we have multiple criteria:
  - Highest priority. If the customer is more than $10,000 in arrears by more than 20 days, kick off the Large Overdue Debt collection process. We'll talk more about this collection process later.
  - Lower priority. If the customer is more than $100 in arrears by more than 20 days, kick off the Normal Unregulated collection process. We'll talk more about this collection process later.
- For deposit debt (regardless of collection class) we have multiple criteria:
  - Highest priority. If the customer is more than $5 in arrears by more than 50 days, kick off the Deposit Severely Overdue collection process. We'll talk more about this collection process later.

Copyright © 2026, Oracle and/or its affiliates.

- Lower priority. If the customer is more than $15 in arrears by more than 20 days, kick off the Deposit collection process. We'll talk more about this collection process later.

Given the above, our matrix will look as follows:

SA's Debt Class	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
Charitable Contribution	If > $0 is older than 20 days, start Charity Reminder collection process.	N/A
Regulated	If > $15 is older than 20 days, start Normal Regulated collection process.	N/A
Unregulated	If > $5 is older than 20 days, start Normal Unregulated collection process.	Highest priority: If > $10,000 is older than 20 days, start Large Overdue Debt collection process. Lower priority: If > $100 is older than 20 days, start Normal Unregulated collection process.
Deposit	Highest priority: If > $5 is older than 50 days, start Deposit Severely Overdue collection process. Lower priority: If > $15 is older than 20 days, start Deposit collection process.	Highest priority: If > $5 is older than 50 days, start Deposit Severely Overdue collection process. Lower priority: If > $15 is older than 20 days, start Deposit collection process.
Write Off	N/A	N/A
N/A	N/A	N/A

Designing Your Collection Class Control Overrides

> **Warning**
>
> Your collection needs may not require any overrides for your collection class control matrix and therefore this section may not be relevant.

The following matrix will help you design your collection class overrides. When the matrix is complete, you're ready to set up the collection class control tables.

Notice that the matrix has two dimensions: one is dependent on collection condition algorithms; the other is dependent on the collection class controls designed in the previous section. Collection condition algorithms are confusing. Think of them as optional conditions that, if met, will subject the collection class control's debt to different collection criteria.

Each cell in the matrix contains the collection criteria that will be applied to the account's debt when the collection condition is met (i.e., the same type of criteria - dollars and days and collection process - are defined in each cell).

Copyright © 2026, Oracle and/or its affiliates.

We label the first collection condition as the Default. The collection criteria associated with this column will be used to analyze an account's debt when none of the other conditions applies. We'll start by indicating the Default collection criteria (this was defined in the previous section).

SA's Debt Class	Account's Collection Class: Default	Account's Collection Class: Credit Rating < Threshold
Residential Charitable Contribution	See default collection criteria defined in previous section.	
Residential Regulated	See default collection criteria defined in previous section.	
Residential Unregulated	See default collection criteria defined in previous section.	
Commercial-Industrial Unregulated	See default collection criteria defined in previous section.	
Residential Deposit	See default collection criteria defined in previous section.	
Commercial-Industrial Deposit	See default collection criteria defined in previous section.	

If a different collection process OR criteria should be used when other conditions are met, you should indicate such by defining the collection criteria in the cell. For example, if we assume that all unregulated residential debt has a different collection process when the account's credit score is less than the threshold credit rating on the installation record or the account’s CIS Division, our matrix will look as follows:

SA's Debt Class	Account's Collection Class: Default	Account's Collection Class: Credit Rating < Threshold
Residential Charitable Contribution	See default collection criteria defined in previous section.	
Residential Regulated	See default collection criteria defined in previous section.	
Residential Unregulated	See default collection criteria defined in previous section.	Override: If Credit Rating is lower than the credit rating threshold: If > $5 is older than 15 days, start Risky Unregulated collection process.
Commercial-Industrial Unregulated	See default collection criteria defined in previous section.	
Residential Deposit	See default collection criteria defined in previous section.	
Commercial-Industrial Deposit	See default collection criteria defined in previous section.	

Once the matrix is complete, you're ready to design your collection process and collection event types.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The collection conditions are limited by your imagination (and business requirements).** We have provided the collection conditions you see above as an example; we don't expect you'll be able to use the exact conditions we supply. Your conditions will be based on any number of factors. For example, if you have different collection criteria that apply during winter months, you should add a new collection condition (called Winter Season). Or if you have different criteria based on years of service, you could have another condition.
>
> **New collection conditions may require programming.** See How To Add A New Algorithm for more information.

Designing Collection Process Templates & Collection Event Types

The following table shows the collection process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Collection Process Template	Collection Event Template	Triggered X Days From Start Of Collection Process
Charitable Contribution Reminder	Charity courtesy reminder letter	0
	Start severance	15
Normal Regulated	Regulated courtesy reminder letter	0
	Regulated second notice letter	10
	Start severance	15
Large Overdue Debt	Large debt courtesy reminder letter	0
	To Do for large overdue debt	3
	Large debt second notice letter	10
	Start severance	15
Normal Unregulated	Unregulated courtesy reminder letter	0
	Unregulated second notice letter	5
	Start severance	10
Risky Customer Unregulated	Unregulated risky letter	0
	Start severance	5
Deposit	Deposit reminder	0
Deposit Severely Overdue	Create To Do entry	0

If we extract each unique event type from the above table, we end up with the following:

Collection Event Type	Event Type

Copyright © 2026, Oracle and/or its affiliates.

Charity courtesy reminder letter	Send Letter - CHARIT REMIN
Start severance	Start Severance Process
Regulated courtesy reminder letter	Send Letter - REGUL REMIN
Regulated second notice letter	Send Letter - REGUL second
Large debt courtesy reminder	Send Letter - LARGE REMIN
Risky debt courtesy reminder	Send Letter - RISKY REMIN
To Do for large overdue debt	Issue To Do
Large debt second notice letter	Send Letter - LARGE second
Unregulated courtesy reminder letter	Send Letter - UNREG REMIN
Unregulated second notice letter	Send Letter - UNREG second
Deposit reminder	Send Letter - DEPOS REMIN
To Do for deposit severely overdue	Issue To Do

Now you're (almost) ready to set up your collection procedures.

Defining Cancellation Process Auto Cancellation Criteria

The topics in the section How Are Collection Processes Cancelled describe the two algorithms that play a part in the cancellation of a collection process. It also describes when to use what type of algorithm. Please read this section and then set up the appropriate cancellation criteria on your Debt Classes, and optionally, on your Collection Process Templates.

Setting Up Collection Procedures

In the previous section, Designing Your Collection Procedures, we presented a case study that illustrated a mythical organization's collection procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

- Setting Up Collection Event Types
- Setting Up Collection Process Templates
- Setting Up Collection Classes
- Setting Up Debt Classes
- Setting Up Collection Class Controls

Setting Up Collection Event Types

Collection event types control what is done by a given collection event. Open **Admin**, and then **Credit & Collection**, and then **Collection Event Type**, and then **Add** to define your collection event types.

Description of Page

Enter a unique **Collection Event Type** and **Description** for the collection event type.

Copyright © 2026, Oracle and/or its affiliates.

Enter the **Collection Event Type**. Permissible values are: Affect Credit, Rating/Cash-Only, Cancel Budget, Generic Algorithm, Send Letter, Create To Do Entry, Start Severance Process. The following discussion describes the parameters that must be defined for each type of collection event.

The Affect Credit Rating/Cash-Only collection event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- Use **Credit Rating Points** to define this event's affect on the account's credit rating. This should be a negative number. An account's credit rating is equal to the beginning credit rating amount defined on the installation record or the account's CIS Division plus the sum of credit rating demerits that are currently in effect. When an account's credit rating is less than the credit rating threshold defined on the installation record or the account's CIS Division, the account's credit rating is displayed as an alert on Control Central.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record or the account's CIS Division, the account is flagged as cash-only during payment processing and on Control Central.
- Use **Credit Rating Months** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

> **NOT_SUPPORTED**
>
> For more information, refer to Account - Credit Rating.

The Send Letter collection event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following **Parameters** for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

> **Note**
>
> **Letter creation is triggered via a customer contact.** You must set up a customer contact type for each type of letter you generate. You specify the necessary customer contact type on the collection event. Refer to Setting Up Letter Templates for more information.

The Cancel Budget collection event type cancel an account's budget plan (if the account is on such a plan). When a budget plan is cancelled, adjustments are issued to synchronize every service agreement's current balance with its payoff balance and each applicable SA's recurring charge amount (i.e., budget amount) is set to zero.

The Generic Algorithm collection event type causes the algorithm defined in the **Collection Event Alg** to be executed. You use this type of algorithm when the standard types of collection events won't do what you need done. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).

Copyright © 2026, Oracle and/or its affiliates.

- On this algorithm, reference an Algorithm Type that determines the algorithm that will be called when events of this type of activated.

The Create To Do Entry collection event type causes a To Do entry to be issued. A good example of where this is used is when the collection event requires that the customer be called on the phone. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-CEVT for the type of To Do entry that's created).

The Start Severance Process type will start a severance process for every service agreement linked to the collection process. No parameters are needed for this type of event.

Enter a **Long Description** to fully describe the collection event type.

Setting Up Collection Process Templates

Collection process templates define the collection events that will be executed when a collection criteria rule is violated. Open **Admin**, and then **Credit & Collection**, and then **Collection Process Template**, and then **Add** to define your collection process templates.

Description of Page

Enter a unique **Collection Process Template** and **Description** for the collection process template.

Select a **Cancel Criteria Algorithm** if your organization allows individual service agreements to be "removed" from a collection process regardless of the debt associated with all service agreements in the debt class. In other words, if your cancel criteria are based on the debt associated with ALL service agreements in a debt class DO NOT SPECIFY THIS ALGORITHM. If this algorithm is specified, it is executed by the collection process monitor when it detects that a credit has been applied to a service agreement linked to an active collection process. This algorithm will indicate if the specific service agreement that has been credited no longer has debt that warrants a collection process. Refer to How Are Collection Processes Cancelled for more information. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that "removes" a service agreement from a collection processes if the service agreement's debt so warrants.

The **Response** grid contains an entry for every collection event that will be created when a collection process that references this template is created. The following information must be defined for each event:

Event Sequence controls the order in which the collection event types appear under the collection process template. The sequence number is system-assigned and cannot be changed. If you have to insert a collection event type between two existing templates, you'll have to remove the latter events, insert the new event, and then re-specify the removed events.

Collection Event Type Specify the type of collection event to be generated.

Days After Process Creation Specify the number of days after the creation of the collection process that the related collection event will be triggered. Refer to Calendar vs Work Days for a description of how this system uses this information to set the trigger date on the respective collection events.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about collection event types, see Setting Up Collection Event Types. For more information about trigger dates, see Collection Event Trigger Date.

Setting Up Collection Classes

Every account has a collection class. This class is one of several fields that control the collection method applied to the account's debt. Open **Admin**, and then **Credit & Collection**, and then **Collection Class** to define your collection classes.

NOT_SUPPORTED

For more information about collection classes, see Designing Your Collection Classes.

Description of Page

Enter a unique **Collection Class** code and **Description** for each collection class.

Indicate which method is used to monitor the member accounts' unpaid debt:

- If you practice balance-forward accounting for accounts belonging to this collection class, select Collection, Severance & Write-Off. This method of collection is described throughout this chapter.
- If you practice open-item accounting for accounts belonging to this collection class, select Overdue. This method of collection is described under Defining Overdue Processing Options.
- If accounts belonging to this collection class are not subject to either of the above collection methods, select Not Eligible For Collection. Please be aware that these accounts will NOT be reviewed for overdue debt.

Setting Up Debt Classes

Every SA type has a debt class. This class is one of several fields that control the collection criteria applied to the service agreement's debt. Open **Admin**, and then **Credit & Collection**, and then **Debt Class**, and then **Add** to define your debt classes.

NOT_SUPPORTED

For more information about debt classes, see Designing Your Debt Classes.

Description of Page

Enter a unique **Debt Class** and **Description** for the debt class.

Copyright © 2026, Oracle and/or its affiliates.

Turn on **Eligible for Collection** if service agreements belonging to this debt class have their debt monitored by the collection process. This should only be turned off if this debt cannot be collected, e.g., write-off debt.

The grid that follows contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to an arbitrary number as only one algorithm for each system event is allowed in this case.

> **Warning**
>
> These algorithms are typically significant system processes. The absence of an algorithm may prevent the system from operating correctly.

You can define algorithms for the following **System Events**:

System Event	Optional / Required	Description
Collection Process Cancellation	Required if debt class is eligible for collection	This algorithm determines if a collection process can be canceled, and if so, it cancels it. Refer to How Are Collection Processes Cancelled for more information.
Severance Process Cancellation	Required if debt class is eligible for collection	This algorithm determines if a severance process can be canceled, and if so, it cancels it. Refer to How Are Severance Processes Cancelled for more information.

Copyright © 2026, Oracle and/or its affiliates.

Override Arrears Due to Pay Plan	Required if you use Pay Plans	This algorithm is called to temporarily override a customer's arrears using a pay plan's scheduled payments when the system looks at an account's debt from a credit & collections perspective (i.e., the ADM calls this algorithm before it subjects a customer's debt to the collection criteria and when a pay plan is created). It does not actually change any data, but overlays the current arrears with the pay plan scheduled payments. This algorithm is also called by the above algorithms when a pay plan is created in order to evaluate if the scheduled payments actually cover the arrears (if so, the collection / severance processes are cancelled). It is also called periodically by the ADM in order to establish if the current state of the pay plan still covers the arrears. Refer to How Pay Plans Affect The Account Debt Monitor.

Setting Up Collection Class Controls

The topics in this section describe the windows on which you set up your collection class control information.

> **Warning**
>
> The information in this page is what controls how the system analyzes your customer's debt. The flexibility of this control information provides you with almost unlimited options. This is very powerful, but it requires careful analysis. Refer to Designing Your Collection Procedures **for more information**.

- Collection Class Control - Main Information
- Collection Class Control - Debt Criteria

Collection Class Control - Main Information

The information on this transaction defines the conditions that will be checked by the Account Debt Monitor when it checks if an account has violated your debt criteria.

Copyright © 2026, Oracle and/or its affiliates.

Open **Admin**, and then **Credit & Collection**, and then **Collection Class Control**, and then **Add** to define this information.

> **NOT_SUPPORTED**
>
> For more information about collection class control, refer to Designing Collection Class Controls.

Description of Page

Enter a unique **Collection Class Control** code and **Description** for the collection class control (CCC).

Enter the **CIS Division** to which the CCC's criteria applies.

Enter the **Collection Class** to which the CCC's criteria applies.

Enter the **Debt Class** to which the CCC's criteria applies.

Enter the **Currency Code** in which the CCC's criteria are denominated.

Use **Long Description** to further describe the CCC.

> **NOT_SUPPORTED**
>
> The information in the following grid is not intuitively obvious. Refer to Designing Collection Class Controls and Designing Your Collection Class Control Overrides for more information.

The grid which follows contains the conditions that are checked by the Account Debt Monitor (ADM) to determine the type of criteria (defined on the next tab) that will be applied against an account's debt. In other words - the ADM will check each condition (from highest to lowest **Priority**). The first condition that returns a value of true will cause the system to compare the account's debt against the debt criteria defined on the next tab.

Multiple conditions may be defined if different conditions result in a different type of debt thresholds (or a different type of collection process). The following fields are required for each condition:

Collection Condition Priority The priority controls the order in which the ADM checks if a collection condition applies (the lower the number, the higher the priority). Higher priorities are checked before lower priorities.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is COLL_CAT_PRIO_FLG.

Condition Algorithm Define the algorithm used to check if an account should be subject to the collection criteria defined on the next tab. If the algorithm returns a value of true (i.e., the condition is met), the ADM will compare the account's debt against the **Debt Criteria** (defined

Copyright © 2026, Oracle and/or its affiliates.

on the next tab) and start a collection process if the account has debt that violates these criteria.

You must have at least one collection condition; otherwise the system will not have criteria against which to compare the account's debt. This entry should have the lowest priority code and reference the "default" algorithm. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that references the "default" collection condition algorithm type (COLL COND DF).

If you have other conditions that should be checked before the default condition, you must create an entry for each in this grid. Each entry should have a priority consistent with your business requirements (and this priority should be higher than the default condition's priority). In addition, you should reference an algorithm that contains the conditions that will be checked to determine if the account should be subject to the debt criteria (defined on the next tab). The system is supplied with many additional algorithm types. In order to take advantage of them, you will need to create an algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that references one of the collection condition algorithm types.

Where Used

Collection class controls contain the data that controls the Account Debt Monitor. Refer to How Does The Account Debt Monitor Work? for more information.

Collection Class Control - Debt Criteria

The information on this page defines the debt and age thresholds used by the Account Debt Monitor when it checks if an account has violated your acceptable levels of debt. Open **Admin**, and then **Credit & Collection**, and then **Collection Class Control**, and then **Search** and navigate to the **Debt Criteria** page to define this information.

> **Note**
>
> **The information on this page is not intuitively obvious.** Refer to Designing Collection Class Controls and Designing Your Collection Class Control Overrides for more information.

Description of Page

The **Debt Criteria** scroll contains an entry for each collection criteria algorithm defined on the **Main** tab. The following information appears

The **Collection Condition Priority** controls the order in which the Account Debt Monitor (ADM) checks if a collection condition applies. Higher priorities are checked before lower priorities.

The **Condition Algorithm** is called by the ADM to determine which collection criteria should be applied to the account's debt. If this algorithm returns a value of true (i.e., the condition is met), the ADM will compare the account's debt against the **Debt Criteria** defined below. If the account violates any criteria, a collection process will be started (using the respective **Collection Process Template**).

The grid that follows contains the debt age and amount of debt that must be violated by the account in order for the ADM to create a collection process template. The following fields should be defined:

Copyright © 2026, Oracle and/or its affiliates.

Arrears Priority controls the order in which the arrears criteria will be checked by the Account Debt Monitor (the lower the number, the higher the priority). The first criteria, if any, that is met will cause a collection process to be created (using the **Collection Process Template**).

Collection Process Template If the Account Debt Monitor determines that the account's debt violates the corresponding criteria, it creates a collection process using the specified collection process template.

Arrears Amount When the Account Debt Monitor checks an account's debt, it determines if the account has debt older than "> Number of Days" (the next field) AND the debt exceeds "> Arrears Amount". If so, a collection process is started.

Days When the Account Debt Monitor checks an account's debt, it determines if the account has debt older than **Days** AND the debt exceeds **Arrears Amount**. If so, a collection process is started.

Where Used

Collection class controls contains the data that controls the Account Debt Monitor.

Designing Your Severance Procedures

The following matrix will help you design your severance procedures. When the matrix is complete, you're ready to set up the severance process control tables.

Notice that the matrix has two dimensions: one is dependent on severance criteria algorithms; the other is dependent on the SA type of the service agreement being severed. The number and type of SA types is dependent on how your organization sets up the SA type table (the SA types shown below are characteristic of those used by a simple utility).

SA Type	Severance Criteria Algorithm: Default	Severance Criteria Algorithm: Customer On Life Support
Electric Residential		
Electric Commercial		
Gas Residential		
Gas Commercial		
Charitable Contribution		

Once you know the values of each dimension, you fill in each cell with its respective severance events. We've completed the sample matrix with some characteristic events.

SA Type	Severance Criteria Algorithm: Default	Severance Criteria Algorithm: Customer On Life Support

Copyright © 2026, Oracle and/or its affiliates.

Electric Residential	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	Create a To Do entry asking a collection rep to call the customer. 5 days later, create a 72-hour warning field activity. 2 days after completion, create a To Do entry telling collection rep of impending life support cutoff. 3 days after completion of warning, create a disconnect service field activity AND generate a To Do entry informing a collection agent of such. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection expire the service agreement.
Electric Commercial	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	N/A
Gas Residential	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.
Gas Commercial	Create a 48-hour warning field activity. 2 days after completion, create a disconnect service field activity. Immediately after completion of the disconnect field activity, send a letter to the customer. 10 days after completion of disconnection, expire the service agreement.	N/A
Charitable Contribution	Expire service agreement	Expire service agreement

Copyright © 2026, Oracle and/or its affiliates.

Once the matrix is complete, you determine the severance process templates needed to implement your severance procedures. The following table shows the severance process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Severance Process Template	Event Number	Severance Event Template	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Utility severance - default	10	Field activity - 48 hour disconnect for non-payment warning	N/A - first event	0
	20	Field activity - disconnect for non-payment	10	2
	30	'Service has been disconnected' letter	20	0
	40	Expire service agreement	20	10
Electric life support residential severance	10	Generate delinquent life support customer To Do entry	N/A - first event	0
	20	Field activity - 72 hour disconnect for non-payment warning	10	5
	30	Generate impending life support cutoff To Do entry to C&C rep	20	2
	40	Field activity - cut for non-payment	20	3
	50	Service has been disconnected letter	40	0
	60	Expire service agreement	40	10
Just expire severance	10	Expire service agreement	N/A - first event	

If we extract each unique severance event type from the above table, we end up with the following:

Severance Event Template	Event Type
48-hour warning	Generate Field Activity - Disconnect Warning
72-hour warning	Generate Field Activity - Disconnect Warning
Disconnect for non payment	Generate Field Activity - Cut For Non-Payment
Delinquent life support customer	Create To Do Entry - C&C Rep Role

Copyright © 2026, Oracle and/or its affiliates.

Impending life support cutoff	Create To Do Entry - C&C Rep Role
Service has been disconnected letter	Send Letter - Customer Contact Type is Disconnect Letter
Expire service agreement	Expire Service Agreement

> **Warning**
>
> The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the type of service point, its state (e.g., connected, meter is off.) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

Now you're (almost) ready to set up your severance procedures.

- Defining Severance Process Auto Cancellation Criteria

Defining Severance Process Auto Cancellation Criteria

The topics in the section How Are Severance Processes Cancelled describe the two algorithms that play a part in the cancellation of a collection process. It also describes when to use what type of algorithm. Please read this section and then set up the appropriate cancellation criteria on your Debt Classes, and optionally, on your Severance Process Templates.

Designing Your Reconnection Procedures

If a customer pays for a service agreement after the service has been cut for non-payment AND BEFORE THE SA HAS BEEN EXPIRED, they need to be reconnected. Counter-intuitively, you must set up a severance process to initiate the field activities to reconnect service.

> **Note**
>
> **Why do you use a severance process to reconnect service?** Because a severance process is nothing more than a series of events that take place one after another. Some of the events create field activities, others send letters, others create To Do entries. So, why not use a severance process? You just have to send different letters and perform different field activities.

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> The system will automatically create a reconnection process if a severance process is cancelled as a result of a payment (or other credits). Please note that this will only happen if you plug-in the appropriate post cancellation algorithm on your severance process templates. Refer to What Happens When A Severance Process Is Cancelled? **for more information**.

While you don't define the reconnect procedures for an SA type, we recommend you think about the reconnection steps for each of your SA types that can be disconnected for nonpayment. We've completed the sample matrix with some characteristic events.

SA Type	Steps
Electric Residential	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.
Electric Commercial	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.
Gas Residential	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.
Gas Commercial	Create a reconnect service field activity. Immediately after completion of the reconnect, send a letter to the customer.

Once the matrix is complete, you determine the severance process templates needed to implement your reconnection procedures. Notice each SA type has the same reconnection steps. This means you just need one severance process. The following table shows this severance process template and its events.

Severance Process Template	Event Number	Severance Event Template	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Reconnect	10	Field activity - reconnect service	N/A - first event	0
	20	Service has been reconnected letter	10	0

If we extract each unique severance event type from the above table, we end up with the following:

Severance Event Template	Event Type
Reconnect	Generate Field Activity - Reconnect
Service has been reconnected letter	Send Letter

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the type of service point, its state (e.g., connected, meter is off.) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

> **Important**
>
> If you want the system to automatically create a reconnection process if a customer pays after they have been cut, you must specify the appropriate post cancellation algorithm on your severance process templates.

And now you're ready to set up your severance (and reconnection) procedures.

Setting Up Severance Procedures

In the previous section, Designing Your Severance Procedures, we presented a case study that illustrated a mythical organization's severance procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

- Setting Up Severance Event Types
- Setting Up Severance Process Templates

Setting Up Severance Event Types

Severance event types control what is done by a given severance event. Open **Admin**, and then **Credit & Collection**, and then **Severance Event Type**, and then **Add** to define your severance event types.

> **NOT_SUPPORTED**
>
> For more information refer to Designing Your Severance Procedures.

Description of Page

Enter a unique **Severance Event Type** code and **Description** for the severance event type.

Enter the Severance **Event Type**. Permissible values are: Affect Credit Rating/Cash-Only, Send Letter, Generic Algorithm, Create To Do Entry, Create Field Activities, Expire Service Agreement. The following discussion describes the parameters that must be defined for each type of severance event.

Copyright © 2026, Oracle and/or its affiliates.

The Affect Credit Rating/Cash-Only collection event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- account debt monitor.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record, the account is flagged as cash-only during payment processing and on Control Central.
- Use **Cash-Only Points** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record or the account's CIS Division, the account is flagged as cash-only during payment processing and on Control Central.

The Send Letter severance event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following **Parameters** for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

The Generic Algorithm severance event type causes the algorithm defined in the **Sev. Event Algorithm** to be executed. You use this type of algorithm when the standard types of severance events won't do what you need done.

The Create To Do Entry severance event type causes a To Do entry to be created. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-SEVT for the type of To Do entry that's created).

The Generate Field Activities severance event type causes one or more field activities to be generated. Enter the following **Parameters** for this type of event:

- Select the **Customer Event** associated with the field activity. Valid values are: Cut For Non-Payment (CNP), Disconnect Warning (Disc Warn), Reconnect for Payment (Reconn), Start Service (Start) , Start/Stop Service (Start/Stop), Stop Service (Stop), Reread (Reread).
- The system uses the **Customer Event** to select the appropriate field activity type(s) from the field activity type profile associated with the service points linked to the service agreement.

> **⚠ Warning**
>
> The field activity types are NOT specified directly on the severance event type. Why? Because each service point linked to the service agreement being severed could necessitate a different type of field activity. Therefore, the system uses the type of service point, its state (e.g., connected, meter off) and the desired customer event (e.g., Disconnect Warning, Cut For Non-Payment) to determine which field activity type(s) to generate. Refer to Setting Up Field Activity Type Profiles for how to set up the specific disconnect field activity types for your various types of service points.

The Expire Service Agreement severance event type causes the service agreement to be moved to the pending stop state (it also populates the service agreement's stop date). Refer to Finalizing Pending Stops for how the system will eventually stop the service agreement (and then final bill it). There are no parameters for this type of event.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Cut for non-payment field activities are special.** The Expire Service Agreement severance event type also makes any "cut for non-payment" field activities created by earlier severance events available to stop service. Specifically, it changes the linkage type of the field activities from Severance Activity to Stop Activity. You can see a service agreement's service points' field activities and their respective linkage type on Service Agreement - Service Point.

Enter a **Long Description** to fully describe the severance event type.

Setting Up Severance Process Templates

Severance process templates define the severance events that will be executed when a service agreement is severed. Open **Admin**, and then **Credit & Collection**, and then **Severance Process Template**, and then **Add** to define your severance process templates.

Description of Page

Enter a unique **Severance Process Template** and **Description** for the severance process template.

If severance processes of this type should be automatically canceled when the customer pays the collection amount on the severance, turn on **Auto Cancel**. This switch would typically only be turned off for severance processes used to reconnect a cut service because you don't want such a reconnection process to be canceled when a payment is made. Refer to Designing Your Reconnection Procedures for more information.

In addition to turning on the **Auto Cancel** switch, specify a **Cancel Criteria Algorithm** if your organization allows a severance process to be cancelled regardless of the debt associated with all service agreements in the debt class. In other words, if your cancel criteria are based on the debt associated with ALL service agreements in a debt class DO NOT SPECIFY THIS ALGORITHM.

If the **Cancel Criteria Algorithm** is specified, it is executed when a credit is posted to the service agreement associated with a severance process. This algorithm will indicate if the service agreement no longer has debt that warrants a severance process. Refer to How Are Severance Processes Cancelled for more information. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that cancels a severance process if the service agreement's debt so warrants.

If you wish to perform any special processes after a severance process is canceled, specify a **Post Cancel Algorithm**. This can be used to start a reconnection in case the severance process was canceled too late to stop the disconnection. If you haven't done so already, you must set up this algorithm in the system. To do this, create a new algorithm (refer to Setting Up Algorithms). On this algorithm, reference an Algorithm Type that cancels a severance process if the service agreement's debt so warrants.

When a service agreement is to be severed due to non-payment, the system creates a severance process and links to it one or more severance events based on the **Event Types** entered here. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event:

Copyright © 2026, Oracle and/or its affiliates.

Event Sequence controls the order in which the severance event types appear under the severance process template. The sequence number is system-assigned and cannot be changed. If you have to insert a severance event type between two existing templates, you'll have to remove the latter events, insert the new event, and then re-specify the removed events.

Severance Event Type Specify the type of severance event to be generated.

Dependent On Other Events Turn this indicator on if the trigger date of the event can only be determined after earlier events are complete. For example, you would turn this switch on for the event that initiates the field activity to disconnect service. Why? Because you only want to disconnect service after the preceding event that warned the client of impending disconnection is complete.

Days After Prev Response Specify the number of days after the completion / cancellation of the dependent events on which the severance event will be triggered. If this event is not dependent on the completion of other events, this field contains the number of days after the creation of the severance process that the related severance event will be triggered. Refer to Severance Event Dependencies and Trigger Date and Calendar vs Work Days for a description of how the system uses this information to set the trigger date on the respective severance events.

When the **Dependent On Other Events** switch is on, use the grid to define the events on which this event is dependent. If multiple events are specified in the grid, all such events must be completed or cancelled before the event will be triggered.

Sequence is system-assigned and cannot be specified or changed.

Dependent On Sequence Specify the sequence number of the severance event on which the above severance event depends.

NOT_SUPPORTED

For more information about severance event types, see Setting Up Severance Event Types. For more information about trigger dates, see Severance Event Dependencies & Trigger Date.

Designing Your Write-Off Procedures

The design of your write-off procedures is relatively straightforward. Simply follow the instructions in the following topics.

- Designing Your Write-Off Debt Classes
- Designing Write-Off Controls
- Designing Write Off Process Templates & Write Off Event Types

Designing Your Write-Off Debt Classes

Multiple write-off debt classes are needed when you have different write-off procedures for different types of service agreements. If all service agreement debt is written-off the same way, then you'll have just one write-off debt class (call it Generic). However, if you're like many

Copyright © 2026, Oracle and/or its affiliates.

organizations, you will have multiple write-off debt classes. The following points will help you understand why:

- If you bill for 3 rd parties, you probably have different write-off debt classes for the 3 rd party service agreements. Why? Because you probably treat 3 rd party uncollectable debt differently from your own debt.
- You will need a separate write-off debt class for service agreements whose debt cannot be written off. Why? Because there is a switch on the write-off debt class control table that controls if service agreements in the write-off debt class are eligible for write-off processing. Given that you will have some service agreements that hold debt that aren't eligible for write-off processing (e.g., service agreements that hold written-off debt and service agreements that overpayments), you will need at least one other write-off debt class.
- If you use the system to calculate charges for your organization's company usage, you'll need another write-off debt class (we refer to it as the "N/A" write-off debt class below). Why? Because all service agreements must have a write-off debt class, even those that will never have debt.

SA's Write-Off Debt Class	**Account's Collection Class**	**Account's Collection Class**
Normal W/O		
N/A		

Designing Write-Off Controls

Set up a matrix using the collection classes you designed when you were designing your collection procedures (Designing Your Collection Procedures).

SA's Write-Off Debt Class	**Account's Collection Class Residential**	**Account's Collection Class Commercial/Industrial**
Normal Write Off		
N/A		

Each cell should have a "write-off control" that defines what to do when the system detects finaled debt that hasn't been paid. This is true even of the "N/A" write-off debt class. Why? Because you may want the system to write-down these stopped SAs when they have a small balance. For example, if you have a write-off service agreement that subsequently receives a partial payment that leaves a small amount owing, you probably want the system to generate a write-down adjustment (so that the write-off service agreement will close). We'll initially fill in the matrix for the "N/A' write-off debt class.

SA's Write-Off Debt Class	**Account's Collection Class Residential**	**Account's Collection Class Commercial/Industrial**
Normal Write Off		

Copyright © 2026, Oracle and/or its affiliates.

N/A	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment.

> **Note**
>
> If the Write Off Monitor encounters debt associated with a non-defined collection class and write-off debt class, it will issue an error.

For each cell that isn't designated as N/A, you need to answer the following questions:

- Are you allowed to transfer debt to other non-closed service agreements linked to the account? If so, you need to define the algorithm used to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- Are you allowed to write-down small amounts of debt (or small credits)? If so, you need to define the algorithm used to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- Should you refund credit balances with a check? If so, you need to define the algorithm to do such. Refer to Setting Up Write-off Control for more information about this algorithm.
- If debt remains after doing the above, how do you write it off (e.g., do you first refer the debt to a collection agency and only write it off after waiting 30 days)?

We'll fill in the above matrix with our assumptions:

SA's Write-Off Debt Class	**Account's Collection Class Residential**	**Account's Collection Class Commercial/Industrial**

Copyright © 2026, Oracle and/or its affiliates.

Normal Write-Off	Attempt to reduce the SA's balance to zero using the following methods: Attempt to transfer debt to another active service agreement linked to the account. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. If the debt is <= $-1, create an A/P adjustment to refund the credit to the customer. If debt remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for residential debt.	Attempt to reduce the SA's balance to zero using the following methods: Attempt to transfer debt to another active service agreement linked to the account. If the debt / credit is < $10 and > $-10, write down the debt using a write-down adjustment. If the debt / credit is <= $-10, create an A/P adjustment to refund the credit to the customer. If debt remains: Highest priority: If customer has a non-cash deposit, create the non-cash deposit write-off process. Otherwise, create the default write-off process for commercial debt.
N/A	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. Because this debt class isn't eligible for further write-off processing, criteria used to process debt are not necessary.	Attempt to reduce the SA's balance to zero using the following methods: Synchronize current balance with payoff balance. If the debt is < $10 and > $-1, write down the debt using a write-down adjustment. Because this debt class isn't eligible for further write-off processing, criteria used to process debt are not necessary.

We can now use the information in the above matrix to design the necessary Write Off Process Templates and Write Off Event Types.

Designing Write Off Process Templates & Write Off Event Types

The following table shows the write-off process templates referenced in the previous section's matrix. Adjacent to each process are its events and an indication of when they are triggered.

Write Off Process Template	Write Off Event Type	Triggered X Days From Start Of Write Off Process
Residential	Refer to collection agency	0
	Letter notifying customer of referral	0
	Cancel collection agency referral	60
	Write off	60

Copyright © 2026, Oracle and/or its affiliates.

Non-Cash Deposit Exists	To Do for non-cash deposit redemption	0
	To Do to highlight unpaid SA(s) still exist (and they will be reconsidered for write-off processing)	10
Commercial	Refer to collection agency	0
	Letter notifying customer of referral	0
	To Do to check up on collection agency's efforts	30
	Cancel collection agency referral	60
	Write off	60

If we extract each unique event type from the above table, we end up with the following:

Write Off Event Type	Event Type
Notification of write-off referral	Send letter - Debt referred to a collection agency
Refer to collection agency	Refer to collection agency
Cancel collection agency referral	Cancel collection agency referral
Write off	Write off
To Do for non-cash deposit redemption	Generate To Do - Redeem non-cash deposit
To Do to highlight unpaid SA(s) still exist	Generate To Do - SA(s) linked to a non-cash deposit remain unpaid (and will be reconsidered for write-off processing)
To Do to check up on collection agency's efforts	Generate To Do - Check up on collection agency's efforts

And now you're ready to set up your write-off procedures.

Setting Up Write-Off Procedures

In the previous section, Designing Your Write-Off Procedures, we presented a case study that illustrated a mythical organization's write off procedures. In this section, we'll explain how to set up the control tables to implement these procedures:

- Setting Up Write Off Debt Classes
- Setting Up Write Off Event Types
- Setting Up Write Off Process Templates
- Setting Up Write Off Control
- Setting Up Collection Agencies

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Write Off Debt Classes

Every SA type has a write-off debt class. This class is one of several fields that control the write off criteria applied to the service agreement's debt. Select **Admin**, and then **Credit & Collection**, and then **Write Off Debt Class** to define your debt classes.

> **NOT_SUPPORTED**
>
> For more information about debt classes, see Designing Your Write-Off Debt Classes.

Panel controls

To modify a write-off debt class, simply move to a field and change its value. To add a new write-off debt class, click + to insert a row, then fill in the information for each field. The following fields display:

Write Off Debt Class Code The unique identifier of the write off debt class.

Eligible for Write Off Indicates if service agreements belonging to this write off debt class are eligible for write-off processing. This should only be turned off if this debt cannot be written off, e.g., write off debt.

Description The description of the write off debt class.

Setting Up Write Off Event Types

Write-off event types control what is done by a given write-off event. Select **Admin**, and then **Credit & Collection**, and then **Write Off Event Type**, and then **Add** to define your write-off event types.

Description of Page

Enter a unique **Write Off Event Type** code and **Description** for the write-off event type.

Select the **Write Off Event Type**. Permissible values are: Affect Credit Rating/Cash-Only, Cancel Agency Referral, Generic Algorithm, Refer to Agency, Send Letter, Create To Do Entry, Write Off using Distrib Code, and Write Off using SA Type. The following discussion describes the parameters that must be defined for each type of write-off event.

The Affect Credit Rating/Cash-Only write-off event type causes a credit rating demerit record to be linked to the account. This record is constructed using the following **Parameters**:

- Use **Affect Credit Rating By** to define this event's affect on the account's credit rating. This should be a negative number. An account's credit rating is equal to the beginning credit rating amount defined on the installation record or the account’s CIS Division plus the sum of credit rating demerits that are currently in effect. When an account's credit rating is less than the credit rating threshold defined on the installation record or the account’s CIS Division, the account's credit rating is displayed as an alert on Control Central.
- Use **Affect Cash-Only Score By** to define this event's affect on the account's cash-only score. This should be a positive number. When an account's cash-only score exceeds the

Copyright © 2026, Oracle and/or its affiliates.

cash-only threshold score defined on the installation record or the account's CIS Division, the account is flagged as cash-only during payment processing and on Control Central.

- Use **Months Affecting Credit Rating** to define the length of time the demerit remains in effect. This information is used to define the effective period of the credit rating demerit record.

> **NOT_SUPPORTED**
>
> For more information, refer to Account - Credit Rating.

The Cancel Agency Referral event type will cancel previous collection agency referrals. No parameters are needed for this type of event.

The Generic Algorithm write-off event type causes the algorithm defined in the **Generic Algorithm** to be executed. You use this type of algorithm when the standard types of write off events won't do what you need done. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the algorithm that will be called when events of this type are activated.

The Refer to Agency event type will refer the debt associated with the process' SAs to a collection agency. You must supply the **Agency Selection Algorithm** that is used to determine the collection agency associated with the referral. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines the collection agency to which bad debt should be referred.

> **Note**
>
> **Letters.** You must set up a customer contact type for each type of letter you generate. You specify the necessary customer contact type on the write off event type. Refer to Setting Up Letter Templates for more information.

The Send Letter write-off event type causes a customer contact to be generated that, in turn, generates a letter. Enter the following parameters for this type of event:

- Select the **Contact Class** used to categorize the customer contact.
- Use **Contact Type** to define the type of customer contact to create. The type of customer contact controls the type of letter that is generated.

The Create To Do Entry write-off event type causes a To Do entry to be issued. Refer to The Big Picture of To Do Entries for more information about To Do entries (refer to the To Do type TD-WOEVT for the type of To Do entry that's created).

The Write Off using Distrib Code event type causes bad debt to be written off in accordance with the distribution codes associated with the financial transactions that caused the debt in the first place. Use this method if, for example, you want to write-off revenue differently than you write-off liabilities. When this type of event is activated, the system accumulates the distribution

Copyright © 2026, Oracle and/or its affiliates.

codes from GL details associated with recent financial transactions linked to each write-off service agreement. When the system has accumulated enough distribution codes (i.e., where the amount associated with the distribution code equals or exceeds the amount to write off), the debt will be transferred to a new or existing write-off service agreements. The type of service agreements to which the debt is transferred is defined on the distribution codes. Refer to Setting Up Distribution Codes for more information.

The Write Off using SA Type event type causes all debt associated with the process' SAs to be transferred to a write-off service agreement linked to the account. Enter the following **Parameters** for this type of event:

- **CIS Division / SA Type** is the type of write-off service agreement to which the debt will be transferred. Note well,
  - The system will reuse an existing service agreement if an active SA of this type is already linked to the account; otherwise the system will create a new service agreement of this type.
  - The adjustment type used to set the offending service agreement's current balance equal to its payoff balance is defined on the write-offable SA Type. Refer to SA Type - Main Information for more information.
  - The adjustment type used to transfer the delinquent debt to the write-off service agreement is defined on the write off SA type. Refer to SA Type - Detail for more information.

Enter a **Comment** to fully describe the write-off event type.

Setting Up Write Off Process Templates

Write-off process templates define the write-off events that will be executed when a write-off criteria rule is violated. Select **Admin**, and then **Credit & Collection**, and then **Write Off Process Template**, and then **Add** to define your write-off process templates.

Description of Page

Enter a unique **Write Off Process Template** code and **Description** for the write-off process template.

The rows in the following grid define the events that will be created when a write off process is created using this template. The following fields display:

Event Sequence Controls the order in which the write-off event is executed. The sequence number is system assigned and cannot be changed. If you need to insert a write-off event between two existing events, you must remove the latter events, insert the new event, and then re-enter the removed events.

Write-off Event Type Code Specify the type of write-off event to be generated. The event type's description is displayed adjacent.

Days After Process Creation Specify the number of days after the creation of the write-off process that the related write-off event will be triggered. Refer to Calendar vs Work Days for a description of how this system uses this information to set the trigger date on the respective write-off events.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about write-off event types, see Setting Up Write Off Event Types. For more information about trigger dates, see Write-off Event Trigger Date.

Setting Up Write Off Control

Write-off controls define how the system handles finaled, unpaid debt belonging to a given collection class and write off debt class.

- Write Off Control - Main
- Write Off Control - Criteria

Write Off Control - Main

Select **Admin**, and then **Credit & Collection**, and then **Write Off Control**, and then **Add** to define basic information about a write-off control. After entering basic information, navigate to the **Criteria** tab to define the type of write-off process to start when given criteria are met.

NOT_SUPPORTED

For more information about write-off control, refer to Designing Write-Off Controls.

Panel controls

Enter a **Write Off Control** code and **Description** for the write-off control (WOC).

Enter the **Collection Class** to which the WOC applies.

Enter the **Write Off Debt Class Code** to which the WOC applies.

Enter general **Comments** to further describe the WOC.

Define the **Synch All Algorithm** used by the system to generate adjustments that cause current balance to equal payoff balance on the service agreements to be written off. This type of algorithm is issued before you actually start a write-off process as current balance is meaningless at write-off time (the customer owes you the payoff balance). If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that synchronizes current and payoff balances.

Define the **Debt Transfer Algorithm** used by the system when it attempts to transfer the unpaid debt to another active service agreement linked to the stopped SA's account. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that transfers unpaid balances.

Copyright © 2026, Oracle and/or its affiliates.

Define the **Write Down Algorithm** used by the system when it attempts to write-down small debt and/or credit balances. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that writes down small amounts.

Define the **Credit Refund Algorithm** used by the system when it refunds a credit balance to a customer. If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that refunds credit balances.

Write Off Control - Criteria

Select **Admin**, and then **Credit & Collection**, and then **Write Off Control**, and then **Search** and navigate to the **Criteria** page to define the type of write-off process to start when given criteria are met.

> **NOT_SUPPORTED**
>
> The following information is not intuitively obvious. Refer to Designing Write-Off Controls for more information.

Panel controls

The information in the grid defines the write-off process to be executed for debt belonging to the previously defined collection class and write off debt class. The type of write-off process may differ depending on some condition. For example, you may have a different write-off process if the customer has a non-cash deposit. You must have at least one entry in this collection otherwise the system will not start a write-off process. This entry should have the lowest priority code and should reference a **Write Off Criteria Algorithm** that references the WO CRIT DFLT the algorithm type.

The following fields display:

Priority controls the order in which the criteria will be checked by the Write Off Monitor (higher priorities are checked before lower priorities). The first criteria algorithm that is met (i.e., returns a value of *True*) will cause the associated write-off process to be initiated.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is CRIT_PRIO_FLG. Be aware that this field is used for multiple tables: Collection Class Control, Severance Criteria, and Write Off Control .

Write Off Criteria Algorithm The Write Off Monitor checks if the condition defined by the W/O Condition Algorithm applies to the account whose debt is being analyzed. If a condition is met, a write-off process is created using the associated write-off process template.

Copyright © 2026, Oracle and/or its affiliates.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if a customer's bad debt should be processed using the associated **Write Off Process Template**.

> **Important**
>
> You must have at least one entry in this grid otherwise the system will not start a write-off process. This entry should have the lowest priority code and should reference a **W/O Criteria Algorithm** that references the WO CRIT DFLT algorithm type.

Write Off Process Template If the Write Off Monitor determines the condition defined by the w/o condition algorithm applies, a write-off process is created using the associated write-off process template.

Where Used

Write-off controls contain the data that controls the Write Off Monitor. Refer to How Does The Write-Off Monitor Work? for more information.

Setting Up Collection Agencies

You must set up a collection agency for each such organization to which you refer delinquent debt. To define a collection agency, select **Admin**, and then **Credit & Collection**, and then **Collection Agency.**

Description of Page

Enter an easily recognizable **Collection Agency** code and **Description** for each collection agency.

A collection agency must be associated with a Person. Choose the **Person ID** of the organization from the prompt.

> **NOT_SUPPORTED**
>
> Information about how to set up persons is discussed in Maintaining Persons.

Turn on the **Active** switch if the collection agency is actively receiving referrals.

Specify the **Batch Control** that's used to route new and cancelled referrals to the collection agency. The batch control's description is displayed adjacent.

Where Used

Collection agencies get assigned to collection agency referrals when the collection agency referral background process executes. Refer to How Do Collection Agency Referrals Work? for more information.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Feature Configuration

You must set up a Feature Configuration if you use the champion / challenger functionality.

The following describes the various **Option Types** that must be defined:

Champion Template$Challenger Template$Percentage(1-100). You need only set up options of this type if your implementation implements Champion / Challenger functionality. Options of this type are entered in the format ABnnn where A is the collection process template of the champion template, B is the collection process template of the challenger template, and C is the percent of the time that the system should create the challenger template. The collection monitor uses this option to override the champion collection process template X% of the time with the challenger template. You may enter any number of these options (but only one per Champion Template).

How To

The contents of this section describe how to set up various credit and collections scenarios.

- How Arrears History is Captured
- How To Nominate A Single Service To Sever (Rather Than Sever Everything That's In Arrears)

How Arrears History is Captured

The system captures a history of arrears (amount owed) for accounts under an active collection activity – those with an active collection, severance, or overdue process. The data is maintained from the creation of the process to when it is set to inactive. Currently, there is no user interface to view the history records. Rather, they are stored in the system so they can be retrieved by other processes and APIs.

Where is this information captured? The system uses child maintenance objects that extend their parent object to capture this information. Arrears history records are stored in the following tables: Collection Process Arrears History, Collection Process SA Arrears History, Severance Process Arrears History and Overdue Process Arrears History. Arrears history is stored on the Collection Process SA Arrears History table when the system is configured to use service agreement-oriented cancel criteria. Refer to How Are Collection Processes Cancelled and How Are Severance Processes Cancelled for more information on debt class or service agreement-oriented cancel criteria configurations.

Why is this information captured? One reason is because it's difficult to derive real-time. The system relies on plug-in spots to know if a collection activity can be canceled. These plug-in spots calculate the amount owed under a collection activity, sometimes adjust that amount by promised future payments, and compare that amount to a threshold that is configured as an algorithm parameter. So, outside of the system processes that call these plug-in spots, it is difficult for an API or query to replicate the logic. Therefore, the system captures the information from the plug-in spot drivers – the programs that call the plug-ins. The drivers are base programs and will work with algorithms delivered with the system or your custom algorithms; just make sure they are populating the output elements for the plug-in spot.

Copyright © 2026, Oracle and/or its affiliates.

The above explains the technical aspect of this, but why or how can this data be used? One reason is to allow inquiries across many accounts belonging to the same person or business. Consider a large entity such as a local government with hundreds or even thousands of accounts and the desire to know how much debt is under collection at any moment. Aside from being difficult to replicate the logic that derives these amounts, that query could be a timely one to perform.

The most recent amount owed and related information should meet the business needs for cases like the one described above. However, instead of keeping a current snapshot of amount owed for each process, the system creates a record upon process creation and a new record each time there is a change in the amount owed or other data captured. One reason for this is to have the data to analyze the payment patterns for debt under collection. Is the debt paid in one lump sum or in small amounts? Is it paid close to the process creation or days or weeks later? Are customers paying the full amount owed or just enough to get below the threshold to cancel collection activity? This data makes these types of analytics possible.

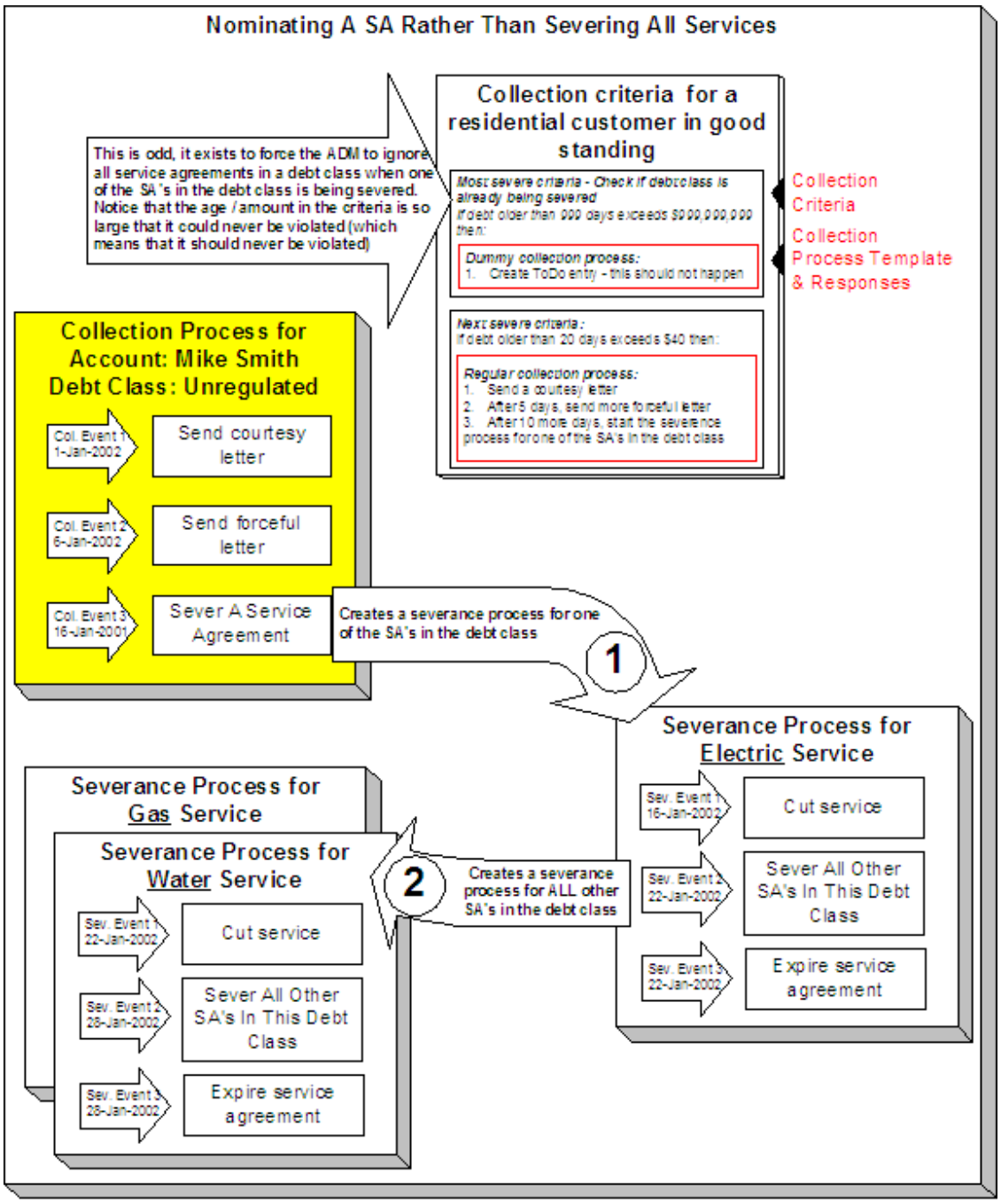
What data is being stored? The key information is the amount still owed for the collection activity, the cancelation threshold amount, and the amount owed before being adjusted by future payments. Refer to How Pay Plans Affect The Account Debt Monitor for more information about how pay plans affect the amount owed. The system is also able to determine if debt class or service agreement-oriented cancel criteria configurations are used based on which plug-in spots are being used and this information is captured. This is important in knowing how to use and interpret the data. Refer to the system metadata for these tables to see exactly what data is captured.

What system processes call the plug-in spots responsible for capturing the arrears history? These plug-in spots are called when the processes that collect on debt are created. They are called in a special mode that does the calculations but doesn't take any action. This is to capture the initial record. These plug-in spots are also called by the processes that monitor ongoing collection of debt, some updates to these processes, and by system events such as payments or other credit financial transactions.

How To Nominate A Single Service To Sever (Rather Than Sever Everything That's In Arrears)

Some organizations that offer multiple services do not sever all services when the customer falls into arrears. Rather, they nominate a single service agreement to sever in the hopes that the lack of service will cause the customer to remit payment. The following diagram illustrates the control tables values required to implement this type of requirement.

Copyright © 2026, Oracle and/or its affiliates.



The following important concepts are illustrated above:

- The collection process's last event does NOT sever all services. Rather, it calls an algorithm that selects a single service to sever. A base package algorithm allows you to define the primary service to sever and a secondary service to sever (if the customer does not have the primary service). If you sold electricity and gas, you would probably define the primary service as electricity and the secondary as gas (because electricity is easier to cutoff than gas).
- The severance process that is started for the primary service cuts service. If the customer doesn't remit what is owed, the second severance event calls an algorithm that severs all other service agreements in the debt class.

Copyright © 2026, Oracle and/or its affiliates.

- Because you are nominating a single service to sever, you must set up a special value in collection class control to force the ADM to ignore all service agreements in a debt class when one of the SAs in the debt class is being severed. Notice that the age / amount in the criteria is so large that it could never be violated.
- In addition, because the entire debt class must no longer be in arrears to stop the collection and severance processes, you must plug-in the appropriate collection process and severance process cancellation criteria on the debt class. Refer to How Are Collection Processes Cancelled and How Are Severance Processes Cancelled for more information about how these algorithms are used. Also note, you do not need service agreement cancellation criteria defined on your collection process templates and severance process templates (because cancellation is controlled at the debt class level).

Defining Asset Options

> **Note**
>
> The asset management functionality described in this section is available with the following products:
>
> - Oracle Utilities Advanced Meter Solution (C2M-AMS)
> - Oracle Utilities Customer to Meter (C2M)
> - Oracle Utilities Customer Cloud Service (CCS)
> - Oracle Utilities Meter Data Management (MDM)
> - Oracle Utilities Meter Solution Cloud Service (MSCS)

- Asset Activity Types
- Asset Completion Event Types
- Asset Inbound Communication Types
- Asset Types
- Service History Questions
- Service History Types
- Configuration Types
- Configurations
- Configuration Reports

Asset Activity Types

- Understanding Asset Activity Types
- Defining Asset Activity Types

Understanding Asset Activity Types

Copyright © 2026, Oracle and/or its affiliates.

Activity types define the main attributes broad range of activities, such as maintenance, repairs, inspections, field work etc. The base package includes an asset activity type business object (W1-SOMActivityType) that should be used when created asset activity types.

Activity types are defined by:

- **Activity Business Object**: The business object used for asset activities of this type
- **Activity Schedule Information**: Default scheduling information for asset activities of this type, including Activation Lead Days, Work Lead Days, and Activity Disposition.
- **Attachments**: One or more attachments associated with asset activities of this type, such as procedures or checklists.
- **Exception Handling**: Details regarding exception handling for asset activities of this type, including default To Do Type, To Do Role, and retry options.
- **Completion Event Types**: One or more completion event types associated with asset activities of this type. See Understanding Completion Event Types for more information.
- **Object To Create**: The service history type to be created by asset activities of this type. Service history records are created by Create Activity Type's Service History completion events.

Refer to Defining Asset Activity Types for more information about setting up asset types.

> **Note**
>
> Asset activity types are created by the system when field task types are created in Service Order Management.

Defining Asset Activity Types

You use the **Activity Type** portal to display and maintain asset activity types.

Refer to Understanding Asset Activity Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Activity Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type**: This zone displays basic information about the activity type.
- **Attachments**: This zone attachments associated with the activity type.

Asset Completion Event Types

- Understanding Asset Completion Event Types
- Defining Asset Completion Event Types

Understanding Asset Completion Event Types

Copyright © 2026, Oracle and/or its affiliates.

Asset completion event types are used to define various processes or events to be executed when completing an asset activity such as asset replication, service history creation, attachment creation, and activity creation.

Asset completion event types are referenced on activity types to indicate which completion events should be included as part of the lifecycle of activities of that type. For example, this affects when service history is created for an activity, and the disposition of the assets on an activity as the activity is processed. Asset completion events can be configured as optional or required for activities of different types.

Asset completion event types are defined by:

- **Completion Event Business Object**: The business object used for asset completion events of this type
- **Identifiers**: One or more identifiers that can be used to characterize asset completion events of this type
- **Inbound Communication**: Information used to create asset completion events of this type using information defined within an inbound communication business object, or within its raw message
- **UI Payload Create Data Mover**: XPath statements executed when creating asset completion events of this type online. These statements are used to populate elements within the payload from the completion event business object.

Refer to Defining Completion Event Types for more information about setting up completion event types.

Defining Asset Completion Event Types

You use the **Completion Event Type** portal to display and maintain asset completion event types.

Refer to Understanding Asset Completion Event Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Completion Event Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Completion Event Type** : This zone provides information about the selected completion event type, including as inbound communications and attachments.
- **Unlink Activity Types**: This zone enables the removal of an activity type from the completion event type. To remove an activity type select the activity type(s) to remove and click **Remove**. The selected activity types will no longer appear in the **Unlink Activity Types** zone.
- **Link Activity Types**: This zone enables adding an activity type to the completion event type. To link an activity type:
    1. Enter search criteria n the **Activity Type Description** field and click **Search**. The zone will display a list of activity types matching your search criteria.
    2. Select the activity type(s) to add and click **Link**. The selected activity types will now appear in the **Unlink Activity Types** zone.

Copyright © 2026, Oracle and/or its affiliates.

Asset Inbound Communication Types

- Understanding Asset Inbound Communication Types
- Defining Asset Inbound Communication Types

Understanding Asset Inbound Communication Types

Asset inbound communication types define the information required for the system to receive messages from external systems. These typically hold value mappings of the inbound communication elements to an xpath element in the completion message.

Asset inbound communication types are defined by:

- **Inbound Communication Business Object**: The business object used for asset inbound communications of this type
- **Exception Handling**: Details regarding exception handling for asset inbound communications of this type, including default To Do Type, To Do Role, and retry options.
- **Message Validation**: Source elements within the inbound communication business object or raw message used for validation of asset inbound communications of this type
- **Communication Data Mapping**: Information used to populate communication business object elements from information within the communication's raw message
- **Attachment Suppression**: Substitution text used in place of large data attachments in the completion event list on the inbound communication. Used when messages received from an external system contain attachment data that are very large.

Refer to Defining Asset Inbound Communication Types for more information about setting up asset inbound communication types.

Defining Asset Inbound Communication Types

You use the **Communication Type** portal to display and maintain asset inbound communication types.

Refer to Understanding Asset Inbound Communication Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Inbound Communication Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Communication Type List**: This zone lists all asset inbound communication type records. Broadcast a record to display the details of the selected record.
- **Communication Type**: This zone provides information about the selected communication type, such as attachment suppression, communication data mapping, message validation and exception handling.

Asset Types

Copyright © 2026, Oracle and/or its affiliates.

- Understanding Asset Types
- Defining Asset Types

Understanding Asset Types

Asset types define the attributes for assets and components of a certain type including information such as the valid location types and service history types for the assets and whether or not such assets can have attached components.

Asset types are defined by:

- **Component Relationship**: A flag that indicates if components can be installed on assets or components of this type
- **Asset Business Object**: The business object used for assets or components based on this asset type
- **Valid Location Types**: A list of location types where assets or components of this type can be located
- **Valid Attached To Asset Types**: A list of valid asset types where components of this type can be attached (applicable to components only)
- **Valid Service History Types**: A list of valid service history types that can apply for assets or components of this type

Refer to Defining Asset Types for more information about setting up asset types.

Asset Classes

Assets are defined by classes, each of which is based on a specific business object, and determines the type of asset: an asset or a component.

The table below lists the asset classes and the business object for each.

Asset Class	Business Object
Asset	W1-AssetType (Operational Device Asset Type)
Component	W1-ComponentType (Operational Device Component Type)

Defining Asset Types

You use the **Asset Type** portal to display and maintain asset types.

Refer to Understanding Asset Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Type** : This zone provides information about the asset type.
- **Asset Type Statistics**: This zone displays statistics for the asset type.

Copyright © 2026, Oracle and/or its affiliates.

Service History Questions

- Understanding Service History Questions
- Defining Service History Questions

Understanding Service History Questions

Service History Questions are used to guide workers through corrective and preventive inspection processes as part of an asset's service history.

Service history questions are associated to a service history type which is then associated to an asset type.

During a corrective or inspection process, workers can be guided through a series of questions assessing the condition of the asset. The answers can be "weighted", which when calculated, gives an overall score as to the condition of the asset. The calculation is generally a sum (answer points * question weight), but can vary depending on the service history type. The total inspection points are then normalized to an overall grade or score. The condition rating and confidence rating are also calculated and the asset is updated with the appropriate values.

A service history question's answers can be defined as check boxes, multiple choice, or free-form text.

Refer to Defining Service History Questions for more information about defining service history questions.

Defining Service History Questions

You use the **Service History Question** portal to display and maintain Service History questions.

Refer to Understanding Service History Questions for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Service History Question**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Question**: This zone provides information about the question.
- **Service History List**: This zone displays a list of service history types that reference the question.

Service History Types

- Understanding Service History Types
- Defining Service History Types

Copyright © 2026, Oracle and/or its affiliates.

Understanding Service History Types

Service history types define the main attributes for service history of a certain type. Types might include maintenance, inspection, service, test, measurement, and so on.

Service history types are defined by the following:

- **Service History Business Object**: The business object used to create service history records of this type.
- **Category**: The category to which service history records of this type belong. The base package includes the following categories:
  - Downtime
  - Failure
  - Inspection
  - Maintenance
  - Repair
- **Subclass**: The specific subclass of the service history type, based on the **Service History Business Object**. Oracle Utilities Customer To Meter supports two subclasses of service history types:
  - Operational Device Service History
  - Questionnaire Service History

  Each service history type subclass includes other specific configuration information, outlined below.

Operational Device Service History Types

Operational Device Service history types are also defined by the following:

- **Service Schedules**: One or more template work orders and time intervals that should be used to schedule work order generation for service history records of this type

Questionnaire Service History Types

Questionnaire Service history types are also defined by the following:

- **Follow-up Actions**: One or more actions intended as “follow-up” actions for service histories of the service history type, such as work orders or work order activities, users can configure those follow-up actions on the service history type. Note that if a regulatory compliance requirement is associated with a follow-up action, a compliance type must be specified. For example, a Service History Type of “Grade 2 Gas Leak”, might include a follow-up action of “Recheck” which could be configured with a “30 day” Compliance Type and a follow-up action of “Repair” which could be configured with a “1 year” Compliance Type.
- **Stop Processing Service History Type**: One or more service history types that should stop the processing of **Follow-ip Actions**. For example, certain problem that needs follow up actions was reported. Then another service history, which indicates the problem has been fixed or its severity has changed, is recorded. Businesses might want to stop/cancel the follow up actions based on the original problem/severity.
- **Questions**: A list of one or more questions used to guide inspection of an asset. Questions are defined in the **Questions** portal.

Copyright © 2026, Oracle and/or its affiliates.

Defining Service History Types

You use the **Service History Type** portal to display and maintain service history types.

Refer to Understanding Service History Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Service History Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service History Type**: This zone provides information about the service history type.
- **Questionnaire Tree**: This zone displays the service history types' questionnaire, based on the questions defined for the service history type. Applicable for Questionnaire service history types only.
- **Unlink Asset Types**: This zone lists the asset types currently linked to the service history type. To unlink one or more asset types, check the corresponding checkboxes and click **Remove**.
- **Link Asset Types**: This zone is used to search for and link asset types to the service history type. To link new asset types,
  1. Enter search criteria in the **Asset Type Description** field.
  2. Click **Search**.
  3. Select the asset types to link by checking the corresponding checkboxes.
  4. Click **Link**.

Configuration Types

- Understanding Configuration Types
- Defining Configuration Types

Understanding Configuration Types

Configuration types define the information required for various asset configurations with other assets, components, firmware, and other data.

Configurations are referenced on specifications which are then referenced on assets and components to indicate the best way that these entities can work with the other entities that they are connected to in the field.

Configuration types are defined by the following:

- **Configuration Class**: The class for configurations of this type
- **Configuration Category**: The category of configurations of this type
- **Configuration Business Object**: The business object used for configurations of this type

Copyright © 2026, Oracle and/or its affiliates.

Configuration Classes

In general configuration classes define the types of connections that are being configured. Each configuration class is based on a specific business object, which is the business object used for configurations of each class and type.

The table below lists the configuration classes and the business object for each.

Configuration Class	Business Object
Single Asset	W1-SingleAssetConfig (Single Asset Configuration)
Asset/Component	W1-AssetCompConfig (Asset/Component Configuration)
Asset to Asset	W1-AssetToAssetCompConfig (Asset to Asset/Component Configuration)
Asset/Component to Asset	W1-AssetCompToAssetConfig (Asset/Component to Asset Configuration)
Asset to Asset/Component	W1-AssetToAssetConfig (Asset to Asset Configuration)
Asset/Component to Asset/ Component	W1-AssetCompToAssetCompConfig (Asset/Component to Asset/ Component Config)

Defining Configuration Types

You use the **Configuration Type** portal to display and maintain configuration types.

Refer to Understanding Configuration Types for more information.

You can access the portal by selecting **Admin**,, then **Asset Management**, then **Configuration Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration Type List**: This zone lists all configuration type records. Broadcast a record to display the details of the selected record.
- **Configuration Type** : This zone provides information about the selected configuration type.

Configurations

- Understanding Configurations
- Defining Configurations

Understanding Configurations

Configurations document and manage valid setups for assets and components. This includes a specification for each asset or component in the configuration as well as the valid value for each parameter, such as the firmware version of each asset.

Configurations are defined by the following:

- **Configuration Type**: The configuration type upon which the configuration is based

Copyright © 2026, Oracle and/or its affiliates.

- **Status**: The current status of the configuration
- **Configuration Detail**: Specific information related to the configuration such as the assets and components involved, the firmware, characteristics, identifiers and other important information

Defining Configurations

You use the **Configuration** portal to display and maintain configurations.

Refer to Understanding Configurations for more information.

You can access the portal by selecting **Admin**, then **Configuration Report**, then **Configuration**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration**: This zone provides information about the selected configuration.
- **Configuration Attachments**: This zone lists any attachments defined for the configuration.

Configuration Reports

- Understanding Configuration Reports
- Defining Configuration Reports

Understanding Configuration Reports

Configuration reports provide a summary of any currently installed configurations.

These reports show whether the configuration in an installation is valid or invalid, the identifiers and values being used, and display all of the assets/components and other elements that make up the installed configuration.

Configuration reports are defined by the following:

- **Reporting Mode**: The mode in which the report is run (One Time or Recurring)
- **Status**: The current status of the report (Pending or Submitted)
- **Report Criteria**: Specific information used to establish the filter criteria when generating the report, such as the assets and components involved, the firmware, characteristics, identifiers and other important information.
- **Report Result**: Specific information for each configuration, including the number of configurations, the assets and components involved, and the specifications and firmware versions of each. If the configuration is invalid, the **Configuration** column is blank in the report results. Users can navigate to specific valid configurations to review and edit the setup as needed. In addition, users can generate a work order to correct, reconfigure, or otherwise maintain any of the configurations by clicking **Generate Activity**.

Copyright © 2026, Oracle and/or its affiliates.

Base Package Configuration Reports

The base package includes a number of configuration reports, each based on one of the support configuration classes (see Understanding Configuration Types.

The table below lists the configuration reports and the business object for each.

Configuration Report	Business Object
Single Asset Configuration Report	W1-SingleAssetCfgRpt
Asset/Component Configuration Report	W1-AssetCompCfgRpt
Asset to Asset Configuration Report	W1-AssetToAssetCfgRpt
Asset/Component to Asset Configuration Report	W1-AssetCompToAssetCfgRpt
Asset to Asset/Component Configuration Report	W1-AssetToAssetCompCfgRpt
Asset/Component to Asset/Component Configuration Report	W1-AssetCompToAssetCompCfgRpt

Defining Configuration Reports

You use the **Configuration Report** portal to display and maintain configuration reports.

Refer to Understanding Configuration Reports for more information.

You can access the portal by selecting **Admin**, then **Configuration Report**, then **Configuration Report**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration Report**: This zone provides information about the selected configuration report.

Defining Location Options

- Asset Location Types
- Organization Types
- Organizations
- Out of Service Location Types
- Out of Service Locations

Asset Location Types

- Understanding Asset Location Types
- Defining Asset Location Types

Copyright © 2026, Oracle and/or its affiliates.

Understanding Asset Location Types

Asset location types define the details of locations of a certain type where assets or components can be installed. Valid types of asset locations can include service points, other locations for installed assets, or any other location as defined by your organization.

Asset location types are defined by the following:

- **Class**: The general class for asset locations of this type, such as commercial or residential, a shop or warehouse, etc.
- **Location Business Object**: The business object used for asset locations of this type
- **Number of Assets Allowed**: A flag that indicates whether or not asset locations of this type can accommodate a single or multiple assets
- **Service Type**: The type of service (electric, water, etc.) that asset locations of this type support
- **Allow Related Work Locations**: A flag that indicates whether asset locations of this type should be allowed to be connected to a work location
- **Valid Parent Location/Organization Types**: One or more valid parent locations and parent organizations that can be referenced on asset locations of this type. Used primarily for reporting.
- **Valid Communicate-To Location Types**: One or more asset location types to which asset locations of this type can communicate. The **Tree View** tab of any asset or location will display other related assets with a visual indication of their relationship.

> **Note**
>
> Asset location types are used to define locations where assets and components are installed. Locations where assets that are out of service are stored and maintained are defined via Out of Service locations and Out of Service location types.

> **Note**
>
> In Oracle Utilities Customer To Meter, asset location types are maintained via Merged Maintenance.

Defining Asset Location Types

You use the **Asset Location Type** portal is used to display and maintain asset location types.

Refer to Understanding Asset Location Types for more information.

You can access the portal by selecting **Admin**, **Location**, then **Asset Location Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Location Type List**: This zone lists all asset location type records. Broadcast a record to display the details of the selected record.

Copyright © 2026, Oracle and/or its affiliates.

- **Asset Location Type**: This zone provides information about the selected asset location type.

> **Note**
>
> In Oracle Utilities Customer To Meter, asset location types are maintained via Merged Maintenance.

Organization Types

- Understanding Organization Types
- Defining Organization Types

Understanding Organization Types

Organization types define the main attributes for organizations of a certain type.

Organization types are defined by the following:

- **Class**: The general class organization. Parent organizations have a class of “Parent”.
- **Organization Business Object**: The business object used for organizations of this type
- **Valid Parent Organization Types**: One or more valid parent organizations that can be referenced on organizations of this type. Used primarily for reporting.

Defining Organization Types

You use the **Organization Type** portal is used to display and maintain organization types.

Refer to Understanding Organization Types for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Organization Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Organization Type List**: This zone lists all organization type records. Broadcast a record to display the details of the selected record.
- **Organization Type**: This zone provides information about the selected organization type.

Organizations

- Understanding Organizations
- Defining Organizations

Understanding Organizations

Organizations are used for categorization and reporting for locations.

Copyright © 2026, Oracle and/or its affiliates.

For example, your reporting may include a hierarchy that groups locations by a structure of parent and child organizations.

Organizations are defined by the following:

- **Organization Type**: The organization's type. Understanding Organization Types for more information.
- **Organization Disposition**: The current status of the organization (Active or Inactive)
- **Parent Organization**: The parent organization, if applicable.
- **Main Contact**: A contact for the organization

Defining Organizations

You use the **Organization** portal to display and maintain organizations.

Refer to Understanding Organizations for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Organization**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Organization**: This zone provides information about the organization.
- **Child Locations / Organizations**: This zone lists any child locations and/or organizations for the current organization.

Out of Service Location Types

- Understanding Out of Service Location Types
- Defining Out of Service Location Types

Understanding Out of Service Location Types

Out of service location types define attributes of types of locations where assets that are out of service are stored and maintained. Examples of out of service locations include warehouses, storerooms, and repair shops.

Out of service location types are defined by the following:

- **Class**: The general class (storeroom, meter shop, etc.) of out of service locations of this type
- **Location Business Object**: The business object used for out of service locations of this type
- **Valid Parent Location/Organization Types**: One or more valid parent locations and parent organizations that can be referenced on out of service locations of this type. Used primarily for reporting.

Defining Out of Service Location Types

Copyright © 2026, Oracle and/or its affiliates.

You use the **Out of Service Type** portal is used to display and maintain out of service location types.

Refer to Understanding Out of Service Location Types for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Out of Service Location Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Out of Service Location Type List**: This zone lists all out of service location type records. Broadcast a record to display the details of the selected record.
- **Out of Service Location Type**: This zone provides information about the selected out of service location type.

Out of Service Locations

- Understanding Out of Service Locations
- Defining Out of Service Locations

Understanding Out of Service Locations

Out of service locations are physical locations where assets are stored or repaired such as a storeroom, warehouse, or service facility. When assets are changed to statuses such as “In Repair” or “Pending Disposition” the asset location is typically moved to an out of service location.

Out of service locations are defined by the following

- **Location Type**: The out of service location’s type. See Understanding Out of Service Location Types for more information.
- **Location Disposition**: The current status of the out of service location (Active or Inactive)
- **Parent Location/Organization**: The out of service locations’ parent location or organization (if applicable)
- **External ID**: An ID used to identify the out of service location in external systems
- **Address Information**: The physical address of the out of service location
- **Contacts**: The primary contact for the out of service location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with out of service locations.
- **Field Information**: Additional information about the physical location, such as GPS coordinates

Defining Out of Service Locations

You use the **Out of Service Location** portal to display and maintain out of service locations.

Refer to Understanding Out of Service Locations for more information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, then **Location**, then **Out of Service Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Out of Service Location**: This zone provides information about the out of service location.
- **Current Activities**: This zone provides a summary of the activities that are currently associated to the location.
- **Asset Summary**: This zone provided statistics about the number of assets, by asset type, that are currently associated with the location.
- **Owned Attachments**: This zone lists attachments owned by the location.
- **Referenced Attachments**: This zone lists attachments related to the location.

> **Note**
>
> See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with out of service locations.

Defining Device Options

- Command Sets
- Head End Systems
- Measuring Component Types
- Measuring Component Comparison Types
- Device Configuration Types
- Device Types

Command Sets

- Understanding Command Sets
- Configuring Command Sets

Understanding Command Sets

Command sets are used to define a group of commands that are not eligible for a particular device. For instance, if Commission or Decommission commands should be considered ineligible for a particular device model, a command set that references the Device Commission and Device Commission business objects could be created and associated with that device model.

Command sets are specified for individual device models via the Manufacturer portal.

Individual devices of a particular model can be configured to override ineligibility if needed.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Command Sets

This portal is used to display and maintain a Command Set.

Refer to Understanding Command Sets for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Command Set**.

The following zones may appear as part of the portal's **Main** tab page:

- **Command Set List:** This zone lists all Command Set records. Broadcast a record to display the details of the selected record.
- **Command Set:** This zone provides information about the selected Command Set.

Head End Systems

- Understanding Head End Systems
- Understanding Processing Methods
- Understanding SGG Adapter Configuration
- Configuring Head End Systems

Understanding Head End Systems

Head end systems are systems that collect measurement data and meter events for eventual submission to the application. Many devices can communicate to the application through a single head-end system, but a utility may have numerous head-end systems through which they communicate with devices.

Head end systems utilize processing methods that specify the type of initial measurement data and device events to create for devices (and their related measuring components) based on measuring component type. Head end systems also utilize processing methods that specify how smart meter commands are processed.

Refer to Understanding Processing Methods for more information about processing methods.

Head End Systems Impact Data Import and Export

Head end systems are configured to identify how a particular external system communicates data with Oracle Utilities Meter Data Management. This includes:

- The identifier type used to locate devices and measuring components. These are used both on import and export of data.
- The date/time format used in various data imports (i.e. whether or not the date/time format includes time zone information).

Please refer to the embedded help for more information about these fields.

Each head end system can be associated to an external system which is used to define the messages that can be sent to that service provider and how each message is sent.

Understanding Processing Methods

Copyright © 2026, Oracle and/or its affiliates.

Head end systems, external applications, and market participants can have one or more associated processing methods that define the format or means by which it receives or sends data from or to the application, such as bill determinants, interval data, or meter events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and usage transactions. Processing methods can also be used to define the information an external system wishes to subscribe to receive from our application.

At the lowest level processing methods are used to identify an outbound message type, business object, batch control, or message category and number.

Each processing method is comprised of a business object that defines what is being mapped as well as how it should be mapped.

Important Processing Method System Events

The actual logic to determine the appropriate output for a given head end system, external application, or market participant and processing role for a processing method is executed by the following system event:

- **Determine Processing Method(s)**: is a business object algorithm system event that takes in a head end system, external application, or market participant, a processing role, and a list of relevant input data aka related object (e.g. a measuring component, device, etc). For the head end system or external application and the input data it will analyze the selection criteria to determine the appropriate output.

How Processing Methods Work

Processing methods perform two basic tasks:

- They define the criteria for selecting the appropriate output. This can be as simple as providing a single object in return without qualification or in more involved situations it could support determining the appropriate return object based on characteristics of the data being processed. This is accomplished through the data structure defined on the processing method business object.
- They evaluate the criteria for selecting the appropriate output given a specific head end system, external application, or market participant. This is accomplished through the Determine Processing Method system event on the processing method business object.

Each processing method business object can be associated to one or more processing roles. This is done by adding the Applicable Processing Role business object option. It is these processing roles that actually create the association between a head end system, external application, or market participant, the processing method, and the functional event that is being executed. For example, when initial measurement data is processed through the IMD Seeder the processing role Initial Measurement Creation is used to identify the processing method "How to Create MC Related Information" which maps a particular measuring component type to the appropriate initial measurement business object to be used for processing.

When system logic requires the results of a processing method the service Determine Service Providers and Methods is used. This service is available to call via Java or from within scripting through the business service of the same name. It can be called in one of two ways:

With a head end system, external application, or market participant: identifies the appropriate output for a single head end system, external application, or market participant being processed for the input processing role and related objects

Without a head end system, external application, or market participant: used to identify subscribing sysems. This will provide a list of any head end system, external application, or

Copyright © 2026, Oracle and/or its affiliates.

market participant that has the input processing role and an appropriate output given the related objects.

Processing Methods Available

There are the following processing methods provided by the base package:

Name	Details	Business Object
How to Create OB COMM/Send OB Message	Identifies Message Number/ Category, Business Object, Outbound Message Type and allows for an override by a device type.	D1-HowToCreateActivityOBComm
How to Create MC Related Information	Identifies a business object and allows for an override by measuring component type.	D1-HowToCreateMCInformation
How to Process Device Event Related Info	Identifies a business object, outbound message type, and batch control by device event category allowing for an override by device event type.	D1-HowToProcDvcEvtsInformation
How to Process Device Related Information	Identifies a business object and allows for an override by device type.	D1-HowToProcessDeviceInfo
How to Send Activity Related Information	Identifies a batch control and business object by activity type and allows for an override by device type.	D1-HowToSendActInformation
How to Send Activity Related O\B Messages	Identifies an outbound message type, message category, and message number and allows for override by activity type.	D1-HowToSendActivityResponse
How To Create US Related Information	Identifies a business object and allows for override by usage subscription type.	D2-HowToCreateUSInformation
How To Send US Related Information	Identifies a batch control, business object, and outbound message type and allows for override by usage subscription type.	D2-HowToSendUSInformation
How to Process Service Point Related Info	Identifies a business object and allows for override by service point type.	D1-HowToProcSPRelatedInfo
How to Send Field Activity Related Info	Identifies a outbound message type and allows for an override by field task type.	D1-HowToSendFARelatedInfo
How to Send Field Activity Remark Info	Identifies an outbound message type for an activity remark type.	D1-HowToSendActivityRemarkInfo
How to Translate External Value	Identifies a business object and allows for override by identifier.	D1-HowToTranslateExternalValue
How to Request Customer Notification	Identifies a list of outbound message types.	D1-HowToRequestCustomerNotific

Copyright © 2026, Oracle and/or its affiliates.

How to Process Business Flag Related Info	Identifies an outbound message type and allows for an override by business flag type.	D1-HowToProcessBusinessFlagInf

Understanding SGG Adapter Configuration

This section describes how to use a head end system's SGG Adapter Configuration portal.

You can use the SGG Adapter Configuration portal to view configuration information and access configuration components for an SGG adapter head end system.

Note: This portal displays configuration information for head-end systems that reference an SGG Adapter Configuration Sheet extendable lookup..

To use the configuration information portal for an SGG adapter head end system:

Select **Admin > Device > Head End System**.

In the Head End system List zone, click the Broadcast icon for the head-end system you wish to view.

Click the SGG Adapter Configuration tab to view the configuration information.

The SGG Adapter Configuration portal contains the following zones:

- **SGG Adapter Configuration Tracker**: This zone displays the configuration details of the adapter, as defined by the SGG Adapter Configuration Sheet extendable lookup referenced on the head end system. The configuration details include:

The components required for usage and event processing and command processing. To view more details about the components, you can click the component name to go to the business object for the component. For example, you can click the business object "SSN - Connect or Disconnect" to go to the business object portal for the SSN - Connect or Disconnect business object.

Status messages describing the configuration status of components. The following table lists the status messages that may be displayed and the possible actions you can take:

If the status message is...	You can...
Set up this processing method	Click the processing role to set up the processing method.
This processing method has been configured	Click the processing role to view the configured processing method.
Update your processing method with a communication BO	Click the status message to set up the processing method.
Update your external system / outbound message type with an Message sender	Click the status message to go to the external system.
Update your processing method with an outbound message type	Click the status message to set up the processing method.
Add a value to get started	Click the status message to go to the extendable lookup.
Values Existing: (number)	Click the status message to go to the extendable lookup.
Add a communication type	Click the status message to go to the communication type portal.

Copyright © 2026, Oracle and/or its affiliates.

Add a device event type	Click the status message to go to the device event type business object for the communication type. This message appears only for Echelon type adapters.
Master Configuration has been added	Click the status message to view the Master Configuration portal.
Add Master Configuration for this adapter	Click the status message to view the Master Configuration portal.

Upload Statistics Aggregators: This zone lists the IMD Upload Statistics Aggregator measuring components associated with the head-end system.

Configuring Head End Systems

This portal is used to display and maintain Head End Systems.

Refer to Understanding Head-End Systems for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Head End System**.

The following zones may appear as part of the portal's **Main** tab page:

- **Head End System List:** This zone lists all Head End System records. Broadcast a record to display the details of the selected record.
- **Head End System:** This zone provides information about the selected Head End System.
- **Processing Method List:** This zone provides the list of processing methods defined for the Head End System.
- **Translation Methods List:** This zone provides the list of translation methods defined for the Head End System.
- **Inbound BOs Send By Service Provider:** This zone lists inbound Business Objects that are sent by this Head End System. The identification is driven by the Business Object having a Business Object Option of type "Sent By Service Provider" that references the current Head End System.

Measuring Component Types

- Understanding Measuring Component Types
- Configuring Measuring Component Types

Understanding Measuring Component Types

Measuring component types define the most important properties of a measuring component.

Measuring component types define what a measuring component measures (KWH, temperature, etc.), how regularly it measures it, and whether it should be connected to a physical device, or if it's used as a scratchpad measuring component or an aggregator measuring component. Measuring component types also specify how the measuring component's final measurements should be stored, how the measuring component's user-defined values should be calculated, and specific rules governing validation, editing, and estimation (VEE) for measuring components of the type. In addition, measuring component

Copyright © 2026, Oracle and/or its affiliates.

types define display properties and valid attribute values for measuring components belonging to the type.

The following configurable items are available for most measuring component types:

- **Anchor Time**: A time used to define the start and end time of the business day for purposes of processing measurement data and usage transactions for measuring components of this type. Applicable only to dynamic aggregation measuring component types, standalone measuring component types, and profile measuring component types.
- **Initial Load IMD Ordering (Fallback)**: A flag that indicates if incoming initial measurements for measuring components of this type should be processed using Measuring Component Ordering. This is used most often in the case of daily scalar measuring components for which data is occasionally sent to the system out of order, but can also be used to ensure that "check channels" are processed before related channels. Applicable only to measuring component types used with Direct Measurement Processing (see Direct Measurement Processing and Measuring Component Ordering for more information).
- **Value Identifiers**: These store the values of UOM, TOU, and SQI that identify the measured amounts for measuring components of this type. Value identifiers specify the quantities stored on the measurement records for measuring components of this type. Please refer to the Measuring Component Type Value Identifiers topic later in this section for more information.
- **Valid VEE Groups**: These define the VEE groups considered valid for measuring components of this type. Each group supplied here will be available to be selected on measuring components of this type and act as an override to the Fallback VEE Groups.
- **Fallback VEE Groups**: These define default VEE groups for a specific VEE Group Role that can be used with all measuring components of this type. This alleviates the need to specify the same VEE groups on multiple measuring components of the same type. Changes made to these groups will automatically apply to all measuring components of this type unless they have specified their own VEE groups for that particular VEE Group Role. Each VEE group is designated a VEE group role that indicates when and how the VEE group is used (for initial load, manual override, estimation, etc.).
- **Enable ML Based Validation (Fallback)**: A flag that indicate (Yes or No) if initial measurements for measuring components of this type will be processed through a machine learning scoring model. IF set to "Yes", VEE processing for initial load data will only be used when data fails machine learning normalcy checks. See Machine Learning Anomaly Scoring for more information.
- **Anomaly Score Threshold (Fallback)**: The anomaly score threshold for measuring components of this type. This overrides the value set in the MDM Master Configuration. This field is only visible when the **Enable ML based Validation (Fallback)** flag is set to "Yes". See Machine Learning Anomaly Scoring for more information.
- **Eligible Profile Factors (interval only)**: These define the profile factors that are considered to be eligible for interval measuring components of this type. One profile factor can be identified as the default. The default profile factor will be automatically selected in system processing when a profile factor is required.
- **Valid Profile Factors for Conversion from Scalar to Interval (scalar only)**: These define the profile factors that are considered to be eligible for scalar measuring components of this type when converting scalar measurements to interval measurements. These profile factors are used to produce a curve of interval data from a scalar value. Without one of these factors defined scalar to interval conversion will use a flat line method (i.e. evenly divide the scalar value across the intervals). One profile factor can be identified as the default. The default profile factor will be automatically selected in system processing when a conversion profile factor is required.

Copyright © 2026, Oracle and/or its affiliates.

- **Valid Scratchpad Measuring Component Types**: These define the scratchpad measuring component types considered valid for measuring components of this type.
- **Related Statistics Measuring Component Types**: These define the measuring component types that will be used to store statistical information about the historical usage of measuring components of this type. Please refer to Configuring Measuring Component Statistics for more information on how this list is used.
- **Display Properties**: Defines how measurement data for measuring components of this type is displayed, including:
  - *Display Configuration*: Details related to how measurements are displayed, including the 360 chart rendering method, number of hours of data to display, the maximum days to search for measurements, the default TOU map used, the TOU by Day Profile factor used, and default measurement condition.
  - *Event Bar Profiles*: The event bar profiles used when displaying measurement data for measuring components of this type. Event bar profiles are defined as values for the 360 View Event Bar Profile extendable lookup.
  - *Final Values Overlay Profiles*: The final values overlay profiles used when displaying measurement data for measuring components of this type. Final values overlay profiles are defined as values for the Final Values Overlay Profile extendable lookup.
  - *Measurement Conditions Not Shown on Chart*: The measurement conditions that should be omitted from rendering onto 360 Degree charts. Measurements whose conditions match these values will be rendered as gaps. For example, many 360 Degree charts use the condition "No Read - System" to represent the lack of a measurement, by adding this condition to this list it a gap will be rendered instead of a line with a 0 quantity measurement and a condition of "No Read - System".

When creating a measuring component type the following options are available:

Name	Details	Business Object
Interval Channel Type - Physical	Provides the configuration for a physical interval channel (including interval size). This is recommended for measuring components that measure consumptive usage in interval sizes up to 1 hour.	D1-IntervalChannelTypePhysical
Interval Channel Type - Scratchpad	Provides the configuration for a scratchpad interval measuring component.	D1-IntervalChannelTypeScratchp
Interval Channel Type - Physical Subtractive	Provides the configuration for a physical subtractive interval channel. In addition to standard interval configuration (including interval size) it also provides additional subtractive specific configuration (e.g. rollover validation, estimate reevaluation, etc). This is recommended for measuring components that measure subtractive usage in interval sizes up to 1 hour.	D1-IntrvlChanTypPhysSubtractiv

Copyright © 2026, Oracle and/or its affiliates.

Interval Channel Type - External	Provide the configuration for interval measuring components that measure data not typically stored in the system, such as Amps. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist. Filtering of data for external measuring components is configured via the **IMD Handling** flag.	D1-IntervalChannelTypeExternal
Register Type	Provides the configuration for a physical register that is manually read (e.g. rollover validation). These can be either consumptive or subtractive but are expected to be read infrequently (e.g. once a month).	D1-RegisterTypePhysical
Auto-Read Register Type	Provides the configuration for a physical register that is automatically read. In addition to the standard register configuration (e.g. rollover validation) it also provides details around the schedule of expected readings (e.g. first daily measurement time and expected hours between measurements). These can either be consumptive or subtractive but are expected to be read frequently (e.g. at least once per day). Note: these measuring component types can also be used to model interval data that is received with a large interval size (24 hours and larger).	D1-AutoReadRegisterType

Copyright © 2026, Oracle and/or its affiliates.

Scalar Channel Type - External	Provide the configuration for scalar measuring components that measure data not typically stored in the system. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist. Filtering of data for external measuring components is configured via the **IMD Handling** flag.	D1-ScalarChannelTypeExternal
Aggregator Type	Provides the configuration for an aggregation measurement.	D2-AggregatorType

> **Note**
>
> For “interval” data with an interval size greater than one hour, use the D1-AutoRegisterReadType measuring component type business object and the D1-RegisterAutoRead measuring component business object. This business object is well suited to daily and sub-daily readings. Using an interval data business object can result in measurement date problems during daylights saving time.

Refer to Configuring an Out-of-the-box Aggregation for more information about aggregation configuration

Measuring Component Type Value Identifiers

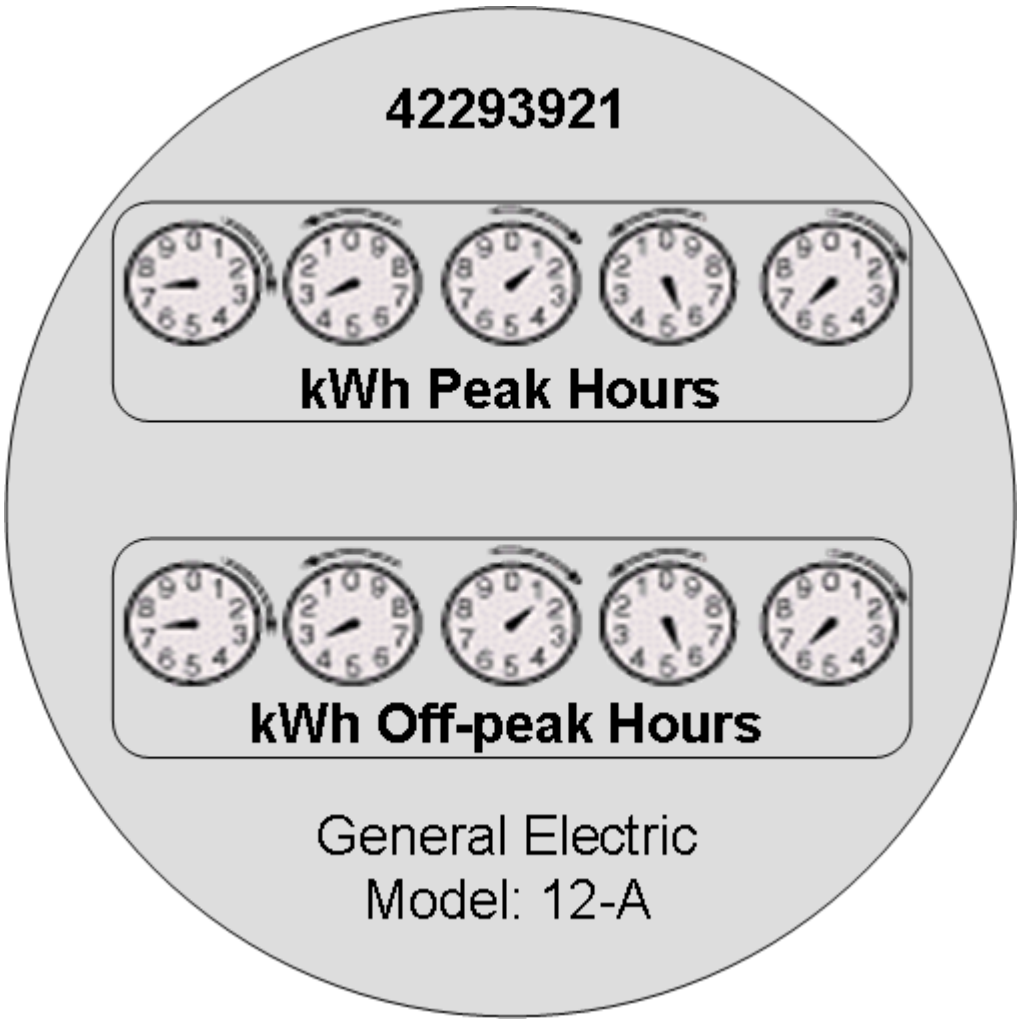
Measuring components are configured to measure specific types of quantities this is defined by the list of value identifiers on the measuring component type. Up to eleven value identifiers can be specified. The primary measured quantity should be identified using the Value Identifier Type of Measurement. An additional ten derived values can be computed based on the primary measurement, these are identified with the Value Identifier Type of Value 01 through Value 10. Each value identifier is constructed of:

- **Unit of Measure**: The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc. Refer to Understanding Units of Measure for more information.
- **Time of Use**: Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc. Refer to Understanding Time of Use for more information.
- **Service Quantity Identifiers**: Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components measure the same thing, but in different ways. A meter that measures both generation KWH and consumption KWH could use SQIs to differentiate between the two. Refer to Understanding Service Quantity Identifiers for more information.

Copyright © 2026, Oracle and/or its affiliates.

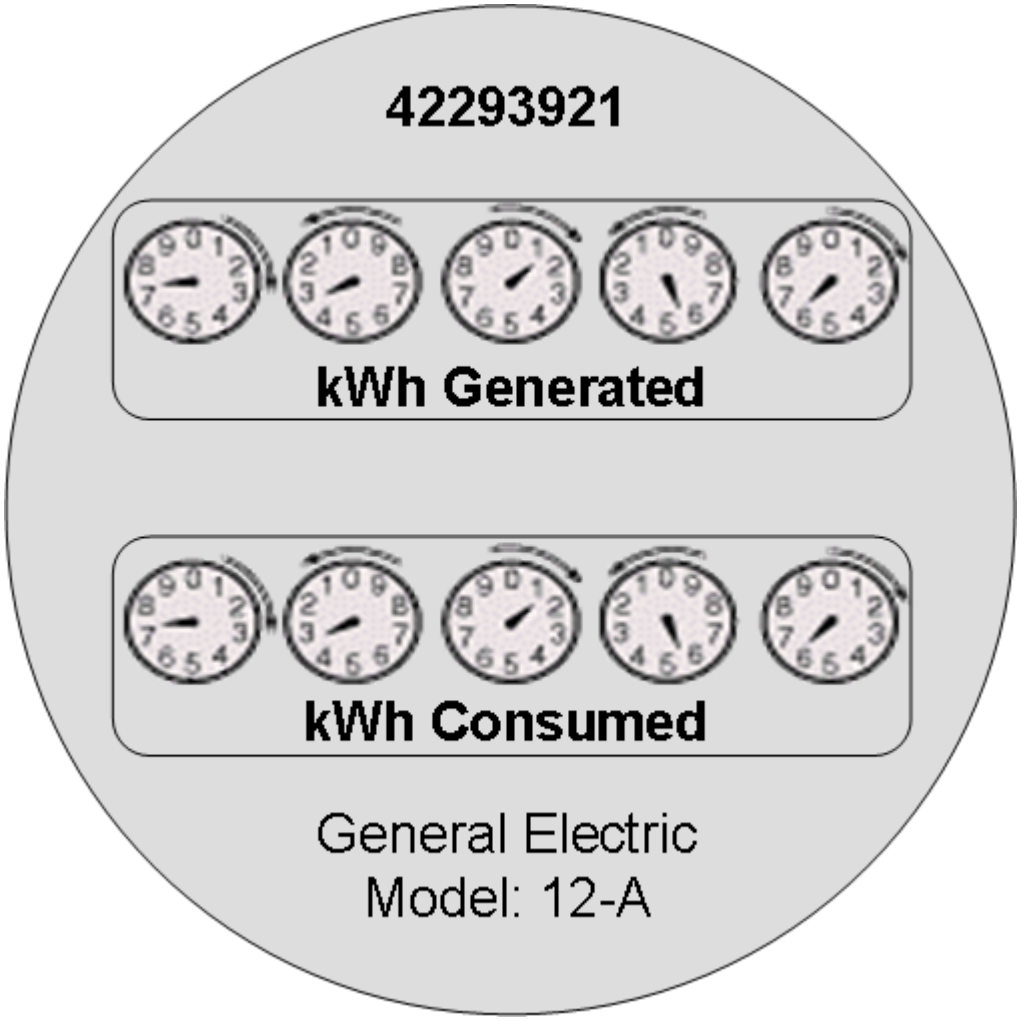
- **Value Derivation Algorithm**: Unlike UOM, TOU, and SQI this is not used in the identification of what is measured but rather is used to calculate a derived value based on the primary measurement. An algorithm from the list should be selected that contains the appropriate logic for calculating the derived value. This is applicable for those value identifiers with a Value Identifier Type of Value 01 through Value 10. More information on how derived values are calculated can be found in the Important Measuring Component Type System Events topic in this chapter. For more functional information about derived values please refer to About Final Measurements.

The combination of UOM, TOU and SQI define what a measuring component measures. TOU and SQI are optional, but UOM must be defined for all value identifiers. For example, consider a meter (as illustrated in the image below) with two measuring components, both measuring the same unit of measure (kWh), but each measuring component measures consumption in different time of use (TOU) periods (peak and off-peak).



Another example might be a meter that records both generated KWH and consumed KWH. This meter would be configured to measure both UOM and SQI.

Copyright © 2026, Oracle and/or its affiliates.



A measurement is recorded each time a measuring component is read. This means that for a meter with two measuring components that is read once a month, two measurements, one for each measuring component, would be recorded each month.

Important Measuring Component Type System Events

The measuring component type supports several business object algorithm system events that relate to calculating the consumption for measuring components of that type:

- **Calculate Interval Consumption**: receives the interval list and performs any necessary calculations on that interval data to compute consumption. Since interval data is already received as consumption data algorithms for this system event are typically limited to application of the appropriate multipliers. Refer to the algorithm type Calculate Interval Consumption (D1-IN-CNSUMP) as an example.
- **Calculate Scalar Consumption**: receives information about the scalar reading and calculates the consumption as appropriate. Algorithms for this event typically support calculating the consumption using a stop and start reading or backing into a reading using consumption and a start reading. Much like the interval counterpart it will apply the appropriate multipliers. Refer to algorithm type Calculate Scalar Consumption (D1-SC-CNSUMP) as an example.
- **Calculate Subtractive Interval Consumption**: receives the interval list and supporting information (e.g. the start reading for the first interval) and either calculates the consumption by subtracting the interval's reading from the prior interval's reading or

Copyright © 2026, Oracle and/or its affiliates.

calculates the reading by adding the current interval's consumption to the prior interval's reading. Refer to algorithm type Calculate Subtractive Interval Consumption (D1-SIN-CNSUM) as an example.

- **Condition Mapping**: receives the details of a single subtractive interval along with the details for its start reading and computes the applicable final condition and reading condition. This is leveraged solely for subtractive interval measuring component types. Refer to algorithm type Subtractive Interval - Condition Mapping (D1-SIN-CNMAP) as an example.

These system events are typically called from within initial measurement processing during the initial stages of the initial measurement lifecycle (e.g. the Pre VEE status of most initial measurement business objects).

In addition to the measuring component type business object algorithms there is an additional system event provided on measuring component type itself:

- **Value Derivation**: receives details of an initial measurement and an associated final measurement. Using these inputs it can compute a value derived either from the primary measurement or one of the other derived values. Refer to algorithm type Derive a quantity using a formula (D1-DERIVAQTY) as an example.

Important Measuring Component System Events

The measuring component business object that is associated to a given measuring component type supports a special system event that is used in the periodic estimation process:

- **Periodic Estimation**: this system event will scan the measuring component's final measurement history to identify missing measurements and create either a To Do, or an estimation initial measurement, or both. More detail about this system event can be found by visiting the following algorithm types: Refer to algorithm type Create Interval IMD and To Do Based Upon Install History (D1-CIITBIH) and Auto-Read Scalar Periodic Estimation (D1-ARSPE) as an example.

Measuring Component Business Object Options

The device business object that is associated to a given device type plays an important role in how a device is processed in a system beyond defining the data associated to that device. Below are a list of business object options that are defined on the device business object and their impact on system processing:

- **Estimation Initial Measurement Data BO**: Is used to identify the appropriate estimation IMD to create.
- **Manual Override IMD BO**: Is used to identify the appropriate manual override IMD to create.
- **System IMD BO**: Is used to identify the appropriate system IMD to create.
- **Measuring Component Consumption Function**: identifies a function that can be executed from any compatible measuring component 360 Degree zone. Each measuring component can support 0 to many of these functions.
- **Interval Initial Measurement Function**: identifies a function that can be executed from any compatible initial measurement zone. Each measuring component can support 0 to many of these functions.

More detail about these options can be found by visiting a measuring component business object and inspecting the business object options.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Measuring Component Types

This portal is used to display and maintain a Measuring Component.

Refer to Understanding Measuring Component Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Measuring Component**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Type:** This zone provides information about the selected Measuring Component Type.

Measuring Component Comparison Types

- Understanding Measuring Component Comparison Types
- Configuring Measuring Component Comparison Types

Understanding Measuring Component Comparison Types

Measuring component comparison types define the details by which measuring component data can be compared to determine the days that most closely resemble a specific day being evaluated. For example, measuring component comparison types can be used on measuring components that store weather data so that comparisons can be made based on temperature. These are used when performing “proxy day” estimations, allowing the application to search for days that are similar to historical data (for instance, the same date from the previous month or previous year) to use as the basis for estimation.

Measuring component comparison types use the following parameters:

- A specific metric to be calculated to enable data comparisons. The “Calculate Comparison Coefficient - Pearson Correlation Coefficient” algorithm, included with the base package can be used to calculate a coefficient based on the linear correlation between two data sets (specifically between weather data stored on a measuring component and on the comparison object).
- The specific object to be compared. This can be either the same measuring component (to compare against historical data for the same measuring component), or a factor profile (to compare against profile data).
- The date range for the comparison period, including the start of the comparison period (current date, same date last year, or both), the time each day through which the comparison will be calculated, and the number of days prior to and after the start date that will be compared with the comparison object.
- Average Difference Eligibility: An average value either above or below the average for the current date that will be eligible for comparison. For example, assume the current date’s average is 75, and the prior day’s average is 72. If the Average Difference Eligibility is set to 3 then the prior day value of 72 will be eligible for comparison to the current day average of 75.
- The maximum number of resulting calculations that should be stored as measuring component comparable periods.

Copyright © 2026, Oracle and/or its affiliates.

- A template that defines which days are considered like each day of the week (for example, weekdays are like other weekdays, weekend days are like other weekend days). This template is defined as a record for the “Like Day Eligibility Template” extendable lookup.
- Work calendars used to determine if specific data should be compared to other weekend days or to the same date in prior years, and which dates, if any, should be excluded from historical data used for comparison.

Configuring Measuring Component Comparison Types

Refer to Understanding Measuring Component Comparison Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Measuring Component Comparison Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Comparison Type List**: This zone lists all measuring component comparison type records. Broadcast a record to display the details of the selected record.
- **Measuring Component Comparison Type**: This zone displays details for the selected measuring component comparison type.

Device Configuration Types

- Understanding Device Configuration Types
- Configuring Device Configuration Types

Understanding Device Configuration Types

Device Configuration types define the measurements a given device can record. It should be noted that the device configuration type itself does not specify this information but rather the device configuration type will identify the valid list of measuring component types that can be referenced associated to a device.

When creating a new device configuration type there are the following options:

Name	Details	Business Object
Device Configuration Type	This represents configurations for physical meters and communication components. It identifies a list of valid measuring components types.	D1-DeviceConfigurationType
Item Configuration Type	This represents badged items that are installed at a particular service point. Given the nature of items there are no measuring component types associated to this type.	D1-ItemConfigurationType

Refer to Configuring Consumption Synchronization for more information on how a device configuration type can be set up to synchronize consumption between two related channels of measurement data.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Device Configuration Types

This portal is used to display and maintain a Device Configuration Type.

Refer to Understanding Device Configuration Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Device Configuration Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Configuration Type List:** This zone lists all Device Configuration Type records. Broadcast a record to display the details of the selected record.
- **Device Configuration Type:** This zone provides information about the selected Device Configuration Type.

Device Types

- Understanding Device Types
- Configuring Device Types

Understanding Device Types

Device types define information about a class of devices, including properties that apply to all devices of a type. Properties defined for a device type can be overridden for an individual device

Specifically, device types provide information about:

- How a device can be configured by defining the valid list of device configuration types
- How a device communicates by defining a fallback head end system and a list of valid head end systems.
- How a device records measurement data through the fallback incoming data shift which plays an important role in daylight savings processing. Refer to Daylight Savings Time Support for more information.
- The consumption profile of an item: the unit of measure consumed and method of calculation (profile or straight line)
- Whether the device is attached to a service point or meter (used with items and device communication equipment)
- Characteristics common to all devices of this type (defined in the **Device Type Characteristics** zone).

When creating a new device type there are the following options:

Name	Details	Business Object

Copyright © 2026, Oracle and/or its affiliates.

Smart Meter Device Type	These devices measure consumption for a given service point, support remote commands, and remote collection of measurement and event data. These devices are associated to a head end system.	D1-SmartMeterType
Manual Meter Device Type	These devices measure consumption for a given service point but require a visit to the service point to collect the measurement data. They are not associated to head end systems.	D1-ManualMeterType
Item Device Type	These devices represent various items that consume consumption at a constant rate which are not directly metered. As consumption is calculated rather than measured so there is no need for an association to a head end system. Devices of this type can either be created individually for "badged" items or can be listed directly on a service point for "unbadged items." These devices may also be configured as equipment that can be attached to a service point.	D1-ItemType
Communications Component Device Type	These devices provide remote collection of consumption data measured by a manual meter on which they are installed. These devices are associated to a head end system.	D1-CommunicationCompMeterType
Device Communication Equipment Type	These devices provide remote collection of consumption data measured by a manual meter to which they are attached. Devices of this sort are sometimes referred to as Encoding Receiver/Transmitter (ERT) modules, or communication modules. These devices are associated to a head end system.	D1-DvcCommEquipmentType

Device Business Object Options Drive Functionality

The device business object that is associated to a given device type plays an important role in how a device is processed in a system beyond defining the data associated to that device. Below is a list of business object options that are defined on the device business object and their impact on system processing:

Copyright © 2026, Oracle and/or its affiliates.

- **Device Category**: This defines the type of device and correlates to the core device types that are supported (i.e. smart meter, AMR meter, manual meter, item, or communication component). This value is used by smart meter commands, service order management, and other system processes to make processing decisions.
- **Install Event BO**: This identifies the appropriate install event business object to use when this device is being installed at a service point. Refer to About Install Events for more information about install events.
- **Valid Command Request BO**: This identifies the commands that are valid for any device of this type. This option can be repeated for as many commands as the device supports. Note: the combination of the device and service provider (aka head end) define the true list of available commands by device.
- More detail about these options can be found by visiting a device business object and inspecting the business object options.

Device Type - Service Quantity

For "badged" and "unbadged" items consumption is not directly measured. Instead, for each item type, the average daily consumption amount is provided. The commodity represented by the daily consumption amount is defined by the item type's unit of measure.

The average daily consumption amount can be effective dated over time to support changes in the consumption profile. This is used in conjunction with the item's calculation method to derive the consumed amount per interval.

Refer to Multiple Time Zone Support for additional information on how device type service quantities are impacted by multiple time zones.

Valid Device Characteristic Types

The **Valid Device Characteristic Types** zone lists the valid characteristic types that may be instantiated on devices of this type. Characteristics of these types are maintained using the **Device Characteristics** zone in the **Device** portal.

Configuring Device Types

This portal is used to display and maintain a Device Type.

Refer to Understanding Device Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Device Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Type List:** This zone lists all Device Type records. Broadcast a record to display the details of the selected record.
- **Device Type:** This zone provides information about the selected Device Type.
- **Device Type Service Quantity:** This zone displays the list of effective dated service quantities supported by the selected Device Type. This applies only to Items.
- **Device Type Characteristics**: This zone lists characteristics common to all devices of this type.
- **Valid Device Characteristic Types**: This zone lists the valid characteristic types that may be instantiated on devices of the device type. To add or edit valid characteristic types, click **Edit**.

Copyright © 2026, Oracle and/or its affiliates.

Defining Device Installation Options

- Service Point Types
- Service Point Quantity Types
- Divisions
- Entity Groups
- Weather Stations
- Postal Code Defaults
- Measurement Cycles
- Measurement Cycle Processing
- Measurement Cycle Schedules
- Setting Up Metered Premise Options
- Configuring Cross Installed Meter Processing
- Setting Up Consumption Estimation Parameters

Service Point Types

- Understanding Service Point Types
- Configuring Service Point Types

Understanding Service Point Types

Service point types define a specific type of point at which service is delivered.

Specifically, service point types define how the application manages many aspects of the service point's behavior. A service point type may have one or more valid device types defined that limit the types of devices that can be installed at service points of this type.

Service point types are defined by the following:

- **Service Point Business Object**: the business object used by service points of this type
- **Service Point Category**: the types of devices that can be installed at service points of this type. Valid values include:
  - **Meter**: Indicates that a single meter can be installed at service points of this type.
  - **Item**: Indicates that a single "badged" item can be installed at service points of this type.
  - **Multi-Item**: Indicates that one or more "unbadged" items can be installed at service points of this type.
- **Divisions**: one or more divisions associated with service points of this type.
- **Valid Device Types**: the device types considered valid for service points of this type.
- **Valid Field Task Types**: the field task types considered valid for service points of this type. The field task types available when creating field activities can be based on the field task

Copyright © 2026, Oracle and/or its affiliates.

types listed here via the **Restrict Field Task Type by Service Point Type** option on the Service Order Management Master Configuration.

- **Usage Snapshot Configuration**: the configuration(s) to be used to take weekly or monthly usage snapshot(s)
- **Unreported Usage Snapshot Configuration**: the configuration(s) to be used to take weekly or monthly unreported usage snapshot(s)

Configuring Service Point Types

You use the Service Point Type portal to view and maintain service point types.

Refer to Understanding Service Point Types for more information.

You can access the portal by selecting **Admin**, **Geographic**, **Service Point Type**.

> **Note**
>
> If the **Control By CIS Division** setting on the **Main** tab on the **Installation Options** portal is set to "Yes", this zone displays only unsecured service point types as well service point types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Type List:** This zone lists all service point type records. Broadcast a record to display the details of the selected record.
- **Service Point Type:** This zone provides information about the selected service point type.
- **Valid SP Characteristic Types**: This zone lists the valid characteristic types that may be instantiated on service points of the service point type. To add or edit valid characteristic types, click **Edit**.

  > **Note**
  >
  > Only characteristics types defined for both the “Service Point — MDM” **and** “Service Point — CCB” characteristic entities can be selected in this zone.

- **Valid Field Task Types**: This zone lists the field task types considered valid for the selected service point type. To add or edit valid field task types, click **Edit**.
- **Valid SA Types**: This zones lists the service agreement types considered valid for the selected service point type. To add valid service agreement types, click **Add** in the zone title bar. To edit, delete, or copy a listed service agreement type, select the service agreement type from the list and click **Edit**, **Delete**, or **Copy** as appropriate.
- **Valid Disconnect Location**: This zone lists disconnection locations considered valid for the selected service point type.

> **Note**
>
> In Oracle Utilities Customer To Meter, Service Point Types are maintained via Merged Maintenance.

Copyright © 2026, Oracle and/or its affiliates.

Service Point Quantity Types

- Understanding Service Point Quantity Types
- Configuring Service Point Quantity Types

Understanding Service Point Quantity Types

Service point quantity types define types of quantities that can be stored for a service point. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class. For example, a customer quantity may be used to store a scaling factor that describes how a given customer's usage compares to the profiled usage for their rate class.

In a settlement implementation, service point quantity types can be used to define values calculated monthly, annually, etc., such as annual peak load contribution (PLC).

Service quantity types use the following parameters:

- The service type (electric, gas, water, etc.) for service point quantities of this type
- Quantity identifiers related to the current service point quantity type (used to provide shorthand descriptions of the various types of values measured by service point quantities of this type)

Configuring Service Point Quantity Types

Refer to Understanding Service Point Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Service Point Quantity Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Quantity Type List**: This zone lists all service point quantity type records. Broadcast a record to display the details of the selected record.
- **Service Point Quantity Type**: This zone displays details for the selected service point quantity type.

Divisions

- Understanding Divisions

Understanding Divisions

Divisions are used to delineate between different operating companies within a large conglomerate of utilities.

Divisions can have one or more characteristics defined.

Divisions can be defined on several other objects and entities in the system, including:

Copyright © 2026, Oracle and/or its affiliates.

- Service Point
- Service Order Activities (Request Orchestration activities)
- Service Order Field Activities
- Usage Subscription/Usage Subscription Type
- Usage Calculation Group
- Settlement Calculation Group
- Aggregation Group
- Data Source
- Market Product Type
- Market Product Set

Divisions can be associated with individual users on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

Entity Groups

- Understanding Entity Groups
- Entity Groups

Understanding Entity Groups

Entity Groups are used to define groups of entities, such as devices (including meters or items, as well as controllable devices used with Customer Program Management Cloud Service and Customer Program Management) or service points.

An entity can belong to more than one group if they have different Group Categories. For example, a controllable device associated with a specific program and rate could belong to a group category based on the specific program as well as a group category based on the specific rate.

Entity group processing uses dynamic SQL queries to identify entities that belong to a entity group. These dynamic queries are based on configuration of the Group Category (which defines Template SQL that contains tokens) and Criteria Values (which define specific values for the tokens in the Template SQL). These queries are used by "Eligible <Entity>s for this Group" zones that identify entities eligible for but not currently assigned to a group, as well as <Entity> Group Evaluation batch processes (see **Batch Processing** in **Configuring Entity Groups** for more information).

Entity Groups are defined by the following:

- **Status**: The current status of the entity group
- **Group Category**: The category to which the entity group belongs. Group categories are defined using the Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup. See below for more information.
- **Group Deactivation Option**: A flag that defines what happens to the relationship (captured in the Device Entity Group or Service Point Entity Group table, as appropriate) between the group and entities belonging to the group if or when the group is deactivated. Options include "Expire" and "Do not Expire".

Copyright © 2026, Oracle and/or its affiliates.

- **Group Assignment Option**: A flag indicating the manner in which groups are assigned to entities. Options include System (via batch process), User (manually), System and User (either).
- **Criteria Value**: One or more criteria values used for selecting entities for the group, based on a Criteria Source and Criteria Value.
  - **Criteria Source**: Defines the source for criteria used for selecting entities for the group (such as a Specification, Rate, or Program). Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See below for more information.
  - **Criteria Value**: Defines specific values for the criteria that are used when selecting entities for the group. For instance, values for a **Criteria Source** of "Rate" could be rates defined in an extendable lookup.

System Data

Entity Groups are based on the following system data:

Data Type	Description
Maintenance Object	Entity Group (D1-ENTGRP)
Business Object(s)	Entity Group (D1-EntityGroup)

Entity Group Category Extendable Lookup

The Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup defines categories of entity groups. These are used to define groups of entities (either devices or service points) that share the criteria and template SQL for identifying and retrieving entities but that may have different criteria values. For example, two device entity groups, "Device Group A" and "Device Group B" can reference the same "Specification and Rate" category, which uses Specification and Rate Schedule criteria to identify and retrieve devices (devices in group A are those with a unique combination of Specification and Rate Schedule, while those in group B have a different unique Specification and Rate combination). The category also defines template SQL that uses the "Specification" and "Rate" criteria when identifying and retrieving devices. The Entity Group Category Template SQL contains tokens, and an Entity Groups Criteria Values define specific values for the tokens in the Template SQL.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group category.
- **Target Maintenance Object**: The maintenance object of the entities that belong to entity groups of this category. Valid options include Device (D1-DEVICE) or Service Point - MDM (D1-SP).
- **Criteria Option**: Flag that indicates whether or not the criteria source along with the SQL template and SQL builder algorithm will be recorded for the entity group in this category. Valid values are "Applicable" and "Not Applicable".
- **Template SQL**: The template SQL used to identify and retrieve entities for entity groups of this category. Applicable only when Criteria Option is set to "Applicable".
- **Entity Group Algorithm**: An algorithm that executes the template SQL, inserting the criteria values from the entity group, to identify and retrieve entities for entity groups of this category.
- **Criteria Sources**: The criteria sources used by the template SQL when identifying and retrieving entities for entity groups of this category. Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See

Copyright © 2026, Oracle and/or its affiliates.

below for more information. Applicable only when Criteria Option is set to "Applicable". Each criteria source is defined by the following:

- **Criteria Source**: The name of the criteria source (from the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup)
- **Is Required**: A flag that indicates if a value is required for the criteria source (Required or Optional)
- **Marker**: The token used in the template SQL for the criteria source and value, in the format of "#xxx". For example, the marker for a Specification criteria source might be "#SPEC".
- **Condition Operator**: The operator used in `WHERE` clauses in the template SQL to filter data based on the criteria source. Values are defined in the Condition Operator (CONDITION_OPR_FLG) lookup.

The base package includes the "SP Measurement Extract" value for this extendable lookup.

Criteria Source Extendable Lookup

The Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup defines the criteria that can be configured for an entity group. Each value defines a single criteria source, such as Specification, Rate, or Program, as well as how the user will enter criteria values for the criteria source, and the source of criteria values for the criteria source. This configuration is used in the **Criteria Value** section of an Entity Group.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group criteria source
- **Presentation**: The presentation used for the criteria source. Options include:
  - **Drop down**: The user selects a the criteria value from a drop-down list.
  - **Input Text**: The user enters text.
  - **Input Text with Date Picker**: The user selects a date.
  - **Input Text with Date Time Picker**: The user selects a date and time.
- **Value Source**: The source of values for the criteria source. Options include:
  - **Characteristic Type (Pre-Defined)**: Values are from a list of pre-defined values for a characteristic type
  - **Extendable Lookup**: Values are from an extendable lookup
  - **Service Script**: Values are retrieved by a service script
  - **Standard Lookup**: Values are from a standard lookup
- **Value Retriever**: The specific value source used for the criteria source, based on the Value Source. For example, if the Value Source is "Standard Lookup", this would be the lookup code.

Example: A "Rate" criteria source where values are derived from the Customer Rate Schedule (D2-CCBRateScheduleLookup) extendable lookup could be defined as follows:

- **Criteria Source**: Rate
- **Description**: Rate
- **Status**: Active
- **Presentation**: Drop down
- **Value Source**: Extendable Lookup

Copyright © 2026, Oracle and/or its affiliates.

- **Value Retriever**: D2-CCBRateScheduleLookup

Entity Groups

You use the **Entity Group** portal to display and maintain entity groups.

Refer to Understanding Entity Groups for more information.

You can access the portal by selecting **Admin**, select **General**, then select **Entity Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Entity Group**: This zone displays basic information about entity group.
- **Eligible Devices for this Group**: This zone shows a list of devices that are eligible for assignment to this entity group. The list is based on the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more devices to the group, select the devices and click **Assign**. Applicable only for "Device" entity groups.
- **Devices in this Group**: This zone shows a list of devices that have been assigned to this entity group. To delete the relationship between the device and the entity group, select the device and click **Delete**. To expire the relationship between the device and the entity group, select the device and click **Expire**. Applicable only for "Device" entity groups.
- **Eligible Service Points for this Group**: This zone shows a list of service points that are eligible for assignment to this entity group. The list is built from the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more service points to the group, select the service points and click **Assign**. Applicable only for "Service Point" entity groups.
- **Service Points in this Group**: This zone shows a list of service points that have been assigned to this entity group. To delete the relationship between the service point and the entity group, select the device and click **Delete**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. Applicable only for "Service Point" entity groups.
- **Customer Address Search for Grouping**: This zone lists service points that are eligible for assignment to this entity group (and provides search parameters to filter the results). To assign one or more service points to the group, select the service points from the search results and click **Assign**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".
- **Service Points**: This zone shows a list of service points that have been assigned to this SP Measurement Extract entity group. To unassign a service point from the entity group, select the device and click **Unassign**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. To create an extract request for the group, click **Add**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".

The following zones may appear as part of the portal's **Requests** tab page:

- **Requests**: This zone lists the requests made for the entity group, such as requests for measurement consumption extract. To a request for the entity group, click **Add** in the zone title bar.

Copyright © 2026, Oracle and/or its affiliates.

Batch Processing

The Device Group Evaluation (D1-DGEV) and Service Point Group Evaluation (D1-SPGEV) batch controls are used to identify devices or service points (respectively) that are eligible for a particular group and assigns them to a specified group, or expires relationships between devices or service points (respectively) and a specified group.

Devices or service points that are not currently assigned to an entity group for a specified category will be assigned to an eligible group. Only one group for the category may be assigned. If the device or service point is currently assigned a group for that category, it will be evaluated for eligibility, and if the current group is no longer eligible, the group is expired. If the same device becomes eligible for another group, that group is assigned to the device or service point.

Parameters:

- **Group Code Category**: A group category used to identify eligible entities. Required if the **Mode** is "Evaluate Group Code Category"
- **Group Code**: A group used to identify eligible entities, Required if the **Mode** is "Expire Device Group Code", "Expire Service Point Group Code" or "Evaluate Group Code", and is optional if the **Mode** is "Evaluate Group Code Category".
- **Mode** (Required): The mode by which the process evaluates entities. Options include:
  - **Expire Device Group Code (D1ED)**: Used to expire device entity group relationships for the specified Group Code
  - **Expire Service Point Group Code (D1ES)**: Used to expire service point entity group relationships for the specified Group Code
  - **Evaluate Group Code (D1EG)**: Used to search entities based on Group Code, and assigns eligible devices or service points to the specified Group Code
  - **Evaluate Group Code Category (D1GC)**: Used to search entities based on Group Code Category (and optionally a Group Code). This mode assigns devices or service points eligible for a group of the specified category (and Group Code if provided), and expires relationships between devices or service points and groups of the specified category (and Group Code is provided).

Note that these batch controls are used with groups whose Group Assignment Option is either "System" or "System or User".

Weather Stations

- Understanding Weather Stations
- Weather Stations

Understanding Weather Stations

Weather stations are facilities that measure atmospheric conditions and collect weather data such temperature, humidity, wind speed and direction, and barometric pressure.

Weather stations are defined by the following:

- **Weather Station**; An ID for the weather station.

Copyright © 2026, Oracle and/or its affiliates.

- **Weather Meter**: The weather meter (see below) that captures weather data for the weather station.
- **Status** A flag that indicates if the weather status is Active or Inactive.
- **Geographic Longitude**: The geographic longitude of the location of the weather station.
- **Geographic Latitude**: The geographic latitude of the location of the weather station.
- **External ID**: An ID used for the weather stations in external systems, such as a distributed energy resource management (DERM) system.

Weather Meters

Weather meters are virtual devices used to store weather data for a weather station within the application. Weather meters can have one more interval measuring components, each configured to store a specific type of weather data (temperature, wind speed, humidity, and so on).

Weather meters are defined by the following:

- **Device Type**: The device type for the weather meter. This should be based on the Weather Meter Type (D1-WeatherMeterType) business object.
- **Status**: The current status of the weather meter (Active or Inactive)
- **Serial Number**: An identifier for the weather meter. This is based on the ID of the Weather Station.

Like other devices, weather meters require device types. Device types for weather meters are created based on the Weather Meter Type (D1-WeatherMeterType) business object. In addition, weather meters have related device configurations based on the Device Configuration (D1-DeviceConfiguration) business object, and measuring components based on the Interval Channel (D1-IntervalChannel) business object.

See Creating Weather Metersfor more information about how weather meters are created.

Weather Stations

You use the **Weather Station** portal to display and maintain weather stations.

Refer to Understanding Weather Stations for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Weather Station**.

The following zones may appear as part of the portal's **Main** tab page:

- **Weather Station List**: This zone lists all weather stations defined in the system. Broadcast a record to display the details of the selected record.
- **Weather Station**: This zone displays the selected weather station.
- **Weather Station Characteristics**: This zone displays characteristics for the selected weather station.

Importing Weather Stations

Weather stations can be imported along with other facility and network location data such as transformers, feeders, and substations. Refer to Importing Facility and Network Location Data from a Distributed Energy Resource System for more information.

Copyright © 2026, Oracle and/or its affiliates.

Creating Weather Meters

Weather meters, including device configurations and measuring components, are created automatically when a weather station is created, based on the Device Type selected when creating the weather station.

This process requires the following data be set up prior to creating weather stations.

Measuring Component Type(s)

One or more measuring components used to capture weather data. Each should be defined as follows:

- **Business Object**: Interval Channel Type - Physical (D1-IntervalChannelTypePhysical)
- **Measuring Component Business Object**: Interval Channel (D1-IntervalChannel)
- **Measurement Business Object**: Measurement (D1-Measurement)
- **Service Type**: The service type associated with how weather data stored by measuring components of this type will be used (typically "Electric").
- **Interval /Scalar**: Interval
- **Consumptive/Subtractive**: Consumptive
- **Interval Size**: The size of the intervals for weather data values stored by measuring components of this type, represented as hours:minutes:seconds (HH:MI:SS)
- **Value Identifiers**: One or more value identifiers for the weather data captured by measuring components of this type (Degrees F, Degrees C, and so on).

Device Configuration Type

A device configuration type that defines the types of measuring components to be created for the weather meter. This should be defined as follows:

- **Business Object**: Device Configuration Type (D1-DeviceConfigurationType)
- **Device Configuration Business Object**: Device Configuration (D1-DeviceConfiguration)
- **Service Type**: The service type associated with how weather data stored by devices of this device configuration type will be used (typically "Electric") This must match the service type defined for the measuring component type.
- **Valid Measuring Component Types**: One or more measuring components (defined above) used to capture weather data. Each should be marked as "Required".

Device Type

A device type that defines the type of weather meter to be created. This should be defined as follows:

- **Business Object**: Weather Meter Type (D1-WeatherMeterType)
- **Device Business Object**: Weather Meter (D1-WeatherMeter)
- **Service Type**: The service type associated with how weather data stored by devices of this type will be used (typically "Electric"). This must match the service type defined for the device configuration type and measuring component type.
- **Valid Device Configuration Types**: A device configuration type that defines the measuring component types to be created for the weather meter.
- **Characteristics**: Characteristics that define the type of data stored by the device (such as temperature)

Copyright © 2026, Oracle and/or its affiliates.

Batch Processing

The Map Service Points to corresponding Weather Stations (D1-MAPWS) batch process can be used to map service points (including metered service points) to weather stations when and as needed.

The Initial Data Export D1-WTHRSTN (D1WSTNX) batch process can be used to export weather station data when and as needed.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Importing Weather Data

Weather data can be imported into the system using the Adapter Development Kit (ADK) in the same manner as used for measurement data. When importing weather data, the weather station ID should be used as the Device ID (the Serial Number on Weather Meters is based on the Weather Station ID). In addition, this requires that appropriate Unit of Measure (UOM), Time of Use (TOU), and Service Quantity Identifiers (SQI) values are defined to represent the weather data used in your implementation, and corresponding UOM translation mapping values are defined in the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup.

See About Measurements and Measurement Processing in the *Business User Guide*, and Configuring Payload Processing for more information.

Postal Code Defaults

- Understanding Postal Code Defaults
- Postal Code Defaults

Understanding Postal Code Defaults

Postal Code Defaults are used to define ranges of postal codes associated with a weather station. These are used when mapping service points (including metered service points) to weather stations when the longitude and latitude of the service point is not available.

Postal code defaults are defined by the following:

- **Postal Code From**: The starting point of the range of postal codes for the postal code default.
- **Postal Code To**: The ending point of the range of postal codes for the postal code default.
- **Country**: The country in which the postal code default resides.
- **City**: The city in which the postal code default resides.
- **County**: The county in which the postal code default resides.
- **State**: The state in which the postal code default resides.
- **Version**: The version of the postal code default.
- **Time Zone**: The time zone in which the postal code default resides.
- **Weather Station**: The weather station associated with the postal code default.

Copyright © 2026, Oracle and/or its affiliates.

Postal Code Defaults

You use the Postal Code Default portal to display and maintain postal code defaults.

Refer to Understanding Postal Code Defaults for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Postal Code Default - MDM**.

The following zones may appear as part of the portal's **Main** tab page:

- **Postal Code Default List**: This zone lists all postal code defaults defined in the system. Broadcast a record to display the details of the selected record.
- **Postal Code Default**: This zone displays the selected postal code default.
- **Postal Code Default Characteristics**: This zone displays characteristics for the selected postal code default.

Batch Processing

The Data Export D1_POSTAL_DFLT (D1POSTTX) batch process can be used to export postal code default data when and as needed. Use the **Batch Control** portal to view the parameters used by this batch process.

Measurement Cycles

- Understanding Measurement Cycles
- Configuring Measurement Cycles

Understanding Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at Service Points in that cycle. Measurement cycles can have one or more associated routes used to collect measurements.

Measurement cycles can also be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle.

Refer to Measurement Cycle Processing for details on features, configuration and implementation.

Configuring Measurement Cycles

This portal is used to display and maintain a Measurement Cycle.

Refer to Understanding Measurement Cycles for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Measurement Cycle**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Measurement Cycle List**: This zone lists all Measurement Cycle records. Broadcast a record to display the details of the selected record.
- **Measurement Cycle**: This zone provides information about the selected Measurement Cycle.
- **Measurement Cycle Route List**: This zone lists the measurement cycle routes related to the measurement cycle. Click the **Route** link to view details for a measurement cycle route in the **Measurement Cycle Route** portal.
- **Measurement Cycle Schedule List**: This zone lists the measurement cycle schedules related to the current measurement cycle. Click the **Scheduled Selection Date** link to view details for a measurement cycle schedule in the **Measurement Cycle Schedule** portal.

Configuring Measurement Cycle Routes

The Measurement Cycle Route portal is used to maintain individual measurement cycle routes.

You can access the portal by clicking the **Route** for a specific measurement cycle route in the **Measurement Cycle Route List** zone of the **Measurement Cycle** portal. You are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measurement Cycle Route**: This zone provides information about the selected Measurement Cycle Route.
- **Measurement Cycle Route Characteristics**: This zone lists characteristics for the selected Measurement Cycle Route.

Measurement Cycle Processing

Measurement cycle processing supports the following processes:

- **Meter Read Download** (for service point meters scheduled to be read for the measurement cycle): This process stages the activity for the extraction and download of the service point and meter information scheduled for reading.
- **Bill Determinant Push** (for subscribing external applications): This process stages usage transactions that calculate usage that will be sent to the subscribing external application. In this case, measurement cycles can be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle. Even Service Points whose meters are read automatically may reference measurement cycles. Creating bill determinants (by creating a usage transaction) is performed by an algorithm on the "Complete" state of the SP/Measurement Cycle Schedule Route business object (similar to creating activities as described above).

Measurement cycle processing is managed by the following three batch processes:

- **Create Pending Measurement Cycle Schedule Routes** (D1-CMCS): This batch process creates Schedule Routes for Measurement Cycle Schedules whose schedule selection date is on or before the batch business date. This process is used if routes have the same schedule each month, quarter, and so on. This process simply copies the routes from the Measurement Cycle to the Measurement Cycle Schedule on/after the scheduled selection date.
- **Create Pending SP / Measurement Cycle Schedule Route Records** (D1-CSPSR): This batch process creates a "SP/Measurement Cycle Schedule Route" transaction for every Service Point in the Measurement Cycle Schedule Route that is ready for processing.

Copyright © 2026, Oracle and/or its affiliates.

- **Process Pending SP / Measurement Cycle Schedule Route Records** (D1-PSPSR): This batch process transitions the Pending "SP/Measurement Cycle Schedule Route" transactions to their Complete state. Algorithms can be configured to create "Meter Read Download" activities or usage transactions. These algorithms would be configured as an Enter algorithm on the "Complete" state of the SP/ Measurement Cycle Schedule Route business object (see below).

Measurement processing requires the following configurations:

- **Meter Read Download**: Configure the Create Meter Read Download Activity (D1-CRMRDACT) algorithm on the "Complete" state of the SP / Measurement Cycle Schedule Route (D1-SPMsrmtCycScheduleRoute) business object.
- **Bill Determinant Push**: Configure the Create Usage Transaction (D2-CRUSGTRN) algorithm on the "Complete" state of the SP / Measurement Cycle Schedule Route (D1-SPMsrmtCycScheduleRoute) business object.

Refer to A Description of the Cyclical Meter Read Process in the *Oracle Utilities Customer to Meter Administrative User Guide* for more information on meter read download (available with Customer to Meter and Customer Cloud Service only).

Refer to About Usage Transactions for more information about usage transactions.

Measurement Cycle Schedules

- Understanding Measurement Cycle Schedules
- Configuring Measurement Cycle Schedules

Understanding Measurement Cycle Schedules

Measurement cycle schedules define the dates on which devices are scheduled to be read for a given measurement cycle and the routes used to collect measurements for the measurement cycle.

For a deeper functional understanding, refer to the About Route Management section of the *Oracle Utilities Meter Solution Business User Guide*.

Configuring Measurement Cycle Schedules

This portal is used to display and maintain a Measurement Cycle Schedule.

Refer to Understanding Measurement Cycle Schedules and Understanding Measurement Cycles for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Measurement Cycle Schedule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measurement Cycle Schedule Query:** This zone allows you to query for Measurement Cycle Schedules based on Measurement Cycle and select the desired record.
- **Measurement Cycle Schedule:** This zone provides information about the selected Measurement Cycle Schedule.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Metered Premise Options

This section describes tables that must be set up before you can define premises.

- Setting Up Meter Read Instructions
- Setting Up Meter Read Warnings

Setting Up Meter Read Instructions

When you set up a premise you may define instructions to be supplied to the individuals who read the meters located at the premise. To define meter read instruction codes, open **Admin**, and then **Geographic**, and then **Meter Read Instruction.**

Description of Page

Enter a unique **Meter Read Instruction Code** and **Description** for every meter read instruction.

Setting Up Meter Read Warnings

When you set up a premise you may define warnings to be supplied to the individuals who read the meters located at the premise. To define meter read warning codes, open **Admin**, and then **Geographic**, and then **Meter Read Warning.**

Description of Page

Enter a unique **Meter Read Warning** and a **Description** for every meter read warning.

Configuring Cross Installed Meter Processing

Cross installed meters is the term for situations in which meters are deployed at an incorrect customer service point. Consider the following example: Meter A is supposed to be installed at Service Point A and Meter B is supposed to be installed at Service Point B, but Meter B was mistakenly installed at Service Point A, and Meter A was mistakenly installed at Service Point B. This means that both customers will be billed incorrectly (customer at Service Point A will be billed for usage at Service Point B and vice versa) until the system can updated to reflect the correct meter installations.

This process can also be used to issue financial adjustments and/or cancel/rebill any bill segments that are impacted by the change in meters, and create customer contacts related to the changes. See Adjustments and Cancel / Rebill Incorrect Bill Segments for more information about adjustments and rebilling.

The Cross Installed Meters process provides a tool to re-configure the meters through a correction process. When a user initiates the Cross Installed Meters process, the system will update device installation event details (like status, date and time of installation), device configurations, and measuring components with the correct information.

Cross installed meter processing requires the following administrative data:

Copyright © 2026, Oracle and/or its affiliates.

- Customer Contact Class and Type (optional)
- Request Type
- Action Method

Each of these is described in detail below.

See Correcting Cross Installed Meters in the *Business User Guide* for more information about running cross installed meter processing.

Create a Customer Contact Class and Customer Contact Type - Optional

If you want customer contacts created as a result of cross-installed meter processing, you will need to reference a Contact Class and Contact Type. If you want to use a specific contact class and/or contact type, you will need to create these.

See Setting Up Customer Contact Options for more information about creating contact classes and contact types.

Create a Request Type

Each process flow instance has an associated request record, which is based on a specific request type. You must create a request type based on the "Cross Installed Meter Request Type" business object that will be used for cross installed meter processing.

Use the following procedure to create a request type used with cross installed meter processing

1. Select **Admin**, **General**, **Request Type**, **Add**.
2. Select "Cross Installed Meter Request Type" from the **Request Type Business Object** drop-down list on the **Select Business Object** screen and click **OK**.
3. Enter the following on the **Cross Installed Meter Request Type** screen:
    - **Main**:
        - **Request Type**: A code for your request type
        - **Description**: A description of your request type
    - **Crossed Installed Meter Parameters**:
        - **Swap Meters Service Script**: Cross Installed Meter - Swap Meters (this is the script that performs the change to Install Events)
        - **Insight Type**: Cross Installed Meters - Inline (default)
        - * **Field Task Type**: D1-ServiceInvestigation (this defines the Field Task Type for field activities created by clicking the **Investigate** button on the **Swap Meters** panel)
        - **Create Customer Contact**: Indicates if customer contacts should be created as part of the cross installed meter process
        - **Contact Class**: The contact class used for customer contacts created from cross installed meter process flows. Applicable only if **Create Customer Contact** is set to “Yes”.
        - **Contact Type**: The contact type used for customer contacts created from cross installed meter process flows. Applicable only if **Create Customer Contact** is set to “Yes”.
    - **Bill Correction Parameters**:

Copyright © 2026, Oracle and/or its affiliates.

- **Allow Non-Consecutive Periods**: Indicates whether or not selecting non-consecutive bill segments for a service agreement is allowed when correcting bill segments
- **Adjustment Type**: Indicates the type of adjustment used when correcting bill segments
- **Allow Rebill**: Indicates whether or not to cancel/rebill when correcting bill segments
- **Allow Estimates**: Indicates whether or not to allow the use of estimated usage when using cancel/rebill to correct bill segments
- **Billing Option**: Indicates the billing option (Use Cut off Date or Use Schedule) used when using cancel/rebill to correct bill segments
- **Cancel Reason**: Cross Installed Meter (this is the Cancel reason used when using cancel/rebill to correct bill segments)

4. Click **Save**.

See Setting Up Request Types for more information about creating request types.

Create an Action Method

You must also configure an Action Method to associate the base package Cross-Installed Meter Process Flow Type (X1-CROSSEDMETER) with your custom Request Type.

Use the following procedure to create an action method used with cross installed meter processing

1. Select **Admin**, **System**, **Action Method**, **Add**.
2. Select "Cross Installed Meter Action Method" from the **Request Type Business Object** drop-down list on the **Select Business Object** screen and click **OK**.
3. Enter the following on the **Cross Installed Meter Action Method** screen:
   - **Main**:
     - **Action Method**: A code for your action method
     - **Action Method Role**: Cross Installed Meter
     - **Description**: A description of your request type
   - **Default Details**:
     - **Process Flow Type**: Cross Installed Meter (this is provided with the base package)
     - **Request Type**: Cross Installed Meter (the Request Type created above)
     - **To Do Type**: Cross Installed Meter (this is provided with the base package) This defines To Do Type for To Do Entries created by the Create Meter Process Completion Event.
     - **To Do Role**: A To Do Role for To Do Entries created by the Create Meter Process Completion Event.
4. Click **Save**,

See Defining Action Methods for more information about creating Action Method records.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Consumption Estimation Parameters

The system estimates consumption under the following situations:

- At billing time, when "real" consumption cannot be computed due to the lack of a meter read, the system estimates how much the customer used assuming the following conditions are true:
  - The service agreement allows estimation.
  - The rate schedule allows estimation.
  - The bill cycle schedule (if one is being used) allows estimation.
  - The register is not a peak register (this is an artificial constraint, but most organizations do not estimate peak consumption). Note that the base product estimation algorithms may attempt to estimate peak registers.
  - The register is subtractive. This is because only subtractive registers are self-correcting. For example, assume we estimate a reading of 100 and the next real read is 102, the customer will only be charged for 2 units. But consider what would happen if we estimated a consumptive register, the consumption associated with the next real read will be billed in its entirety.
- When a meter read is added, the system verifies that the resultant consumption is congruous with historical consumption trends (i.e., high / low checks are performed).

This section describes the tables that must be set up to estimate consumption.

Defining Device Measurement Options

- Units of Measure
- Service Quantity Identifiers
- Time of Use
- Service Types
- Factors
- Processing Timetable Types
- Initial Measurement Data
- Understanding Interval Price Set Types
- Defining Interval Price Set Types
- Understanding IMD Types
- Configuring IMD Types

Units of Measure

- Understanding Units of Measure
- Configuring Units of Measure

Copyright © 2026, Oracle and/or its affiliates.

Understanding Units of Measure

Units of Measure (UOM) identify quantities measured and recorded, such as KWH, KW, cubic feet, degrees Celsius, etc. UOMs are based on a specific service type.

> **Note**
>
> In Oracle Utilities Customer To Meter, Units of Measure are maintained via Merged Maintenance.

Important Unit of Measure System Events

Unit of Measure supports the following business object algorithm system events:

- **Axis Conversion**: this event receives a list of measurements along with a source UOM and interval size and a target UOM and interval size. It should then perform the necessary actions to convert the source UOM and interval size into the target UOM and interval size. Refer to the algorithm Axis Conversion algorithm (D1-AXIS-CONV) as an example.

Configuring Units of Measure

This portal is used to display and maintain a Unit of Measure.

Refer to Understanding Units of Measure for more information.

You can access the portal by selecting **Admin**, then **General**, then **Unit Of Measure**.

The following zones may appear as part of the portal's **Main** tab page...

- **Unit of Measure List:** This zone lists all units of measure. Broadcast a record to display the details of the selected record.
- **Unit of Measure:** This zone provides information about the selected unit of measure.

Service Quantity Identifiers

- Understanding Service Quantity Identifiers
- Configuring Service Quantity Identifiers

Understanding Service Quantity Identifiers

Service Quantity Identifiers (SQI) are used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Some examples include: Generated, Consumed, etc.

SQIs can also be used as a stand-alone representation of a service quantity that is not measured (one that is not properly described as a UOM) within a usage service quantity collection (such as a billing determinant).

Copyright © 2026, Oracle and/or its affiliates.

Configuring Service Quantity Identifiers

This portal is used to display and maintain a Service Quantity Identifier.

Refer to Understanding Service Quantity Identifiers for more information.

You can access the portal by selecting **Admin**, then **Rates**, then **Service Quantity Identifier**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Quantity Identifier List:** This zone lists all Service Quantity Identifiers. Broadcast a record to display the details of the selected record.
- **Service Quantity Identifier:** This zone provides information about the selected Service Quantity Identifier.

> **Note**
>
> In Oracle Utilities Customer To Meter, Service Quantity Identifiers are maintained via Merged Maintenance.

Time of Use

- Understanding Time of Use
- Configuring Time of Use

Understanding Time of Use

Time of Use (TOU) periods are modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

> **Note**
>
> In Oracle Utilities Customer To Meter, Time of Use periods are maintained via Merged Maintenance.

Configuring Time of Use

This portal is used to display and maintain a Time of Use.

Refer to Understanding Time of Use for more information.

You can access the portal by selecting **Admin**, then **Rates**, then **Time of Use**.

The following zones may appear as part of the portal's **Main** tab page:

- **Time Of Use List:** This zone lists all Time Of Use records. Broadcast a record to display the details of the selected record.

Copyright © 2026, Oracle and/or its affiliates.

- **Time Of Use:** This zone provides information about the selected Time Of Use.

> **Note**
>
> In Oracle Utilities Customer To Meter, Time of Use periods are maintained via Merged Maintenance.

Service Types

- Understanding Service Types
- Configuring Service Types

Understanding Service Types

Service Types define specific types of service for which usage can be recorded and captured, such as electric, gas, steam, etc.

Configuring Service Types

This portal is used to display and maintain a Service Type.

Refer to Understanding Service Types for more information.

You can access the portal by selecting **Admin**, then **General**, then **Service Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Type List:** This zone lists all Service Type records. Broadcast a record to display the details of the selected record.
- **Service Type:** This zone provides information about the selected Service Type.

> **Note**
>
> In Oracle Utilities Customer To Meter, Service Types are maintained via Merged Maintenance.

Factors

- Understanding Factors
- Configuring Factors

Understanding Factors

Factor are a centrally stored set of values for use in validation rules, bill determinants calculations, and other processes.

Copyright © 2026, Oracle and/or its affiliates.

A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a service point. Examples of factors can include minimum and maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components, or VEE groups.

A factor's values are effective-dated values - either a number, a profile measuring component, a temperature profile measuring component, a VEE group, or some custom-defined value - assigned to a factor and associated to the value of some attribute of a system object. For example, consider a service point that can be classified as residential, commercial, or industrial. The tolerance percentage by which a customer's consumption can exceed last month's consumption can be based on the service point category.

For this example, factor values for a single factor called "tolerance percentage" could be:

- Residential - 20%
- Commercial - 10%
- Industrial - 5%.

Configuring Factors

This portal is used to display and maintain Factors and Factor Values.

Refer to Understanding Factors for more information.

Prerequisites: You must define factor characteristic source algorithms, factor characteristic types, and factor characteristic values before you can create a factor. Refer to the Oracle Utilities Application Framework online help for more information about algorithms, characteristic types, and characteristic values.

You can access the portal by selecting **Admin**, then **General**, then **Factor**.

The following zones may appear as part of the portal's **Main** tab page:

- **Factor List:** This zone lists all factor records. Broadcast a record to display the details of the selected record.
- **Factor:** This zone provides information about the selected factor.
- **Factor Char Value and Factor Value List:** This zone lists the characteristic values associated to the factor characteristic. For each characteristic value it will display the related factor values. From this zone you can click the **Factor Value** link to see the factor value in more detail on the **Factor Value** portal.

Processing Timetable Types

- Understanding Processing Timetable Types
- Configuring Processing Timetable Types

Understanding Processing Timetable Types

Processing timetable types define types of schedules that can be referenced by different processes and objects in Oracle Utilities applications such as Oracle Utilities Meter Data Management and Oracle Utilities Market Settlements Management. For example, if a process

Copyright © 2026, Oracle and/or its affiliates.

requires requesting data from an external system on a daily basis, a daily processing timetable type could be used to define the details of this schedule.

Processing timetable types are used by standalone data request measuring components to control the schedule by which requests for data are processed (for example, in a settlement process in which zonal load is requested on a daily basis). They are also used in the attribute data snapshot creation process.

The base package includes two processing timetable type business objects used with specific scheduling patterns:

- **Daily Processing Timetable**: Used for daily schedules
- **Defined Dates Processing Schedule**: Used for schedules that should execute on specific dates.

Parameters used to define processing timetable types include:

- **Execution Frequency Class**: Designates the type of processing timetable, Daily or Defined Dates (based on the business object)
- **Scheduling Information**: Details about how the schedule is executed, including:
  - **Period Cut-Off Time**: The time of day that represents the end of a period. In many instances this will be 12:00 AM. However, for something like a "Gas Day" this might be set to 09:00 AM.
  - **Earliest Execution Time**: Used to prevent records from being processed until after this time each day. This time relates to the processing time.
  - **Lag / Lead**: A flag that indicates whether periods in the past (Lag) or future (Lead) should be processed (applies to Daily schedules only)
  - **Lag Days / Lead Days**: The number of days in the past (Lag) or future (Lead) that should be processed (applies to Daily schedules only)
  - **Restrict Processing by Work Calendar**: Used to restrict processing to only the work days configured in the Work Calendar and will exclude any holidays added to the calendar (applies to Daily schedules only)
- **Workflow Information**: Details about how the processing timetable aligns with process workflows, including warning time and target due time
- **Processing Dates**: A list of specific dates on which the process should be executed (used with Defined Dates Processing Schedule only)

Configuring Processing Timetable Types

Refer to Understanding Processing Timetable Types for more information.

You can access the portal by selecting **Admin**, then **General**, then **Processing Timetable Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Processing Timetable Type List**: This zone lists all processing timetable type records. Broadcast a record to display the details of the selected record.
- **Processing Timetable Type**: This zone displays details for the selected processing timetable type.

Copyright © 2026, Oracle and/or its affiliates.

Initial Measurement Data

- Configuring the Initial Measurement Algorithms
- Configuring Measurement Logging
- Configuring Direct Measurement Processing
- Configuring Machine Learning Anomaly Scoring
- Removing Unwanted Initial Measurements

Configuring the Initial Measurement Algorithms

The behavior of initial measurement data processing can be adjusted by updating parameters in several key algorithms in the initial measurement lifecycle. This includes algorithms on initial measurement business objects used with Legacy Measurement Processing and algorithms on IMD types used by Direct Measurement Processing.

> **Note**
>
> This is not a list of all algorithms and algorithm parameters that can be customized in the initial measurement processing lifecycle. Rather, it is a selection of some of the more impactful parameters.

Initial Measurement Data Seeder Algorithms

- Derive Service Provider and Measuring Component (D1-DER-SPRMC) / Derive Service Provider and Measuring Component Service (D1-DERSPRMCS):
  - Enabling a standalone measuring component search allows initial measurement data to be processed without a device identifier. This is useful when profile or temperature data is loaded through an IMD Seeder or seeder IMD type and a match must be made on measuring component identifiers alone.
  - If measuring component identifiers can be repeated across measuring components for the same device then the error for duplicate measuring components being found can be turned off. This is useful for certain head end systems that will have multiple channels with the same channel identifier. When a duplicate is found the search will fall back on other means for identifier the correct measuring component.
- Perform Date/Time Adjustments and Undercount/Overcount Check (D1-DODTTMADJ) / Perform Date/Time Adjustments and Undercount/Overcount Check Service (D1-DODTTMADS):
  - Undercount validation can be turned off completely or left enabled. When left enabled the automatic filling of gaps can be turned off separately.
  - The overcount check can be turned off.
  - The automatic adjustment of the individual interval date/times to an interval boundary can be turned on.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If your implementation receives initial measurement data with date/times that do not include an explicit time zone and the devices report date/times in standard time, you may need to add the following configuration:
>
> - Navigate to **Admin**, then **Integration**, then **Message Option**.
> - In the **XAI compliance** option, ensure that the following text is provided: `xsd:strict:dstGapInStandardTime`
>
> This will prevent a date/time reported in standard time that falls on the missing hour of the day Daylight Savings Time is entered from being misinterpreted.

Specific Initial Measurement Data Algorithms

- Normalize measurements (overwrite identical existing Measurements) (D1-NORM-IMD) / Normalize Measurements Service (overwrite identical Measurements) (D1-AUTO-NORS):
  - When the initial measurement data includes measurements that match exactly with the existing final measurements they can either be overwritten or skipped.
  - If logging of changes to final measurements is desired then it must be indicated as such in this algorithm. It is suggested to keep this turned off for high volume initial measurement data such as initial load. See Configuring Measurement Logging for more details.

> **Note**
>
> Refer to Configuring Consumption Synchronization for additional parameters that can be adjusted on specific initial measurement data algorithms.

Configuring Measurement Logging

There are two components to logging changes to final measurements:

- **Initial Measurement Finalization**: the "normalization" algorithm on a particular initial measurement data business object will determine how final measurements are updated. For performance reasons certain types of initial measurement data (e.g. initial load) are delivered with a final measurement update method that will skip any logging. This can be controlled by parameters on the algorithm that implements the algorithm type Normalize Measurements for Initial Measurement Data (D1-NORM-IMD) or Normalize Measurements for Initial Measurement Data Service (D1-NORM-IMDS) algorithm types. Specifically, with Legacy Measurement Processing the "Create Measurement Log on Update (Y/N)" parameter should be set to "N" as creating these logs has a performance impact. For Direct Measurement Processing, the "Create Measurement Audit (Y/N)") parameter should be set to "Y".
- **Measurement Business Object**: there must be an algorithm for the audit system event configured. The base package delivers the algorithm type Add Measurement Log Record (D1-AMSRMTLOG). It is this algorithm that actually records the log entries.

Copyright © 2026, Oracle and/or its affiliates.

Logging User Updates to Manual Initial Measurements

By default, manual edits made by users of these zones are not logged on the Log tab. Logging of manual edits to manual initial measurements can be enabled by adding a logging algorithm on an appropriate lifecycle state of the manual initial measurement business objects. The Log User Transaction (D1-LOGUSRTRN) base package algorithm can be used for this. This Enter algorithm is designed to be defined on the Initial state of the manual initial measurement business objects, but it can also be defined on any (non-transitory) Interim or Final state as well.

To ensure logging of any or all manual edits made to manual initial measurements, this algorithm should be specified on any state in which users will make manual edits. This will most often be the Pending or VEE Ready states, but could also include the Error, Exception, or Finalized states.

> ⚠ **Caution**
>
> When defining this algorithm, the user should exercise caution and determine if previous algorithms in the sequence within the state contain any form of transitioning logic that may inadvertently cause this algorithm to be bypassed.

Configuring Direct Measurement Processing

Direct Measurement Processing is a method of processing incoming initial measurement data designed to reduce the volume of initial measurement records in the database. Rather than create and update initial measurement data records, Direct Measurement Processing creates temporary staging records (in the IMD Control Staging (D1_IMD_CTRL_STAGE) table) and processes initial measurement data in memory and creates final measurements in the Measurement (D1_MSRMT) table. With Direct Measurement Processing, records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table are created only if an error is encountered during processing (including VEE processing). See Direct Measurement Processing in the *Business User Guide* for more information about Direct Measurement Processing.

Configuring the system for Direct Measurement Processing involves the following:

IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing. IMD Types are defined for specific measuring component types in the the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Understanding IMD Types for more information about IMD Types.

Configure Extendable Lookups

Extendable lookups are used with Direct Measurement Processing for payload processing as well as with Measuring Component Filtering.

Payload Processing

To configure Payload Processing for use with Direct Measurement Processing, enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Creating SGG Payload Processing Extendable Lookup Values for more information.

Measuring Component Filtering - Pass Through

Copyright © 2026, Oracle and/or its affiliates.

The Head End Status to Condition Map (D1-HeadEndStatusToConditionMap) extendable lookup is used to map head end systems to head end system status code mapping extendable lookups. This allows Direct Measurement Processing to identify which status code mapping extendable lookup to use based on the head end system associated with the incoming measurement (either specified in the measurement or derived from the measuring component).

Configure Head End System Processing Methods

To configure head end systems for use with Direct Measurement Processing, update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Creating Head End Systems for more information.

If your implementation uses "pass through" external measuring components, the appropriate (interval or scalar) pass through IMD type should be specified for each "pass through" external measuring component type.

Batch Processing

Batch processing is used with Direct Measurement Processing for payload processing, Measuring Component Filtering, and Measuring Component Ordering.

Payload Processing

Direct Measurement Processing receives initial measurement payloads via Payload Processing batch processes based on the following templates:

- Payload Processing Monitor Template (D1-PLPRM)
- Payload Processing Monitor Template with Export (D1-PLPSO)

See Creating Payload Processing Batch Controls for more information.

Measuring Component Filtering - Pass Through

When using Measuring Component Filtering, the Extract Passthrough IMDs In Error (D1-EPIMD) batch process can be used to process "Waiting for Extract" records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table. This batch process uses the following parameters:

- **Maximum IMDs Per File:** When extracting data, a new file will be created when this limit is reached. Note: this limit is applied per thread.
- **Stream Out File Location:** The file location or Object Storage bucket where initial measurement are posted for distribution to external systems. This should be a file path or the Value of the File Location Extendable Lookup value for the appropriate bucket (usage). See Referencing URIs for more information.
- **Stream Out File Name Prefix**: The prefix for files created in the Stream Out File Location. The file format will be <Stream Out File Name Prefix><batch run>_<thread>_<thread count>_<timestamp>{<_file sequence>}.json.gz. If the Stream Out File Location is set then this parameter is required.

Measuring Component Ordering

When using Measuring Component Ordering, the Process IMD Control Stage Data (D1-IMDCS) batch process can be used to process "Ready to Process" records in the IMD Control Staging table. This batch process uses the following parameters:

- **Value to restrict to IMD Control Stage Status**: Designates the status of records to process. When processing Measuring Component Ordering initial measurements, this should be set to "D1RD" ("Ready to Process").

Copyright © 2026, Oracle and/or its affiliates.

- **Enable sequence handling**: Designates (true or false) if initial measurements should be retrieved and ordered for a measuring component in ascending order by date time and create date time. When processing Measuring Component Ordering initial measurements, this should be set to "true".

See Measuring Component Ordering in the *Business User Guide* for more information.

Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

Configure Info Exception IMD Header Business Object

If you want Info Exception IMD Header records to be included in Initial Measurement Data Header Export data, the Create Measurement Extract Request for Info Exceptions (D1CMERINFOEX) algorithm must be configured as an "Enter" algorithm on the Finalized state of the Informational Exception Header (D1-InfoExceptionHeader) business object.

Configuring Machine Learning Anomaly Scoring

Machine Learning Anomaly Scoring is a method of processing incoming initial measurement data designed to use Artificial Intelligence and Machine Learning neural network models to recognize "normal" versus anomalous measurement values, and only perform Validation, Editing, and Estimation (VEE) processing for measurements deemed to be anomalous.

Configuring the system for Machine Learning Anomaly Scoring involves the following:

NOTE: Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.

Measuring Components and Measuring Component Types

Specific measuring components and/or measuring component types are enabled for Machine Learning Anomaly Scoring using the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

In addition, the anomaly threshold for those measuring components and measuring component types can be defined using the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** fields (respectively).

MDM Master Configuration

The Machine Learning Configuring section of the MDM Master Configuration defines how results coming back from machine learning scoring are processed.

- **Exception Type**: specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an error during the scoring process.
- **Anomaly Score Threshold**: the limit for the anomaly score of any measurement within an initial measurement data. If the anomaly score (the difference between the Pre-VEE measurement value and the machine learning value) for one or more initial measurement data measurements is equal to or higher than this threshold then the initial measurement data will be considered anomalous and will be processed through VEE rules. Any initial

Copyright © 2026, Oracle and/or its affiliates.

measurements where all anomaly scores are below this threshold skip VEE processing and move to finalizing the initial measurement data to the measurement table.
Note that this can be overridden by the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** fields on specific measuring components or measuring component types (respectively).

Configure Payload Processing Extendable Lookups

To configure Payload Processing for use with Machine Learning Anomaly Scoring, you need to create two types of Payload Processing Extendable Lookup values:

- **Initial Payload Processing**: These are standard SGG Payload Processing Configuration (D1-SGGPayloadProcessing) extendable lookup values. You need to create one of these for each head end system from which you intend to process data using Machine Learning Anomaly Scoring. See Creating SGG Payload Processing Extendable Lookup Values for more information.
- **Machine Learning Payload Processing**: These are values for the SGG ML Payload Processing Configuration (D1-SGGMLPayloadProcessing) extendable lookup. You need to create one of these for each head end system from which you intend to process data using Machine Learning Anomaly Scoring. Values for this extendable lookup include the following:
  - **Payload Handler Class Name**: The Java class name for the processing handler to be used. This should be "com.splwg.d1.domain.sgg.ml.processing.MLResultPayloadHandler".
  - **Head-End System**: The head end system for which the configuration will apply. Select from the drop-down list.
  - **Load All ML Results**: A checkbox that indicates if Post VEE data should be created in the IMD regardless of anomalous status. This should be avoided in production environments as it increases the storage requirement for each Initial Measurement Data record which can lead to slower transaction processing.
  - **Payload Processing Result Type**: The type of payload to which the configuration applies. Machine Learning Anomaly Scoring is only supported for **Initial Measurements** at this time.
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. See Direct Measurement Processing in the *Business User Guide* for more information.
  - **Dynamic Parameters**: A list of parameter names and values that support additional processing rules associated with Machine Learning Anomaly Scoring. These include:
    - **boStateConditionCode**: Optional value. Specifies the business object State Condition (from the State Condition (F1_STATE_CONDITION_FLG) lookup) that the payload handler will check for in the initial measurement's current state before proceeding with processing of the payload. The base package includes the Waiting for ML (D1WM) State Condition defined for the “Waiting for ML” state on the Initial Load IMD business objects. Specifying this parameter with this value ensures that only initial measurements in the "Waiting for ML" state will be processed. If this parameter and value are set and the initial measurement is in some other state, processing stops and an error is written to the log.

Configure Post-ML VEE Processing (Optional)

You can define VEE Groups that are executed after machine learning processing to compare or adjust the measurement values based on a related measuring component, such as a scalar measuring component defined as the **Consumption Reference Measuring Component** on

Copyright © 2026, Oracle and/or its affiliates.

an interval measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the "Post-ML" VEE Role on measuring component types and measuring components.

Post-ML VEE Groups can include only the following VEE Rules:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement

To configure Post-ML VEE processing you need to do the following:

- Create one or more Post-ML VEE Groups containing one or more of the VEE rules supported for post-ML processing (listed above). Note that Post-ML VEE Groups can NOT reference Execute VEE Group or VEE Group Matrix rules.
- Add the Post-ML VEE Groups to the appropriate measuring component types and/or measuring components in the **Post-ML** VEE role. Note that Post-ML VEE Groups can only be added when machine learning is enabled for the measuring component or measuring component type via the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

Batch Processing

Batch processing is used with Machine Learning Anomaly Scoring for both initial payload processing as well as processing data returned from the machine learning model. See Creating Payload Processing Batch Controls for more information about creating Payload Processing batch processes.

Initial Payload Processing

Machine Learning Anomaly Scoring receives initial measurement payloads via Payload Processing batch processes based on the following templates:

- Payload Processing Monitor Template with Export (D1-PLPSO)

This is used to send initial measurements to Object Storage for machine learning processing.

Processing Data Returned from the Machine Learning Model

Machine Learning Anomaly Scoring processing data returned from the machine learning model using Payload Processing batch processes based on the following templates:

- SGG ML Payload Processing Monitor Template (D1-PLMLP)

This is used to retrieve measurements following machine learning processing, populate the IMD with machine learning values and the Anomaly Score Threshold, and calculate the Anomaly Score for each measurement value. Batch processes based in this template require an SGG ML Payload Processing Extendable Lookup value.

- Historical Data Extracts
- Machine Learning Comparative Reporting

Historical Data Extracts

When using Machine Learning Anomaly Scoring with interval measurements, the measurement data being processed includes enough historical data for the machine learning models to perform the anomaly scoring calculations, but when using Machine Learning

Copyright © 2026, Oracle and/or its affiliates.

Anomaly Scoring with scalar measurements, there is a chance that the models will not have access to sufficient historical data to perform those calculations.

In this case, you can extract historical data to supplement the cache of historical data used in the anomaly scoring calculations. This process supports extracts of both interval and scalar data.

Required Configuration

Extracting historical data requires that the **Enable ML-Based Validation** flag be set yo "YES" for all measuring component type for which data is to extracted.

Batch Controls

Extract of data and creation of extract files is performed by the Initial Extract for ML Cache (D1-INIML) batch process.

This batch process uses the following parameters:

- **Start Date/Time**: The start date and time of the data extract
- **Number of Days**: The number of days to extract (the process creates a separate file for each day)
- **Interval/Scalar**: Flag indicating which type of data to extract, interval (D1IN) or scalar (D1SC)
- **File Path**: The file path to where the process will post the file so that it can be ingested into the historical data cache used by the machine learning models. Refer to **Referencing URIs** (LINK) in the user documentation for more details.
- **File Limit**: The number of initial measurements to be included in the file (default is 10,000)

This process extracts data for ALL measuring components where the **Enable ML-Based Validation** flag on the Measuring Component Type is set yo "YES".

File Creation and Naming

Because this process can potentially create very large extract files, the files are created with a controllable sizes. Data is extracted in the Adapter Development Kit Native Format, and each file will only contain initial measurements for a single day, and for a default of 10,000 initial measurements. The logic for creating files differs slightly between interval and scalar data:

- Scalar: Each measurement is an initial measurement. For the common scenario there will be one initial measurement per day. However, some customers get scalar data more often per day or some measuring components may have adhoc reading (field or on demand read command based). For those less common scenarios there may be many initial measurements for a single measuring component for a single day.
- Interval: Files contain and initial measurement for each calendar day. Since each initial measurement can contain a variable number of measurements, there will only ever be one initial measurement per measuring component per day.

Extract files are named based on the following syntax:

pre-vee-init-<channel type>-<date being extracted>-<thread number>-<file number>-<batch run>.xml.gz.

where:

- <channel type> is the type of data to extract: interval (D1IN) or scalar (D1SC)

Copyright © 2026, Oracle and/or its affiliates.

- <date being extracted> is the date for the data being extracted, in YYYYMMDDHH24MISS format YYYY = year, MM = month, DD = day, HH= hour, MI= minute, SS = second)
- <thread number> is the thread number that created the file
- <file number>: is the number of the file, in cases where multiple files are created from same batch process run
- <batch run> is the batch number for the process

Machine Learning Comparative Reporting

Machine Learning (ML) Comparative Reporting enables you to compare ML anomaly results with Validation, Editing, and Estimation (VEE) results and other corrective actions. You can use this functionality to evaluate anomaly score thresholds before deployment and to monitor the effectiveness of ML anomaly scoring in production.

Before configuring comparative reporting, configure Machine Learning Anomaly Scoring, including the required machine learning model, payload processing, and batch controls. This includes set up and configuration of Oracle Cloud Object Storage Buckets (see Creating Object Storage Locations for more information).

Machine Learning (ML) Comparative Reporting can be used in two ways:

- **Historical Evaluation** evaluates existing interval measurement data (IMDs) processed using VEE for a selected period.
- **Ongoing Evaluation** captures comparison statistics for ML exceptions created during normal processing.
- Configuring and Running Historical Evaluations
- Configuring and Running Ongoing Evaluations
- Reviewing Comparative Results

Configuring and Running Historical Evaluations

You can use historical evaluations to score existing initial measurement data (IMDs) originally processed using VEE using ML, compare the ML results with original VEE results, and determine an appropriate anomaly score threshold.

This process use batch processes to create and extract historical "verification" data that can be evaluated by ML and then compared to the original VEE data, Another batch process captures the results in a statistics control record, which can be viewed using the **Meter ML and VEE Dashboard** portal.

> **Note**
>
> Historical verification data is used only for coordinating the scoring by the ML model and evaluation. This data does NOT create or update any measurement data.

> **Note**
>
> Historical evaluation should be used in non-production environments (such as DEV or TEST) ONLY.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Historical Evaluations

Configuring historical evaluation involves the following:

Create ML VEE Exception Type

Define a VEE Exception Type that specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an error during the scoring process when used with historical evaluation. This may be a different VEE exception Type than used for ongoing ML processing. See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

Update MDM Master Configuration

Update the **Machine Learning Configuration** section of the MDM Master Configuration as follows:

- **Exception Type**: Specify the ML VEE Exception Type created for ML historical evaluation (above).
- **Anomaly Score Threshold**: 0 (this ensures that the system retains the ML prediction and anomaly score for all IMDs selected for the evaluation)

See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

Configure Consumption Extract

Configure a Consumption Extraction Type for each ML model to be tested based on the ML Historical Verification Extract Type (D1-MLVerificationExtractType) business object. This construction extract type is used to define what is going to be scored as well as how VEE results should be interpreted as compared to ML results.

This should include the following attributes that define the model:

- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The frequency at which measurements are captured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring component types.
- **Service Point Types**: One or more service point types and UOM/TOU/SQI combination eligible for ML verification. These combinations determine which historical measuring components are evaluated.
- **VEE Exception Type Inclusions**: One or more VEE exception types that represent consumption quality issues ML is expected to detect. When this list is populated, any matching exception qualifies the IMD as having a VEE detected issue, regardless of severity. When empty, any non-informational VEE exception qualifies. You should exclude

Copyright © 2026, Oracle and/or its affiliates.

conditions that ML cannot evaluate so as to ensure ML to VEE comparisons remain meaningful.

- **UT Exception Type Inclusions**: One or more Usage Transaction exception types that indicate user intervention for an IMD with a qualifying VEE exception. The process compares the IMD with Usage Transactions for its Usage Subscription whose periods overlap the IMD period. A qualifying exception is interpreted as evidence of user action to resolve the IMD's data quality issue. When the list is populated, any matching exception counts regardless of severity. When empty, any non-informational Usage Transaction exception counts. Include only exception types that represent data quality issues so that comparisons between ML results and inferred user actions remain meaningful.

See About VEE and Understanding Exception Types for more information.

Configure Statistics Control

Configure a Statistic Control for each ML model to be tested based on the ML Historical Verification Control (D1-MLVerificationControl) business object. Snapshots created based on this statistics control compare the results from ML scoring with the previous VEE results, identifying when they agreed/disagreed. The process attempts to include user actions performed on VEE exceptions to determine if VEE exceptions were true issues or false positives.

These statistic controls are defined by the following:

- **Retention Mode**: Defines how to retain data for snapshots created from this statistic control
- **Snapshot Start Date**: The date on which snapshots created from this statistic control start.
  - **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
  - **Measurement Frequency**: The size of the intervals measured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
    - **For interval data**: This is the interval size.
    - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
  - **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **ML Threshold Override**: Use to override the **Anomaly Score Threshold** on the Master Configuration if applicable. This is not used with this historical evaluations.
- **Snapshot Business Object**: The business object to use when creating snapshots based on this statistic control. This should be ML Verification Statistics (D1-MLVerificationStatistics)

Configure SGG ML Payload Processing

Configure ML payload processing as described in Configure Payload Processing Extendable Lookups in Configuring Machine Learning Anomaly Scoring to process results returned from the ML model.

Running Historical Evaluations

Running historical evaluations involves the following:

Copyright © 2026, Oracle and/or its affiliates.

Create Historical Evaluation IMDs

Run the following batch processes in sequence to create "historical validation" initial measurement data that can be used for evaluation by running the following batch processes in sequence:

- Generate ML Historical Verification IMDs (`D1-GMLVI`) to select and create IMDs for evaluation. The IMDs are placed in the Waiting for ML state.
- Extract ML Historical Validation IMDs (`D1-EXHVI`) to export the selected IMDs for ML scoring.

Historical Verification IMDs:

- Are created only from "initial load" initial measurements,
- Duplicate the preVEE data which is sent to ML,
- Duplicate the VEE exceptions for statitsics computation
- Link to the original IMD for traceability

ML Processing

The machine learning model scores the data and sends the results to an object storage location (similar to the process used with Machine Learning Anomaly Scoring

Run ML Payload Processing and Analysis

Once the ML scoring results have been returned to object storage, run the following processes in sequence to process the payloads and capture information about the evaluation:

- SGG ML Payload Processing Monitor Template (`D1-PLMLP`) to process the returned ML results.
- IMD Monitor - Waiting for ML IMDs (`D1-IMDML`) to capture evaluation information and move the IMDs to the Analysis in Progress state.

Capture Statistics

Open the Historical Statistics Control, and click **Capture**. When the dialog opens, enter the **Start Date** and the **Number of Days** to process and click **OK**. This creates a historical ML Verification Statistics Snapshot for each requested day using the statistics control's **ML Threshold Override** or the **Anomaly Score Threshold** defined on the MDM Master Configuration.

Process ML Historical Statistics Snapshots

Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process to calculate daily ML Verification Statistics Snapshots. The process selects ML Historical Validation IMDs based on service type, interval/scalar mode, and frequency, then stores ML, VEE, user comparisons and anomaly-score distributions for dashboard analysis.

Re-evaluating Historical Results

You can make changes to the Consumption Extract Type or Statistics Control to fine tine the results and identify what works best for your implementation.

If you change the VEE Exception Type or Usage Transaction Exception Type configuration in the Consumption Extract Type:

- Run the Transition ML Historical Validation IMDs (`D1-TMHVI`) batch with the Transition Condition parameter set to `"D1RC"`

Copyright © 2026, Oracle and/or its affiliates.

- Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process again and review the updated results.

If you only change the **ML Threshold Override** on the Statistic Control:

- Perform the **Capture** action again
- Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process again and review the updated results.

This process can be run in stages:

- Stage1: ML scoring, run once
- Stage 2: Capture statistical inputs, which involves running `D1-TMHVI`), generating new snapshots, and running `D1MLVSSM` to calculate statistics
- Stage 3: Anomaly score evaluation, which involved creating snapshots and running statistics collection to see how the statistics change and to identify the right threshold

Removing Historical Evaluation Data

After completing a historical evaluation, run Transition ML Historical Validation IMDs (`D1-TMHVI`) batch processing with the Transition Condition parameter set to "`D1CN`" to complete the IMDs and remove their IMD control records.

To remove historical evaluation IMDs that were created in error, run the Remove ML Historical Validation IMDs (`D1-RMHVI`). batch process.

> **Note**
>
> Removing historical evaluation data is not mandatory, but if historical verification is being run on large amounts of data over a long period of time removing periods where research is complete help ensures that performance of the various processes does not degrade.

Configuring and Running Ongoing Evaluations

You can use ongoing evaluation to collect comparative reporting statistics during normal ML anomaly scoring processing.

> **Note**
>
> It is not expected that historical and ongoing evaluations would be configured and active for the same time period. Ongoing is intended for tracking live production ML results, whereas historical evaluations are intended for data that has already been processed using VEE.

Configuring Ongoing Evaluation

Configuring ongoing evaluations involves the following:

Create ML VEE Exception Type

Define a VEE Exception Type using the ML Exception with Statistics business object (`D1-MLExceptionWithStats`). This specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an

Copyright © 2026, Oracle and/or its affiliates.

error during the scoring process when used with ongoing evaluation. See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

> **Note**
>
> The ML Exception with Statistics business object (`D1-MLExceptionWithStats`) lifecycle delays capturing of statistics until after the production usage loading batch window so it can be done in off times during the day. The VEE exception itself is what is capturing the key inputs to the statistics calculations

Update MDM Master Configuration

Update the **Machine Learning Configuration** section of the MDM Master Configuration as follows:

- **Exception Type**: Specify the ML VEE Exception Type created for ML ongoing evaluation (above).
- **Anomaly Score Threshold**: The fallback threshold used when a threshold is not defined for the measuring component or measuring component type.

See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

Configure Consumption Extract

Configure a Consumption Extraction Type for each ML model to be tested based on the ML Historical Verification Extract Type (D1-MLVerificationExtractType) business object. Extracts created based on this type are used by the VEE Exception to capture the statistic inputs (such as which VEE exceptions to consider).

This should include the following attributes that define the model:

- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The frequency at which measurements are captured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **Service Point Types**: One or more service point types and UOM/TOU/SQI combination eligible for ML verification. These combinations determine which historical measuring components are evaluated.
- **VEE Exception Type Inclusions**: One or more VEE exception types that represent consumption quality issues ML is expected to detect. When this list is populated, any matching exception qualifies the IMD as having a VEE detected issue, regardless of severity. When empty, any non-informational VEE exception qualifies. You should exclude

Copyright © 2026, Oracle and/or its affiliates.

conditions that ML cannot evaluate so as to ensure ML to VEE comparisons remain meaningful.

- **UT Exception Type Inclusions**: One or more Usage Transaction exception types that indicate user intervention for an IMD with a qualifying VEE exception. The process compares the IMD with Usage Transactions for its Usage Subscription whose periods overlap the IMD period. A qualifying exception is interpreted as evidence of user action to resolve the IMD's data quality issue. When the list is populated, any matching exception counts regardless of severity. When empty, any non-informational Usage Transaction exception counts. Include only exception types that represent data quality issues so that comparisons between ML results and inferred user actions remain meaningful.

See About VEE and Understanding Exception Types for more information.

Configure Statistics Control

Configure a Statistic Control for each ML model to be tested based on the ML Ongoing Verification Control (D1-MLOngoingVerificationCntrl) business object. This is similar to the statistics control created for historical evaluation, except that it uses the VEE exception as the source of the statistical inputs.

These statistic controls are defined by the following:

- **Snapshot Frequency Hours**: The frequency (in hours) at
- **Retention Mode**: Defines how to retain data for snapshots created from this statistic control
- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The size of the intervals measured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).\
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **Snapshot Business Object**: The business object to use when creating snapshots based on this statistic control. This should be ML Verification Statistics (D1-MLVerificationStatistics)

Configure SGG ML Payload Processing

Configure ML payload processing as described in Configure Payload Processing Extendable Lookups in Configuring Machine Learning Anomaly Scoring to process results returned from the ML model.

Running Ongoing Evaluation

Running ongoing evaluations involves the following:

Batch Processing

Schedule the following batch processes to run on a regular basis:

Copyright © 2026, Oracle and/or its affiliates.

-
- Payload Processing Monitor Template with Export (`D1-PLPSO`) to process incoming measurement data and sends it to ML for scoring.
- SGG ML Payload Processing Monitor Template (`D1-PLMLP`) to process the returned ML results
- Process IMD Control Stage Data (`D1-IMDCS`) to process records in the IMD Control table, including:
  - Running appropriate VEE processing based on the ML result for each IMD with measuring component ordering.
  - The results are either IMDs created in error, a header IMDs created for exceptions, or published measurements
  - This is when the VEE exception will be created for ML
- VEE Exception Monitor (`D2-VEEDF`) to transition all pending VEE exceptions and create the appropriate statistic characteristics against the IMD associated to the VEE exception
- Statistics Control Monitor (`F1-STATS`) to create snapshots that are available for review on the Meter ML and VEE Dashboard.

Reviewing Comparative Results

You can use the **Meter ML and VEE Dashboard** to:

- Review anomaly-score distributions.
- Select an anomaly score threshold that provides a manageable number of exceptions for investigation.
- Compare ML exceptions with VEE results.
- Investigate cases where ML identifies an exception that VEE does not identify.
- Drill into individual IMDs to assess whether the selected threshold identifies meaningful issues.

You access this portal by selecting **Main Menu**, then **Dashboards**, and then **Meter ML and VEE Dashboard**. This portal includes the following zones:

- **ML Meter Statistics Snapshots Header**: used to define filters for data analysis
- **ML and VEE Comparison Matrix**: Compares the results of VEE rules against ML model predictions based on the current ML threshold.
- **ML and User Comparison Matrix**: Compares the actions users have taken on IMDs with VEE exceptions against ML model predictions based on the current ML threshold.
- **VEE and User Comparison Matrix**: Compares the actions users have taken on IMDs with VEE exceptions against VEE results.
- **User Actions**: Displays the conclusions drawn from the user's actions
- **Anomaly Score Distribution**: Displays the count of IMDs for a given anomaly score range

Embedded help icons provide additional details for each zone.

Removing Unwanted Initial Measurements

Copyright © 2026, Oracle and/or its affiliates.

Over time, implementations can accumulate unnecessary non-final initial measurements which effectively become "stuck" in the system. Examples of these types of initial measurements include:

- Erroneously imported duplicate initial measurements
- Initial measurements that failed validation and were not corrected or replaced
- Initial measurements that triggered service issue monitors that remain unresolved

These initial measurements can also have related VEE Exceptions, To Do Entries, and Service Issue Monitors that remain "stuck" as well. This unnecessary data can impact overall system performance and should be reviewed and removed from time to time.

The Remove Unwanted Initial Measurements feature can be used to identify and close or delete these unwanted non-final initial measurements and any related VEE exceptions, To Do Entries and service issue monitors. This process is initiated by creating Request records which specify the parameters used to locate the non-final initial measurements to be removed. The request in turn initiates a batch process which identifies non-final measurements and related VEE Exceptions, To Do Entries, and/or Service Issue Monitors.

This functionality uses the following system objects:

Object Type	Object Name	Description
Request Type Business Object	Remove IMDs Request Type (D1-RemoveIMDsType)	Used to define types of initial measurements removal requests. The Remove IMDs Request (D1-RemoveIMDsRequest) request business object (see below) is defined as the "Related Transaction BO" for this business object, meaning that requests created for request types based on this business object use the Remove IMDs Request (D1-RemoveIMDsRequest) request business object.
Batch Control	Remove Non-Final IMDs (D1-RENFI)	Used to identify non-final measurements and related VEE exceptions, To Do Entries and/or Service Issue Monitors (Service Tasks).
Request Business Object	Remove IMDs Request (D1-RemoveIMDsRequest)	Used to create and process initial measurement removal requests.
Fact Business Object	Remove IMD Object Count (D1-RemoveIMDOjbectCount)	Used to capture details of the initial measurements and related objects to be removed.

Remove Unwanted Initial Measurements Request Types

Requests used to remove initial measurements are based on request types which specify the batch control used to identify and remove the initial measurements.

To create request types used with this feature:

1. Select Menu, **Admin**, **General**, **Request Type**, **Add**.

Copyright © 2026, Oracle and/or its affiliates.

2. Select "Remove IMDs Request Type" from the **Request Type Business Object** drop-down list, and click **OK**.
3. Enter the following:
    - **Request Type**: A code for the request type
    - **Description**: A description of the request type
    - **Batch Control**: Search for and select the batch control to be used for requests of this type. The default is Remove Non-Final IMDs (D1-RENFI), but implementations can create custom versions of this batch control.
4. Click **Save**.

Remove Unwanted Initial Measurements Requests

To create an initial measurement removal request:

1. Select **Menu**, **Tools**, **Request**, **Add**.
2. Select the appropriate request from the **Request Type** drop-down list and click **OK**.
3. Enter the **Submission Parameters** for the request. These include:
    - **Mode**: The mode in which the process will be run. Options include:
        - **Preview**: Returns a count of non-final initial measurements and related objects (VEE Exceptions, To Do Entries, and Service Issue Monitors)
        - **Close**: Transitions non-final initial measurements to the "Discarded" state and completes and/or closes related objects. Details regarding the data impacted by "Close" requests are captured and displayed in the **Object Log** zone.
        - **Delete**: Deletes non-final initial measurements any related objects. Details regarding the data impacted by "Delete" requests are captured and displayed in the **Object Log** zone.
    - **Thread Count**: The number of processing threads allocated to the process
    - **IMD Business Object**: Used to restrict the initial measurements processed to those of a specific business object
    - **IMD Status**:Used to restrict the initial measurements processed to those in a specific non-final state, including Pending, Held, Additional Mapping, Mapping Error, VEE Ready, Defer, Error, VEE Complete, Exception, or Force Complete
    - **IMD Create Date/Time From**: Required. The start date and time of the time period in which initial measurements to be removed were created. For example, to remove initial measurements created between January 1, 2023 and January 31, 2023, the **IMD Create Date/Time From** would be "01-01-2023 12:00 AM".
    - **IMD Create Date/Time To**: Required. The end date and time of the time period in which initial measurements to be removed were created. For example, to remove initial measurements created between January 1, 2023 and January 31, 2023, the **IMD Create Date/Time To** would be "01-31-2023 12:00 AM".
    - **Device Configuration**: Used to restrict the initial measurements processed to those related to a specific device configuration.
4. Click **Save**. The request will then open in the **Request** portal.
5. Click **Submit Batch** to process the request.

Initially requests are created in the "Pending" status. When the request is run (by clicking **Submit Batch**), a batch process based on the batch control defined on the request type is executed.

Copyright © 2026, Oracle and/or its affiliates.

- If the batch process errors, the request **Status** is set to the "Batch Error".
- If the batch process completes, the request **Status** is set to the "Completed".

Details of the process are captured in the **Submission Information** section, and the **Object Log** zone.

The **Submission Information** section includes:

- **Batch Submission**: Link to the batch job submission for the batch process. Click the link to navigate to the **Batch Job Submission** portal to view details of the batch process.
- **IMD SDE Extract Through Date/Time**: The date and time through which specialized data extracts have been performed for initial measurements impacted by the process.
- **End of Batch Date/Time**: The date and time when the batch process was completed.

The **Object Log** zone contains table with details regarding the initial measurements and related objects processed, including:

- **Object**: The type of object processed (Initial Measurement Data, Service Task, To Do Entry, or VEE Exception)
- **Specialized Data Extract Occurred**: An indicator as to whether or not ("Yes" or "No") occurred for initial measurement data
- **Count**: The number of objects processed.

Example:

Object	Specialized Data Extract Occurred	Count
Initial Measurement Data	No	434
Initial Measurement Data	Yes	134
Service Task		1
To Do Entry		425
VEE Exception		413

Batch Control

The Remove Non-Final IMDs (D1-RENFI) batch control identifies and closes or deletes unwanted initial measurements and their related objects. When run, this batch process creates a fact (using the Remove IMD Object Count (D1-RemoveIMDObjectCount) business object) that captures details regarding the batch process and the objects being removed during the process. These details are viewed in the **Object Log** zone in the **Request** portal.

Note that the Request ID is a required parameter for this batch control, meaning that this batch control cannot be run on its own. It can be only run from an initial measurement removal request.

Implementations can create specific versions of this batch control for use when removing specific types of data. For instance, an implementation could create a "Preview" version of this batch control, or a version used only with interval initial measurements.

Implementations can alter the processing of this batch control by creating alternate algorithms for each of the System Events used by this batch process. Refer to **Ad-hoc Processes** in the **Plug-in Driven Background Processes** section for more information.

Copyright © 2026, Oracle and/or its affiliates.

Recommendations and Warnings

- We recommend using the **Preview** mode to identify potential initial measurements for removal before processing a request in either the **Close** or **Delete** mode. This allows you to review and verify the data to be removed before closing or deleting it.
- We recommend that access to this feature be limited to system administrators and other power users. Access to this feature can be restricted via the Remove IMDs Request BOAS (D1-RMIMDREQBOAS) application service.

> **Warning**
>
> If used improperly, this feature can inadvertently remove valid operational data (such as initial measurements in "Pending" status awaiting VEE processing). There is no process for "undoing" the removal of initial measurements and related objects using this feature.

Understanding Interval Price Set Types

Interval price set types define properties belonging to a class of interval price sets.

Interval price set types are defined by the following:

- **Service Type**: The service type to which interval price sets and interval prices based on this type will be applied
- **Interval Size**: The size of the intervals for interval prices based on this type, represented as hours:minutes:seconds (HH:MI:SS).
- **UOM**: The unit of measure used to define interval price sets and interval prices based on this type
- **TOU**: The time of use period used to define interval price sets and interval prices based on this type
- **SQI**: The service quantity identifier used to define interval price sets and interval prices based on this type
- **Short-Hand Description**: A short-hand description used for interval price sets of this type

> **Note**
>
> Either a UOM or SQI **must** be defined when creating an interval price set type.

System Data

Interval Price Set Types are based on the following system data:

Data Type	Description
Maintenance Object	Measuring Component Type (D1-MCTYPE)
Business Object(s)	Interval Price Set Type (D1-IntervalPriceType)

Copyright © 2026, Oracle and/or its affiliates.

Defining Interval Price Set Types

You use the **Interval Price Set Type** portal to display and maintain interval price set types.

Refer to **Understanding Interval Price Set Types** for more information./

You can access the portal by selecting **Admin**, select **Interval Price Set**, then select **Interval Price Set Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set Type**: This zone displays basic information about interval price set type.

Understanding IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing.

There are two main categories of IMD types - seeder types (based on the Seeder IMD Type (D1-IMDSeederType) business object) and specific types (based on the Specific IMD Type (D1-IMDType) business object).

- **Seeder IMD types** represents an initial measurement data (IMD) seeder, which serves as the universal input into the system for external callers or for initial measurement payload processing. The main purpose of the seeder IMD type is to determine the most appropriate specific IMD type (such as interval or scalar, or a type used with a particular Smart Grid Gateway adapter) to use for further processing and prepare the input data for that IMD type. Additionally, the seeder IMD type is used to perform critical validation (such as an interval count validation), device and channel identification, mapping of external values (such as Unit of Measure and time zone) to internal values, and time zone conversion for date/time fields.

  Note: There can only be one seeder IMD type active at any one time.

- **Specific IMD types** define the rules for processing a specific initial measurement data (IMD) record. Different specific IMD types contain rules for processing the different types of initial measurement data that can be received. For example, one IMD type defines the rules for processing a scalar measurement from the Landys+Gyr head end system while another IMD Type instance defines the rules for processing an interval initial measurement from the Sensus head end system. There are four types of algorithms supported for specific IMD types: Pre Processing, Pre-VEE, VEE, and Post-VEE. These algorithms are responsible for validating that the measurement has been correctly mapped to the IMD Type (Pre Processing), calculating consumption (Pre-VEE), performing VEE (VEE), writing to the final measurement table (Post-VEE), among other responsibilities.

IMD Types are defined for specific measuring component types in the the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Creating Head End Systems for more information.

IMD Types are defined by the following:

- **Initial Measurement Data Seeder**: A flag that indicates (Yes or No) if the IMD type represents a seeder IMD type. There can only be one active IMD type designated as a seeder IMD type at the same time. Seeder IMD types only use Pre-processing Algorithms (see below).

Copyright © 2026, Oracle and/or its affiliates.

- **IMD Business Object**: The business object to use when creating IMDs for the IMD type.
  - For seeder IMD types, this should be "IMD Seeder" (D1-IMDSeeder).
  - For non-seeder IMD types, this should be an initial load IMD business object such as "Initial Load IMD (Interval)" (D1-InitialLoadIMDInterval) or "Initial Load IMD (Scalar)" (D1-InitialLoadIMDScalar).
- **Status**: The status of the IMD type.
- **IMD Information Exception Header BO**: The bunsess to use when creating header records for "Information" exceptions. See VEE Processing and Exceptions with Direct Measurement Processing for more information.
- **Algorithms**: Algorithms used when processing initial measurement data for the IMD type, including:
  - Pre-processing Algorithms
  - Pre-VEE Algorithms
  - VEE Algorithms
  - Post-VEE Algorithms

System Data

IMD Types are based on the following system data:

Data Type	Description
Maintenance Object	Initial Measurement Data Type (D1-IMDTYPE)
Business Object(s)	Specific IMD Type (D1-IMDType)
	Seeder IMD Seeder Type (D1-IMDSeederType)

Base Package IMD Types

The base package includes several pre-defined IMD types that can be used when processing data from different head end systems supported through Smart Grid Gateway Adapters. These IMD types use vendor-specific initial measurement business objects and algorithms when processing data. The table below lists the IMD types provided with the base package.

IMD Type	Description
IMD Seeder Type (D1-IMDSEEDERTYPE)	Used as the seeder, which determines the appropriate specific IMD type to use and to prepare the input data for that IMD type
Interval IMD Type (D1-INT-IMDTYPE)	Used when processing interval data
Interval Passthrough IMD Type D1-INT-PT-IMDTYPE)	Used with "pass through" interval external measuring components to capture errors in pass through interval measurements
Scalar IMD Type (D1-SCA-IMDTYPE)	Used when processing scalar data
Scalar Passthrough IMD Type (D1-INT-SC-IMDTYPE)	Used with "pass through" scalar external measuring components to capture errors in pass through scalar measurements
Landis+Gyr - Interval IMD Type (D3-INT-IMDTYPE)	Used when processing interval data using the Landis+Gyr Adapter
Landis+Gyr - Scalar IMD Type (D3-SCA-IMDTYPE)	Used when processing scalar data using the Landis+Gyr Adapter
MV90 - Interval IMD Type (D5-INT-IMDTYPE)	Used when processing interval data using the MV90 Adapter

Copyright © 2026, Oracle and/or its affiliates.

MV90 - Scalar IMD Type (D5-SCA-IMDTYPE)	Used when processing scalar data using the MV90 Adapter
Sensus - Interval IMD Type (D6-INT-IMDTYPE)	Used when processing interval data using the Sensus Adapter
Sensus - Scalar IMD Type (D6-SCA-IMDTYPE)	Used when processing scalar data using the Sensus Adapter
SSN - Interval IMD Type (D7-INT-IMDTYPE)	Used when processing interval data using the Silver Spring Networks Adapter
SSN - Scalar IMD Type (D7-SCA-IMDTYPE)	Used when processing scalar data using the Silver Spring Networks Adapter
Itron - Interval IMD Type (D8-INT-IMDTYPE)	Used when processing interval data using the Itron Adapter
Itron - Scalar IMD Type (D8-SCA-IMDTYPE)	Used when processing scalar data using the Itron Adapter
ADK - Interval IMD Type (DG-INT-IMDTYPE)	Used when processing interval data using the Adapter Development Kit
ADK - Scalar IMD Type (DG-SCA-IMDTYPE)	Used when processing scalar data using the Adapter Development Kit

Changing and Extending Base Package IMD Types

The base package IMD types can be changed and/or extended by adding new algorithms to one or more of the algorithm areas (Pre-Processing, Pre-VEE, VEE, or Post-Vee).

When adding new algorithms to a base package IMD type, we recommend creating the algorithm logic in a Groovy Library Script that can be called from an IMD type algorithm as well as from an IMD business object algorithm. See Groovy Library Scripts for more information.

Configuring IMD Types

You use the **IMD Type** portal to display and maintain IMD types.

Refer to Understanding IMD Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **IMD Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **IMD Type List**: This zone lists all division records. Broadcast a record to display the details of the selected record.
- **IMD Type**: This zone provides information about the selected division.

Defining VEE Options

- Exception Types
- VEE Groups
- VEE Rules
- Measuring Component Statistics

Copyright © 2026, Oracle and/or its affiliates.

Exception Types

- Understanding Exception Types
- Configuring Exception Types

Understanding Exception Types

Exception Types define the groupings of exceptions for an IMD based on their functional similarity. This provides a way to define VEE Exceptions in a distinct enough way to understand the root issue that was generated from the VEE Rule.

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

Configuring Exception Types

This portal is used to display and maintain an Exception Type.

Refer to Understanding Exception Types for more information.

You can access the portal by selecting **Admin**, then **VEE**, then **Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Exception Type List**: displays all of the Exception Types so you can choose the one you want to display in more detail
- **Exception Type**: shows the specific configuration for the selected Exception Type

There are two different options to use when creating an Exception Type:

Name	Details	Business Object
VEE Exception	This will create a "normal" VEE Exception that is attached to an IMD for tracking of conditions triggered in the VEE process.	D1-VEEException
VEE Exception - Monitor Service Point	In addition to tracking the failure in the VEE process, this VEE Exception generates a Service Issue Monitor. This allows for cumulative tracking of VEE Exceptions that can be configured to result in a Service Investigative Order (field work) for the Service Point.	D2-VEEExceptionServiceMonitor

VEE Groups

- Understanding VEE Groups
- Configuring VEE Groups

Copyright © 2026, Oracle and/or its affiliates.

Understanding VEE Groups

VEE groups are collections of VEE Rules that are applied to initial measurement data. During the VEE process, the system executes the VEE Rules defined in each VEE group. The rules within a VEE group are defined in a specific sequence, allowing control over the order in which the rules are executed.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component. VEE groups associated to a measuring component override those assigned to a measuring component type.

VEE groups can also be referenced by the Execute VEE Group VEE Rule.

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

Post-ML VEE Groups

When using Machine Learning Anomaly Scoring, you can define a VEE Group that is executed after the machine learning processing. These are used to execute rules that compare or adjust the measurement values based on a related measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the "Post-ML" VEE Role on measuring component types and measuring components.

Post-ML VEE Groups can only include the following VEE Rules configured for Post-ML:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement

> **Note**
>
> Post ML VEE Groups can NOT reference Execute VEE Group or VEE Group Matrix rules.

Configuring VEE Groups

This portal is used to display and maintain a VEE Group.

Refer to Understanding VEE Groups for more information.

You can access the portal by selecting **Admin**, then **VEE**, then **VEE Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **VEE Group**: Defines basic information about VEE group
- **VEE Rules List**: lists the VEE Rules belonging to the group

Copyright © 2026, Oracle and/or its affiliates.

- **Referencing VEE Rules List**: lists the VEE Rules that reference the group
- **Referencing VEE Group Factors List**: lists the VEE group factors that reference the group
- **Referencing Measuring Component Type List**: lists the measuring component types that reference the group
- **Referencing Measuring Component List**: lists the measuring components that reference the group

VEE Rules

- Understanding VEE Rules
- Configuring VEE Rules

Understanding VEE Rules

VEE Rules are standard and custom Validation and Estimation rules that perform checking and/or manipulation of initial measurement data. VEE Rules are created for a specific VEE group. For example, if you were configuring two VEE groups and both included a specific VEE Rule, you would need to create two instances of the VEE Rule, one for each group.

The specific validation and estimation processing performed on initial measurement data is defined in individual VEE Rules, each performing a specific set of targeted logic. The base product contains many VEE Rules you can use in your implementation, but you can also create your own custom VEE Rules.

Some VEE Rules generate VEE Exceptions if the initial measurement data fails the conditions specified for the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE Rules change the Post-VEE quantities of initial measurement data, but VEE Rules can change anything on an initial measurement.

Every VEE Rule has an effective period. Rules will only be applied if the initial measurement's start date is within the rule's effective period. For example, an Interval Spike Check rule with a Start Date of 11/15/2010 will only be applied if the start date of the initial measurement is on or after 11/15/2010.

This allows you to update the specifics of a rule without removing the previous version of the rule. For example, you might change the tolerance of an Interval Spike Check rule from 1.2 to 1.5 as of a certain date. However, for initial measurement data for the period prior to the change, you would want to use the tolerance for the original version of the rule (1.2) instead of the new tolerance (1.5).

On almost every VEE Rule, the failure of the rule results in a VEE Exception and the Exception Type for the failure can be configured on the rule. These Exception Types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the initial measurement data to be put into the exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the initial measurement data from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one

Copyright © 2026, Oracle and/or its affiliates.

issue exists after all rules have been applied, the initial measurement data is transitioned to the exception state

- **Terminate**: Used to report a severe issue that will cause the VEE process to stop and the initial measurement data to be transitioned immediately to the exception state. Only one terminate exception can be issued (as the first one causes VEE processing to stop on an initial measurement data).

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

Configuring VEE Rules

This portal is used to display and maintain a VEE Rule. Also, a list of the specific out-of-the-box rules is included below the instructions for using the portal.

Refer to Understanding VEE Rules, Understanding VEE Groups, and Understanding Exception Types for more information. For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

You can access the portal by selecting **Admin**, then **VEE**, then VEE Rule. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **VEE Rule**: this zone displays all of the configuration items specific to this instance of a VEE Rule.
  - Click **Show All Rules** to open the **VEE Rules** List zone, which lists all of the VEE rules in the VEE group to which the current rule belongs.
- **Eligibility Criteria List**: this zone displays any VEE Rule Eligibility Criteria that have been setup. These eligibility criteria determine conditionally whether the VEE Rule should be executed or not. Use the Add button to create a new eligibility criteria for the rule you're viewing.

Validation VEE Rules

Below is a list of the validation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Applicable Data Type(s)	Purpose
Consecutive Interval Check	Interval	This validation rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data.
Duplicate IMD Check	Interval or Scalar	This rule allows for a duplicate IMD to be flagged.

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Comparison Validation	Interval or Scalar	This powerful, flexible validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Then you define formulas (no programming required) for the comparison of current measurements to these statistics.
Ensure IMD Exists for Sibling MCs	Interval or Scalar	This rule validates that an IMD exists for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time.
Final Measurement Replacement	Interval or Scalar	This validation rule allows you to define a variety of configuration options to decide if new data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking.
High/Low Check	Interval or Scalar	This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown.
Inactive Measurement Check	Interval or Scalar	This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a Usage Subscription.
Interval Size Validation	Interval	This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type.
Interval Spike Check	Interval	This rule checks for spikes within an IMD and generates an exception if one is found.

Copyright © 2026, Oracle and/or its affiliates.

Multiplier Check	Interval or Scalar	This rule validates that the register multiplier supplied with the IMD is equal to the multiplier stored on the measuring component. If not, an exception is created using the register multiplier exception type and severity configured on the rule.
Negative Consumption Check	Interval or Scalar	This rule flags any IMD where the total consumption is negative.
Prolonged Estimation Check	Interval or Scalar	This validation rule creates an alert when a device has been estimated for an extended period of time.
Raise Missing Quantity Exception	Interval	This rule flags any missing interval data.
Sum Check	Interval or Scalar	This rule is used to compare the difference between interval data to scalar data for a period of time, or between a set of TOU scalar reads to a "Sum" scalar reading.
Unit of Measure Check	Interval or Scalar	This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type.
Zero Consumption Check	Interval or Scalar	This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid exceptions in that case.

Estimation VEE Rules

Below is a list of the estimation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Applicable Data Type(s)	Purpose
Interval Adjustment From Scalar	Interval	This rule performs adjustments to interval values based on the values from the associated scalar data on the same device.

Copyright © 2026, Oracle and/or its affiliates.

Interval Averaging Estimation	Interval	This rule finds historical interval data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD.
Interval Create Estimation IMD for Gap	Interval	This estimation rule creates a new Estimation IMD to fill gaps between the interval data received from the Head End. ***Note:*** *this estimation rule acts differently from the other rules as it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.*
Interval Interpolation Estimation	Interval	This rule attempts to interpolate gaps within an IMD using prior and subsequent intervals as starting points for linear interpolation.
Interval Profile Estimation	Interval	This rule estimates any missing interval data for the IMD based on a referenced Profile Factor.
Interval Proxy Day Estimation	Interval	This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement.
Scalar Calculation From Interval	Scalar	This rule performs adjustments to scalar values based on the values from the associated interval data on the same device.
Scalar Estimation	Scalar	This rule finds historical scalar data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD.
Scalar Profile Estimation	Scalar	This rule estimates any missing scalar data for the IMD based on a referenced Profile Factor.
Scalar Proration	Scalar	This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries. It will also take into account any related interval data within the same period to exclude from the calculation.

Copyright © 2026, Oracle and/or its affiliates.

Subtractive Interval Adjustment Rule	Subtractive Interval Adjustment Rule	This rule performs adjustments to qualifying interval consumption values for subtractive interval measuring components based on an adjustment target calculated using a start and stop reading for the period that encapsulates the intervals to adjust.

Decision-Making VEE Rules

There are VEE Rules delivered as part of the base product that help with decision-making when executing VEE (listed below). For more information on how each rule executes and can be configured, follow the link provided on the rule.

VEE Rule Name	Purpose
Exception Handler	This rule allows for termination of the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions.
Execute VEE Group	This rule performs a call to execute a separate VEE Group which includes execution of all VEE Rules within that group.
Successful Termination	This rule allows VEE to be successfully terminated based on a list of exceptions.
VEE Group Matrix (Factor)	This rule provides a way to choose different instances of a VEE Rule using a Factor. This factor leverages characteristics that are defined on Service Point, Device, or Measuring Component.

- Validation VEE Rules
- Estimation VEE Rules
- Decision-Making VEE Rules

Validation VEE Rules

- Consecutive Interval Check
- Duplicate IMD Check
- Dynamic Comparison Validation
- Ensure IMD Exists for Sibling MCs
- Final Measurement Replacement
- High/Low Check
- Inactive Measurement Check
- Interval Size Validation
- Interval Spike Check

Copyright © 2026, Oracle and/or its affiliates.

- Multiplier Check
- Negative Consumption Check
- Prolonged Estimation Check
- Raise Missing Quantity Exception
- Sum Check
- Unit of Measure Check
- Zero Consumption Check

Consecutive Interval Check

Overview

This rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data. It can be used to find faulty meters that are reporting consecutive outage codes, zero measurements, or negative values. It can also be used by water utilities to identify leaks based on the interval never reaching zero.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CONSINTRV Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

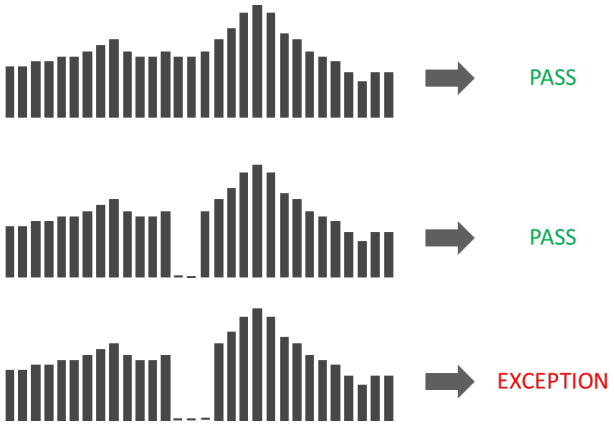
D2-ConsecutiveIntervalCheck

Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

Scenario 1: Rule configured to fail for zero values

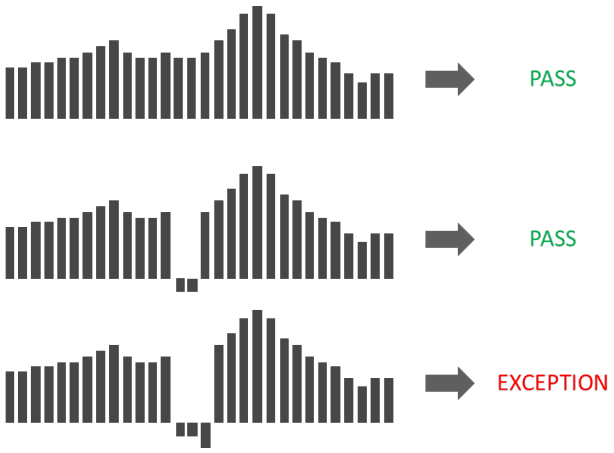
Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	Equal To
Value for Comparison	0
Bottom Range Condition	
Top Range Condition	



Scenario 2: Rule configured to fail for negative values

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	Less Than
Value for Comparison	0
Bottom Range Condition	
Top Range Condition	



Scenario 3: Rule configured to fail for missing condition codes

Parameter Name	Configuration Value
Check performed	Consecutive Condition Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	
Value for Comparison	
Bottom Range Condition	Missing
Top Range Condition	Missing



Scenario 4: Rule configured to fail for outage condition codes only if its zero

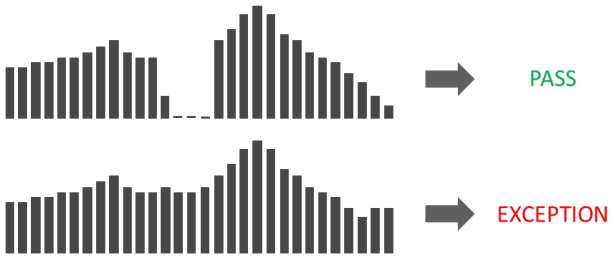
Parameter Name	Configuration Value
Check performed	Both
Maximum consecutive hours	02:00:00
Look To Prior Measurements	No
Equation Type	Equal To
Value for Comparison	0
Bottom Range Condition	Outage
Top Range Condition	Outage



Scenario 5: Rule configured to fail if a long period of consecutive water values are greater than 0.1

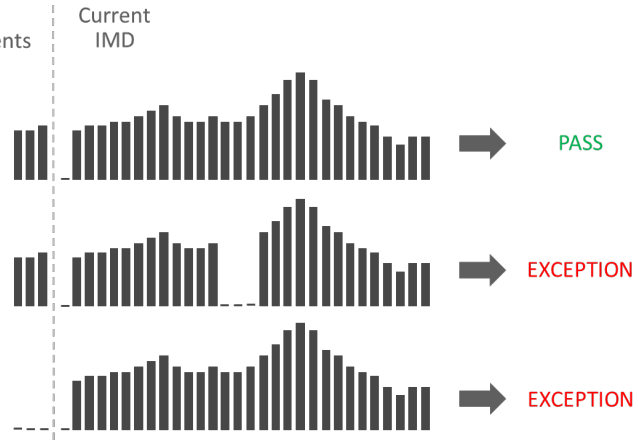
Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	20:00:00
Look To Prior Measurements When First Interval Matches Criteria	No
Equation Type	Greater Than
Value for Comparison	0.1
Bottom Range Condition	
Top Range Condition	



Scenario 6: Look back at prior measurements when the first interval matches criteria

Parameter Name	Configuration Value
Check performed	Consecutive Measurement Values
Maximum consecutive hours	02:00:00
Look To Prior Measurements	Yes
Equation Type	Equal To
Value for Comparison	0
Bottom Range Condition	
Top Range Condition	



Duplicate IMD Check

If the current Initial Measurement Data (IMD) being validated is determined to be a duplicate of an existing IMD for the same measuring component, this rule will produce a VEE exception of the type and severity configured on the VEE Rule. The algorithm logic looks for duplicate IMDs using the following criteria:

- associated to the same measuring component as the current IMD
- utilizes the same IMD business object as the current IMD (for example, Initial Load Scalar)
- references the same To Date/Time (ends on the same date) as the current IMD
- exists in a "Finalized" state
- the contents of pre-VEE are identical to the pre-VEE of the current IMD

If any IMDs are found that meet all of the above criteria, the current IMD is deemed to be a duplicate.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-DUPIMDCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Copyright © 2026, Oracle and/or its affiliates.

Business Object

D1-DuplicateIMDCheck

Dynamic Comparison Validation

This validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Setting up these Measuring Component Statistics is a prerequisite to using the rule.

Users can define formulas (no programming required) for the comparison of current measurements to these statistics. This is powerful rule will allow utilities to look for unusual usage patterns. For example:

- Lowest/highest usage ever
- Current usage is more than three standard deviations from the mean
- Detect unexpected zero usage
- Detect negative usage while ruling out known cases
- Abnormal usage
- Voltage threshold monitoring
-

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-DYNCOMCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

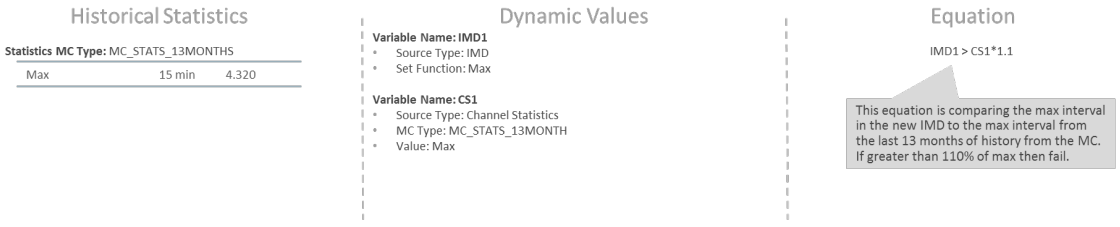
Business Object

D2-DynCompValidation

Example Scenarios

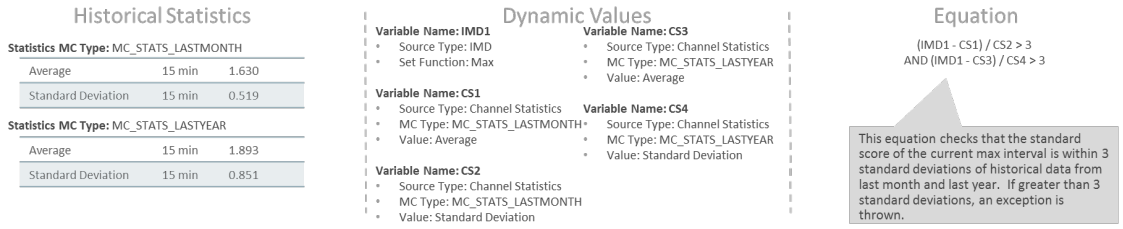
Below are some example scenarios that can be achieved based on configuration of this rule.

Example 1: Detect new "high water mark"

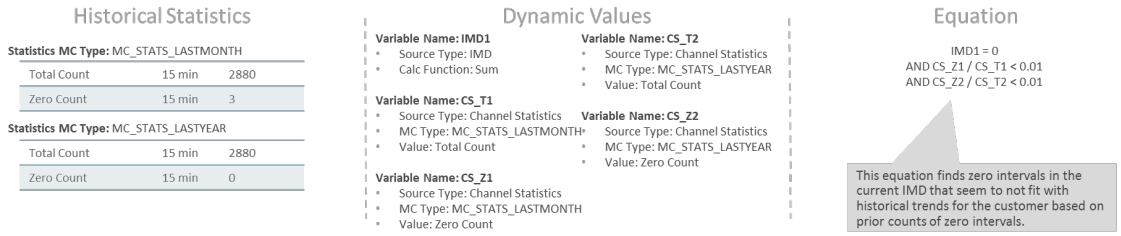


Example 2: High Usage

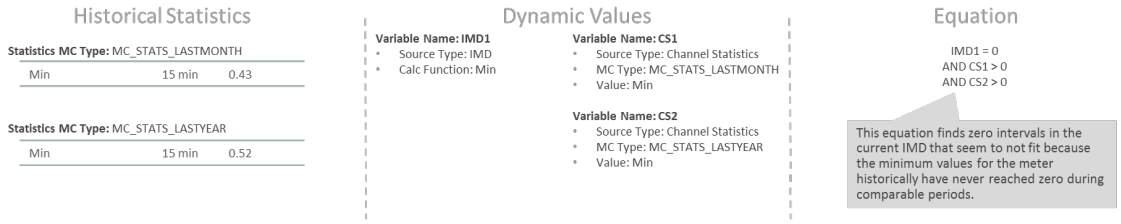
Copyright © 2026, Oracle and/or its affiliates.



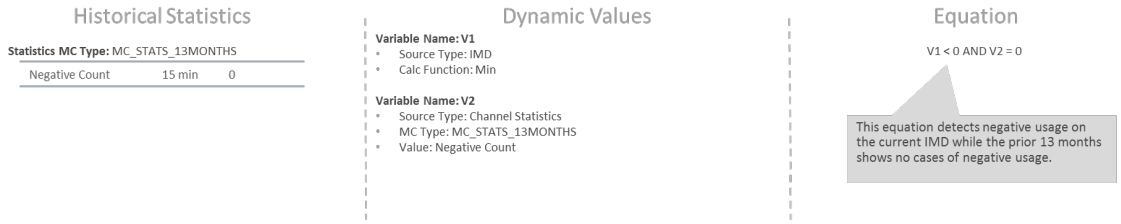
Example 3: Detect unexpected zero usage (method 1)



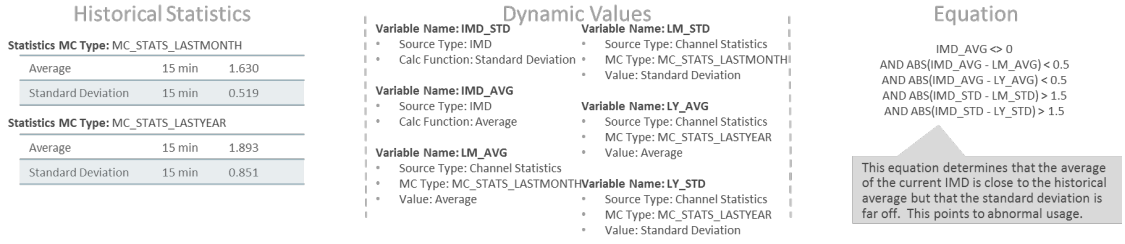
Example 4: Detect unexpected zero usage (method 2)



Example 5: Detect negative usage while ruling out known cases

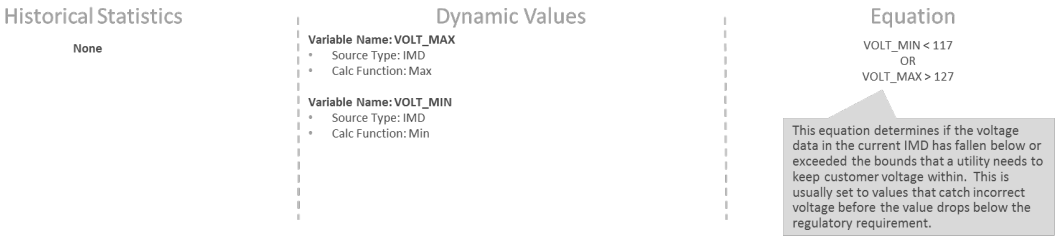


Example 6: Abnormal usage



Example 7: Voltage threshold monitoring

Copyright © 2026, Oracle and/or its affiliates.



Ensure IMD Exists for Sibling MCs

This rule validates that an IMD exist for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time.

A check is also performed that all of the Initial Measurements have the same External Source ID (indicating that they all came from the same usage file).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ENSIMDMC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-EnsureIMDExistsForSibling

Final Measurement Replacement

This validation rule allows you to define a variety of configuration options to decide if new scalar or interval data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking. One common use for this rule is rejecting trivial measurement changes to prevent very small changes for a bill from being sent to a customer.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-VLMSRCOND Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

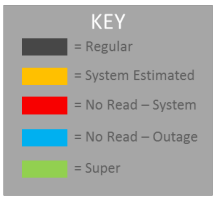
D2-FinalMeasurementValidation

Example Scenarios

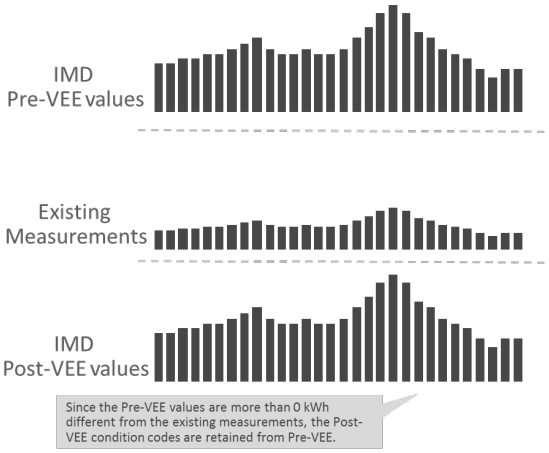
Copyright © 2026, Oracle and/or its affiliates.

Below are some example scenarios that can be achieved based on configuration of this rule.

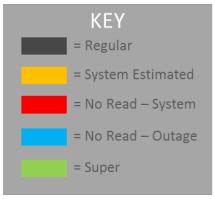
Example 1: Replacement of measurements based solely on value difference



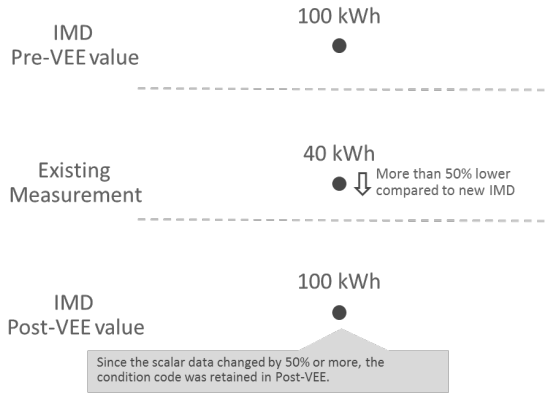
Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	Compare Value Change
Value Change Tolerance	0.1
Percentage Change Tolerance	
Condition Code Prioritization by Groups	



Example 2: Percentage change to replace measurement

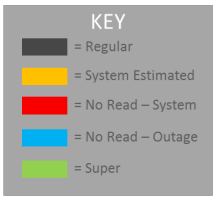


Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	Compare Percentage Change
Value Change Tolerance	
Percentage Change Tolerance	50
Condition Code Prioritization by Groups	

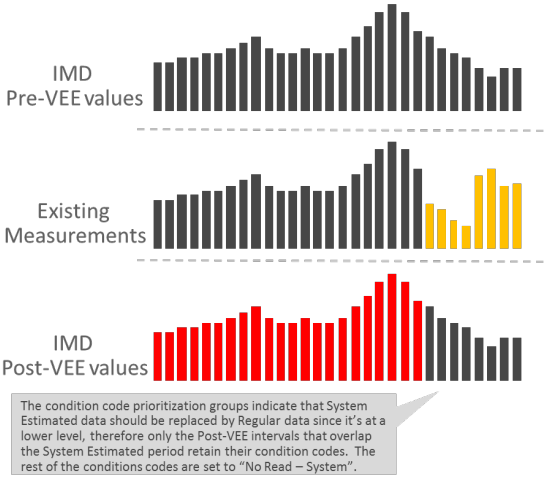


Example 3: Partial replacement of estimated data with regular data based on condition range prioritization

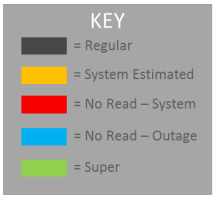
Copyright © 2026, Oracle and/or its affiliates.



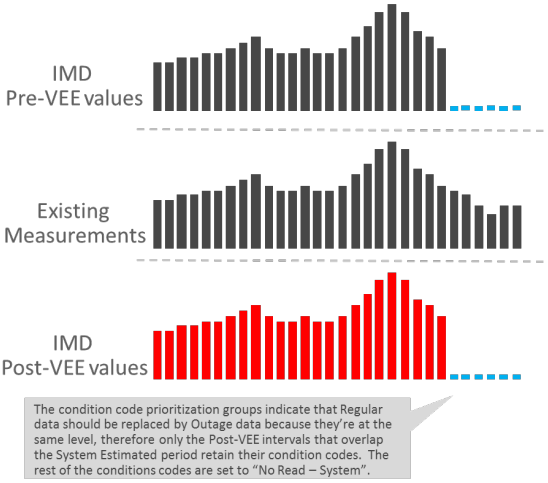
Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	Compare Value Change
Value Change Tolerance	0.1
Percentage Change Tolerance	
Condition Code Prioritization by Groups	10: Super 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System



Example 4: Partial replacement of regular data with outage data based on condition range prioritization

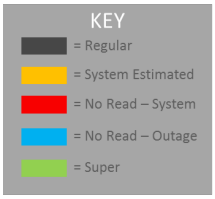


Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	Compare Value Change
Value Change Tolerance	0.1
Percentage Change Tolerance	
Condition Code Prioritization by Groups	10: Super 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System

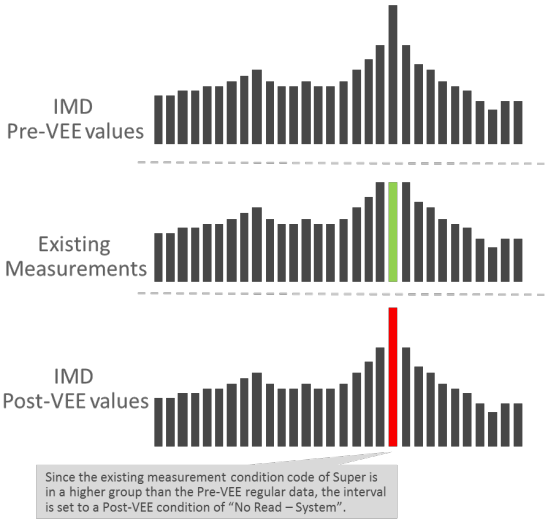


Example 5: Super not replaced by regular (retain a smoothed spike)

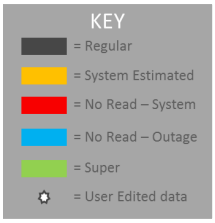
Copyright © 2026, Oracle and/or its affiliates.



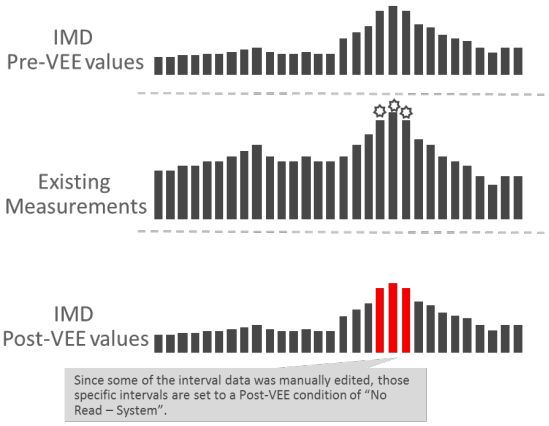
Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	
Value Change Tolerance	
Percentage Change Tolerance	
Condition Code Prioritization by Groups	10: Super 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System



Example 6: Prevent replacement of manually edited data

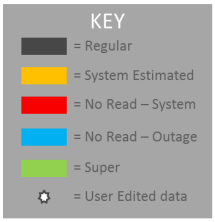


Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	No
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	
Value Change Tolerance	
Percentage Change Tolerance	
Condition Code Prioritization by Groups	

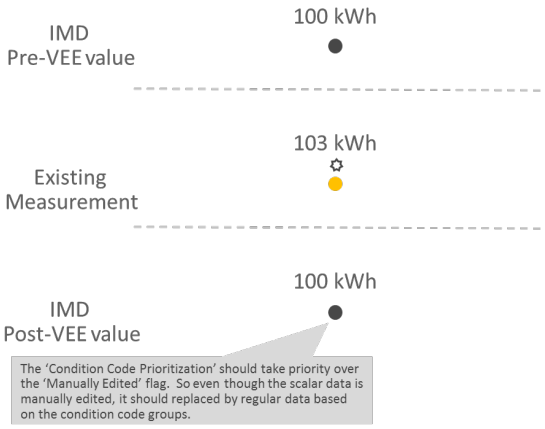


Example 7: Allow replacement of manually edited data if the condition group is lower than new data

Copyright © 2026, Oracle and/or its affiliates.



Parameter Name	Configuration Value
Replacement Method	Reject Based on Configuration
Replacement Condition Value	No Read – System
Replace Manually Edited Data?	No
Use Early and Late MC Type Settings?	
Replacement Evaluation Type	
Value Change Tolerance	
Percentage Change Tolerance	
Condition Code Prioritization by Groups	10: Super 20: Regular, No Read – Outage 30: System Estimate 40: No Read – System



High/Low Check

This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown.

Numerous configuration options are provided including:

- A Historical Pre-Window and Post-Window that determines the number of days to check before and after the period being examined.
- Percentages to control how much of the historical data can be user edited, estimated, or non-normal data.
- What type of historical data to look at first
- What to do when an outage has occurred

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-HILO-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-VEERuleHighLowCheck

Inactive Measurement Check

This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a usage subscription.

Copyright © 2026, Oracle and/or its affiliates.

- When the 'Check Measurements on Disconnected Device' flag is set to "Yes", the logic will compare the IMD measurements total to the threshold configured. If the total measurements are above the threshold, then logic compares the IMD dates to two things: 1) On/Off Dates related to the Install Event and 2) the Installation and Removal Dates of the Install Event. If the On or Off falls within the dates for an interval IMD, then the logic will only sum intervals that fall during the disconnected periods.
- When the 'Check for Uninstalled Device' flag is set to "Yes", the logic will check for a valid Install Event based on the IMD dates.
- When the 'Check for Missing Usage Subscription' flag is set to "Yes", the logic will check for an active Usage Subscription based on the IMD dates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INACTVCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

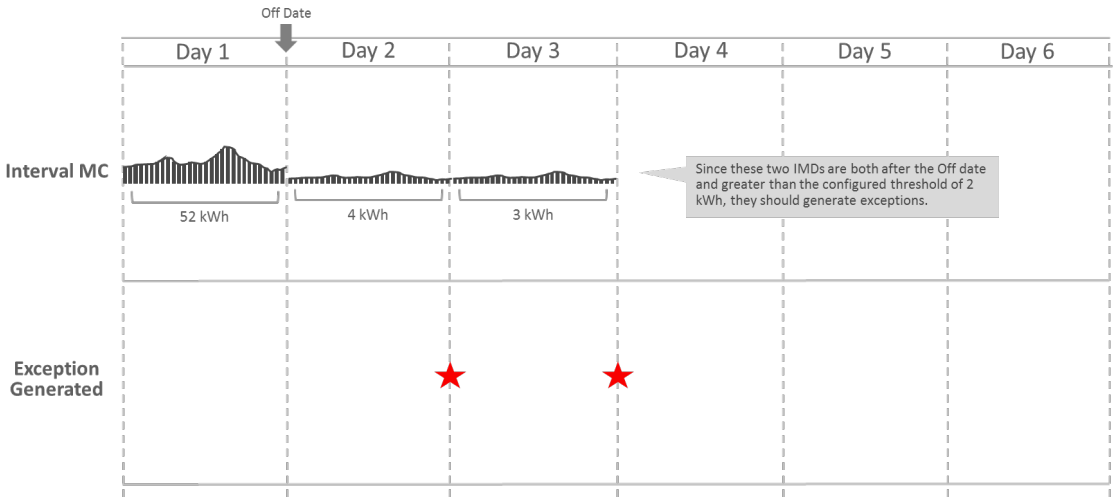
Business Object

D2-InactiveMeasurementCheck

Example Scenarios

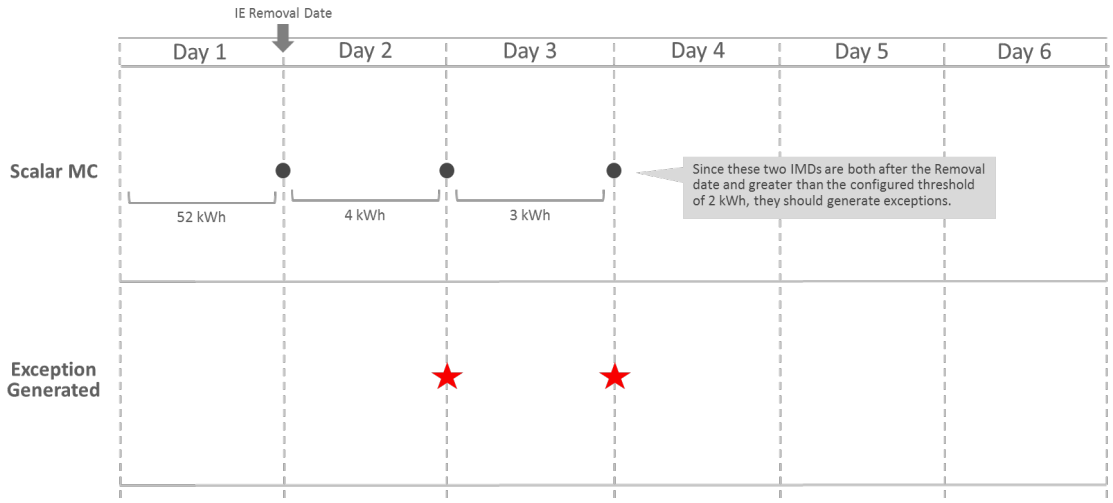
Below are some example scenarios that can be achieved based on configuration of this rule.

Example 1: Non-zero data during disconnected period

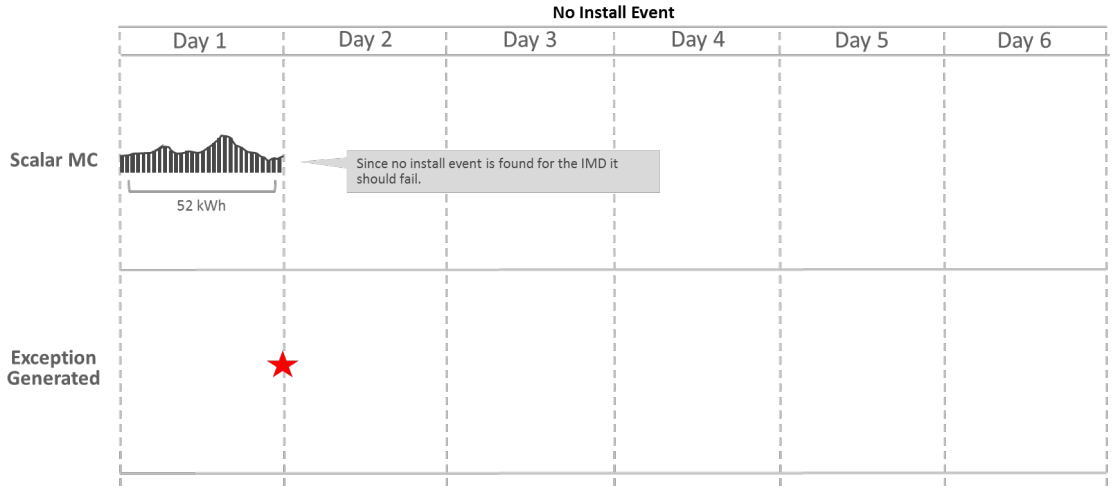


Example 2: Non-zero data after removal date with no new install

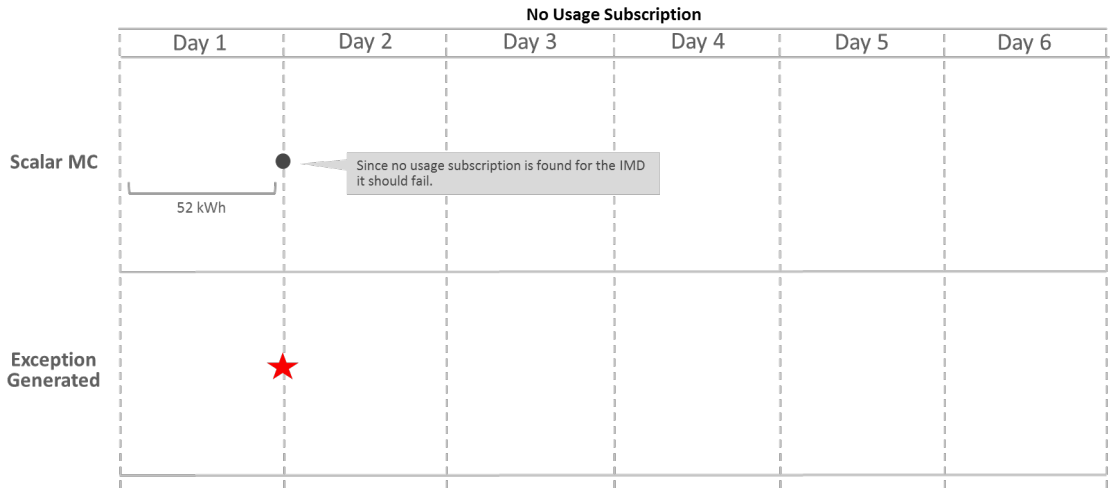
Copyright © 2026, Oracle and/or its affiliates.



Example 3: Data received when device has no Install Event records



Example 4: Data received when not connected to a Usage Subscription record



Interval Size Validation

Copyright © 2026, Oracle and/or its affiliates.

This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTSIZVAL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalSizeValidation

Interval Spike Check

This rule looks for spikes by taking the highest interval and the third-highest interval, and determining the percent difference between the two. If the percent difference is larger than the tolerance configured on the rule, the algorithm logs an exception of the type and severity configured on the rule.

The rule can be executed in one of two modes (as configured on the rule):

- The spike check is performed for every 24-hour chunk of data in the initial measurement data set
- The spike check is performed for the entire initial measurement data set

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTSPKCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalSpikeCheck

Multiplier Check

This rule validates that the register multiplier supplied with the Initial Measurement is equal to the multiplier stored on the measuring component.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-REGMULCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-RegisterMultiplierCheck

Negative Consumption Check

This rule checks if the total consumption of the initial measurement is less than zero. If the rule encounters negative consumption, an exception will be logged if the related Measuring Component Type is not configured to "allow negative consumption", or if the Override MC Type Allow Negative Consumption flag on the rule is set to “Yes”.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-NCON-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-NegativeConsumptionCheck

Prolonged Estimation Check

This validation rule can be used on either interval or scalar and will alert you when a device has been estimated for an extended period of time. The IMD must first fall within the condition code range configured on the rule for this to execute. Next the validation finds the 'Most Recent Non-Estimated Read Date Time' from the Measurement Component. If this date plus the 'Days of Estimation Allowable' from the VEE Rule is less than the IMD End Date then a Prolonged Estimation Exception is created.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-PROESTCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

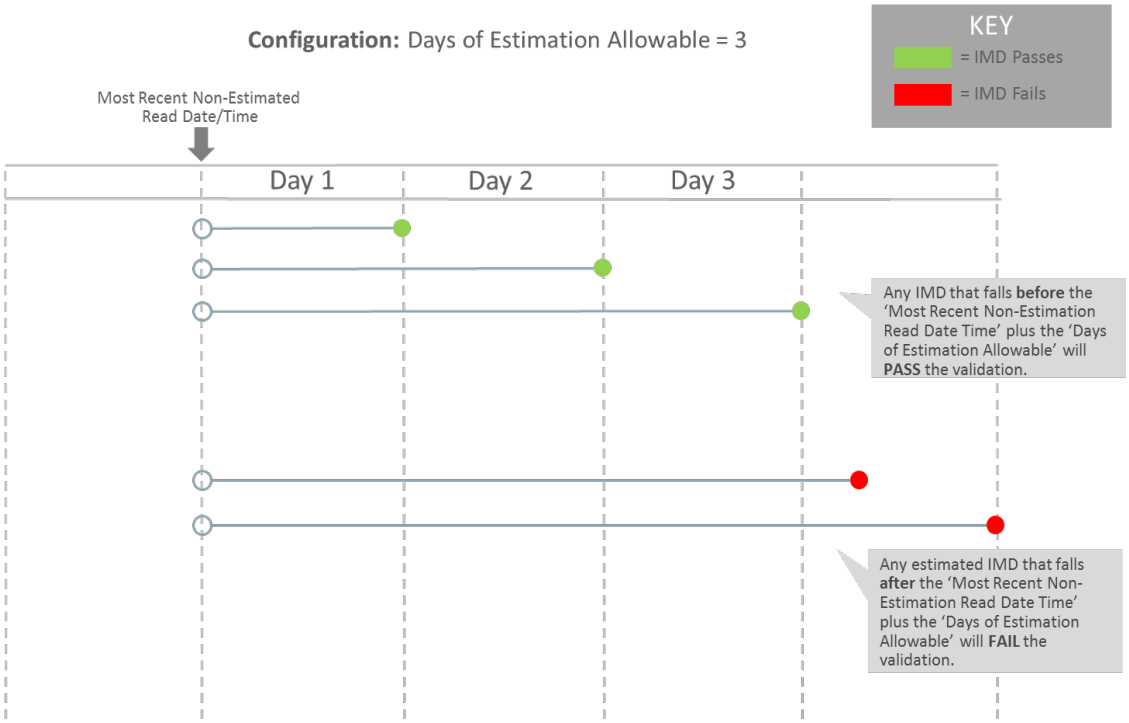
Business Object

Copyright © 2026, Oracle and/or its affiliates.

D2-ProlongedEstimationCheck

Example Scenario

Below is an example scenario that can be achieved based on configuration of this rule.



Raise Missing Quantity Exception

This rule flags any missing interval data while providing a soft parameter on the algorithm to exclude a condition range if desired.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-RAIMISQTY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-RaiseMissingQuantityExcp

Sum Check

This rule evaluates whether consumption for the current Initial Measurement Data is within a tolerance of the sum of the consumption during the same period for any measuring components related to the current one. If the values are not within the defined tolerance of each other, and exception is logged.

Copyright © 2026, Oracle and/or its affiliates.

The rule can be used to evaluate consumption totals for an interval measuring component that has a related scalar measuring component with the same UOM to ensure that the total consumption of the interval measuring component is within a tolerance of that of the scalar value. It can also be used to evaluate consumption totals for scalar TOU meters that have a "check" register (for example, three registers that measure ON-PEAK, OFF-PEAK, and SHOULDER, with a fourth check register that measures the total consumption).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUM-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-SumCheck

Unit of Measure Check

This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-UOMCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-UOMCheck

Zero Consumption Check

This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid To Dos by having a different exception type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ZEROCNCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Copyright © 2026, Oracle and/or its affiliates.

Business Object

D2-ZeroConsumptionCheck

Estimation VEE Rules

- Interval Adjustment From Scalar
- Interval Averaging Estimation
- Interval Create Estimation IMD for Gap
- Interval Interpolation Estimation
- Interval Profile Estimation
- Interval Proxy Day Estimation
- Scalar Calculation From Interval
- Scalar Estimation
- Scalar Profile Estimation
- Scalar Proration
- Subtractive Interval Adjustment Rule

Interval Adjustment From Scalar

This rule adjusts interval Initial Measurement Data based on scalar values using one of two options configured on the VEE Rule. Both options require that a scalar measuring component be related to the current interval measuring component using the Consumption Reference Measuring Component, and that one or more final measurements be present for the related measuring component between the start and end date/times of the current Initial Measurement Data set.

1. Adjust all intervals. In this case, the scalar consumption provides a value that is then used to proportionally adjust all of the intervals in the set. The formula used is (Scalar Consumption / Total Initial Measurement Consumption) * Interval Amount. If the total of all of the intervals had been equal to zero originally, the rule adjusts all of the intervals to the same value.
2. Adjust intervals based on condition. For this option, a range of conditions is configured on the VEE Rule, and the rule adjusts only those intervals with conditions that lie within the configured range. The adjustment is done by applying an adjustment factor to each of the intervals within range.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTADJSCA Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalAdjustmentFrmScalar

Copyright © 2026, Oracle and/or its affiliates.

Interval Averaging Estimation

The algorithm estimates when the current Measuring Component is eligible. Estimation is performed only if all these conditions are satisfied:

- The Measuring Component is interval.
- The Measuring Component is linked to a Service Point or the VEE Rule's Estimate If Not Attached to SP field is set to "Estimate".
- The Percentage of Missing Intervals is less than the VEE Rule's Maximum Percentage Missing Intervals Threshold.

The algorithm will compute the Total Accumulated Consumption and the Total Number of Intervals using a variety of methods including scanning holidays, similar days of the week, or neighboring days. Once a set of valid comparison days are found then these are used as the basis of creating estimated intervals for the current IMD.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTAVGEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalAveragingEstimation

Interval Create Estimation IMD for Gap

This rule is very different from other VEE rules in that it does not examine the current initial measurement but rather looks to see if there are any missing measurements prior to the initial measurement being processed (i.e. a gap exists). If so and the scenario meets the configured options it will generate an estimation initial measurement to fill in that gap.

This rule is intended to provide more real-time filling of missing measurements as opposed to running periodic estimation. However, it is still expected that periodic estimation will be used in conjunction with this rule such that any gaps that are not filled in by this rule would eventually be filled in by periodic estimation.

This rule can be configured to perform minimal validation of that gap that is identified and defer to the estimation initial measurement to validate against other initial measurements and final measurements that may overlap the gap. Conversely, it can also be configured to validate the gap and exclude any periods where a final measurement or in progress initial measurement overlaps the gap's duration.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This estimation rule acts differently from the other rules - it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CREESTIMD Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

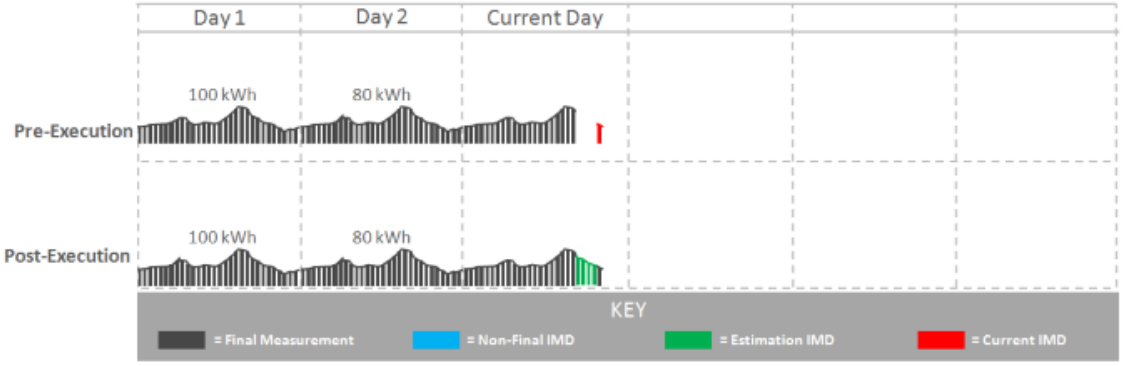
Business Object

D2-CreateEstimationIMDRule

Example Scenarios

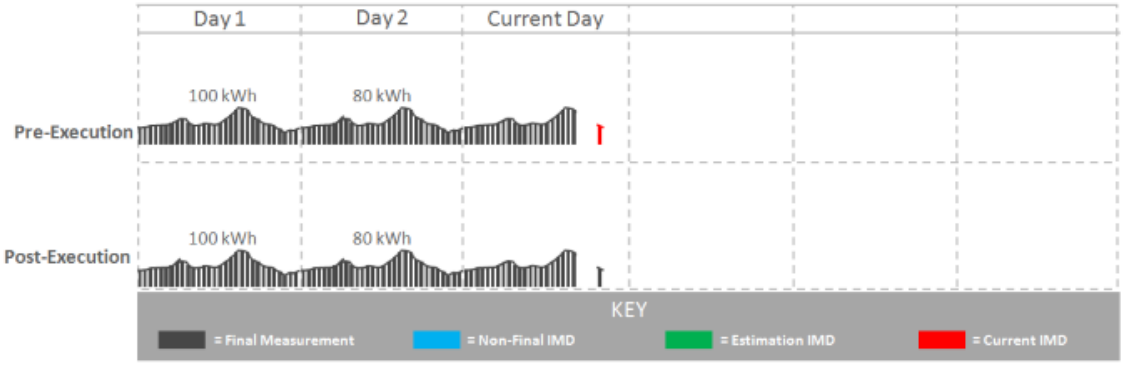
The following are sample scenarios that can be achieved based on configuration of this rule.

Example 1: A 4-hour gap exists and the rule is configured to always fill gaps. Maximum hours are set to 4 and the gap fill is set to "always".



Since the maximum hours has not been exceeded and the rule is configured to always fill the gap, a 4-hour estimation IMD is created to fill the gap.

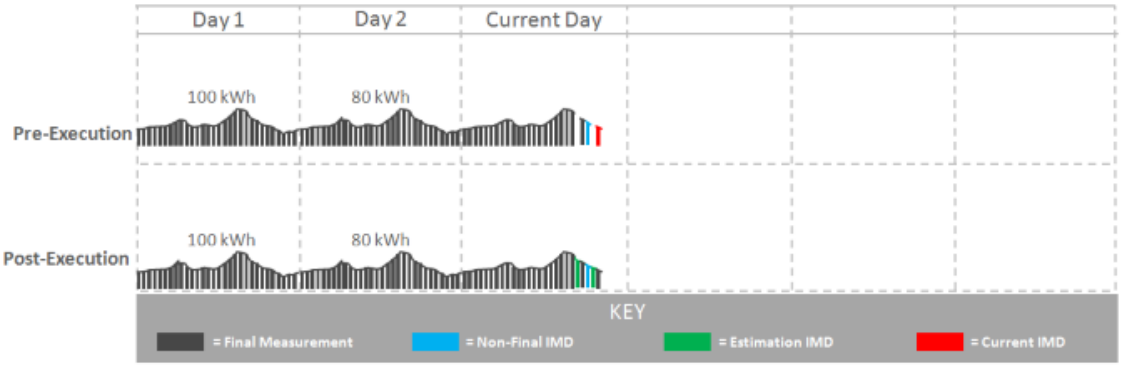
Example 2: A 4-hour gap exists and the rule is configured to always fill gaps. However, the maximum number of hours to fill is 3.



Since the gap is 4 hours long and the maximum hours to gap fill is 3 hours, no estimation has been created. An exception would be created if the rule were configured this way.

Copyright © 2026, Oracle and/or its affiliates.

Example 3: A non-contiguous gap exists and the rule is configured to "Skip In-Progress IMDs and Final Measurements" with a maximum number of IMDs set to 2.



In this scenario:

- One interval in the 4-hour gap is covered by a final measurement.
- One interval in the 4-hour gap is covered by an in-progress IMD.
- Two estimation IMDs are generated one to fill each hour not covered by a final measurement or in-progress IMD.

Interval Interpolation Estimation

This rule attempts to interpolate gaps in Initial Measurement Data sets using prior and subsequent intervals as starting points for linear interpolation. A configuration parameter determines the maximum number of consecutive intervals within a gap before that gap can no longer be interpolated.

If a given gap lies at the beginning or end of the Initial Measurement Data set, this algorithm seeks out final measurements immediately before or after the set in an attempt to find two reference measurement values for interpolation. In this situation, assuming that the condition of the measurement retrieved lies in NEITHER of the algorithm parameter ranges you have configured (the measurement is neither "missing" nor an "outage"), this measurement value is then used as the basis for interpolation.

In the event that the gap is the entire length of the Initial Measurement Data set (and the VEE Rule is configured such that this is not too large of a gap), the routine attempts to find final measurements as described above.

If a valid measurement can be found for only one side of a given gap, the interpolation logic assigns each interval in the gap the same value - the value of the measurement retrieved. This is referred to as applying a "flat load".

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTINTEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

Copyright © 2026, Oracle and/or its affiliates.

D2-IntervalInterpolationEst

Interval Profile Estimation

This rule uses a profile measuring component's interval consumption as a source of values to assign to intervals in the current Initial Measurement Data set that are marked "missing" via the interval's condition. If the interval is marked with a condition indicating "outage", the algorithm skips it.

If a measurement is not available for the profile measuring component on the date/time, the interval is left unchanged.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTPROEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalProfileEstimation

Interval Proxy Day Estimation

This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement. Measuring component comparison periods are used as the main driver for determining similar temperature days.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTPROXY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalProxyDayEstimation

Scalar Calculation From Interval

This rule calculates a single consumption amount for a scalar measuring component's Initial Measurement using the total consumption for the same date/time range for a related interval measuring component.

The scalar value replaces any existing value within the Initial Measurement (in the post-VEE list) and updates the condition to the value configured on the VEE Rule. This VEE Rule

Copyright © 2026, Oracle and/or its affiliates.

updates the condition to the value defined in "Condition Value for High Quality Condition" when the measurement condition value for all underlying interval data and the previous scalar measurement (when Evaluate Condition Of Previous Scalar Measurement is configured as "Yes") is greater than or equal to the "Minimum Condition Quality to Override" (If "Minimum Condition Quality to Override" is defined) otherwise it updates the measurement condition of the Initial Measurement Data to the condition defined in "IMD Created Condition Value".

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCACALINT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ScalarCalcFromInterval

Scalar Estimation

This rule uses historical data for the same measuring component to derive an estimated value for a scalar Initial Measurement. Depending upon a VEE Rule configuration parameter, the routine uses historical data from a month ago followed by a year ago, or vice-versa. If the data for the first historical period turns out to be usable, the second historical period will not be evaluated. Whether historical data qualifies for use in estimation is configured via the rule parameters in conjunction with the below algorithm parameters.

The rule rejects consumption from a historical period as unusable for estimation if too great a portion of the period is covered by final measurements that are not high-quality. The first pair of parameter values are used when evaluating the acceptable System-Estimated Percentage for historical periods as configured on the VEE Rule. Once an estimated value is calculated, the routine backs into a reading (which involves backing out multipliers and, if the measuring component is subtractive, adding the result to the prior reading).

An interim High/Low can also be configured on this rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCALAREST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ScalarEstimation

Scalar Profile Estimation

Copyright © 2026, Oracle and/or its affiliates.

This VEE rule arrives at a scalar estimate by looking at final measurements for a profile measuring component covering the same date range as the current Initial Measurement. The profile measuring component to be used as a source of measurement data is defined via a profile factor on the rule. This rule is meant primarily for a configuration in which the profile measuring components are interval, although the profile could be scalar as well.

If a measurement already exists in the current Initial Measurement with a condition that lies within the ranges specified in the pairs of algorithm parameters for "system estimate" and "regular", the routine does not attempt to estimate.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCAPROEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ScalarProfileEstimation

Scalar Proration

This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries.

If the scalar MC does have a related interval MC and 'Minimum Related Measurement Condition' is defined the algorithm will take into consideration any intervals during the proration period that meet a minimum condition quality defined by the 'Minimum Related Measurement Condition' when performing the proration. This is to keep the prorated scalar estimates in sync with any existing interval measurements. To do this the algorithm will augment the total duration and total quantity being prorated by the related interval MC's qualifying measurements. For example, take the following scenario:

- Scalar proration is executing for a scalar IMD for 01/01 12:00 AM - 01/02 12:00 AM (24 hours)
- The subsequent measurement is for 150 kWh and has a measurement date/time of 01/04 12:00 AM (72 hours)
- The related interval MC has measurements on 01/03 12:00 AM to 01/04 12:00 AM totally 100 kWh (24 hours)

Therefore the result of the calculation for the scalar IMD from 01/01 12:00 AM to 01/02 12:00 AM would be 25 kWh. Since the interval measurements covered 24 of the 72 total hours of the 150 kWh that would leave 50 kWh to be split among the remaining 48 hours. Which leaves 25 kWh for 01/01 12:00 AM to 01/02 12:00 AM and 25 kWh for 01/02 12:00 AM to 01/03 12:00 AM.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCLRPRORT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ScalarProration

Subtractive Interval Adjustment Rule

This rule adjusts qualifying consecutive intervals for a subtractive interval initial measurement (IMD) based on the known consumption between a set of actual readings. The readings can either be leveraged from other intervals within the IMD or from the final measurements when there are no suitable readings within the IMD. The adjustment is done by applying an adjustment factor to each of the qualifying intervals.

The rule supports adjusting multiple sets of qualifying consecutive intervals by calculating a consumption adjustment target and adjustment factor specific to each set of consecutive qualifying intervals.

Intervals are considered to be qualified for adjustment based on condition. A range of conditions is configured on the VEE Rule, and the rule adjusts only those intervals with conditions that lie within the configured range.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUBINADJV Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-SubIntAdjustmentVEE

Decision-Making VEE Rules

- Exception Handler
- Execute VEE Group
- Successful Termination
- VEE Group Matrix (Factor)

Exception Handler

Copyright © 2026, Oracle and/or its affiliates.

This rule allows you to terminate the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-EEH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-VEERuleExceptionHandler

Execute VEE Group

Many implementations need to execute a group of validations for any data being received. For example, you might want to perform device identifier validations, multiplier checks, and UOM checks on all measuring components. One inefficient way to meet this requirement would be to repeat these three rules in multiple VEE groups. However, this solution becomes hard to maintain if changes to the rules are required (or if new "global rules" are introduced) as each group would have to be updated. Instead of this, you can use the "Execute VEE Group" rule to execute a referenced VEE Group. Using the example above, you could create a group called "Rules for All MCs" that contains a device identifier validations rule, a multiplier check rule, and a UOM check rule, and then reference the "Rules for All MCs" group in a Execute VEE Group rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-RFG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-VEERuleReferredVEEGroup

Successful Termination

This rule allows VEE to be successfully terminated based on a list of exceptions.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-EST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-VEERuleSuccessTermination

VEE Group Matrix (Factor)

Another situation likely to occur in many implementations is where specific rules may need to be applied to measurement data based on specific criteria, such as geography. For example, some geographic territories may have unique VEE Rules in addition to rules that are applied to all geographic territories. The "VEE Group Matrix (Factor)" rule allows for a Factor to determine which VEE Rule gets executed based on defined characteristics.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-FCT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-VEERuleGroupFactor

Prerequisite Setup

Since the prerequisite setup items for this rule are more involved, the following procedure describes these in detail:

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the VEE Groups to be associated to the characteristic values.
4. Create the Factor that will be referenced by the rule.
5. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/VEE group pairings.
6. Create the rule, referencing the factor

Measuring Component Statistics

Copyright © 2026, Oracle and/or its affiliates.

- Understanding Measuring Component Statistics
- Configuring Measuring Component Statistics

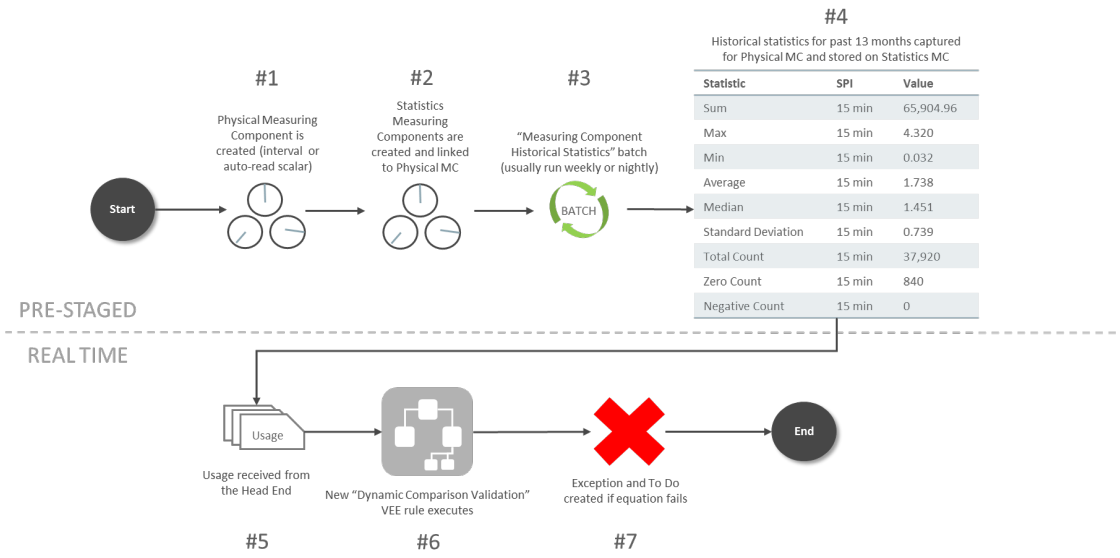
Understanding Measuring Component Statistics

Oracle Utilities Meter Data Management provides a way to stage statistics for Measuring Components in order to speed up VEE processing. These statistics are primarily meant for use in conjunction with the Dynamic Comparison Validation VEE Rule, but could also be generated for other analytics or reporting.

Some of the potential benefits of using Measuring Component Statistics are:

- Reduce the data volume for VEE queries by up to 1000 times (this is most beneficial when replacing VEE Rules that query historical data such as High/Low)
- Offload heavier processing of historical data to "system downtimes" - potentially nights or weekends

Below is an overview example of how Measuring Component Statistics can be implemented along with the Dynamic Comparison Validation:



The steps for pre-staging Measuring Component Statistics are detailed below:

1. A Physical Measuring Component is created. The algorithm to trigger the creation of Statistics Measuring Components is only set on Interval or Auto-Read Scalar BOs as part of base product.
2. New Statistics Measuring Component are created and linked to the Physical Measuring Component.
3. The Measuring Component Historical Statistics Batch (D1-MCHS) is run based on the schedule you've defined. The suggestion is to run this either nightly or on the weekends during quieter times for overall system processing.
4. A measurement is created for the Statistics Measuring Component that holds historical statistics related to the Physical Meter for a specific period of time (in this example it's set for 13 months).

The steps for how VEE can take advantage of Measuring Component Statistics are detailed below:

Copyright © 2026, Oracle and/or its affiliates.

1. IMDs are received from the Head End.
2. The Dynamic Comparison Validation VEE Rule is configured to execute for the meter. This validation uses the historical statistics from the Statistics Measuring Component in order to perform its validation logic.
3. An exception and potentially a To Do is generated based on the failure of the VEE rule.

Storage Impact from Statistics

Much consideration was given to the storage implications when designing Measuring Component Statistics. Below is an outline of the storage impact to customers who choose to calculate and store Channel Statistics. An example impact to a utility with 1 million Physical Meters (each with 4 channels of information) is shown in the far right column:

Area	Additional Data in MDM	Example Impact (assuming utility has 1 million Physical Meters)
Additional MCs *Assumption: Statistics are only configured on the primary billing channel MC Type for the majority of your Physical Meters.*	1-3 extra MCs per Physical Meter *Assumption: customers will only configure Statistics on their primary billing channel MC Type for the majority of their Physical Meters.*	1 - 3 million extra MCs *This increases the Measuring Component table by an extra 25% to 75% from its size without statistics.*
Additional Measurements *If 'Statistics Retention Mode' configured as "Only Retain Most Recent Record".*	1 extra Measurement per Physical Meter	1 - 3 million extra Measurements for the life of MDM *This data is a minuscule fraction compared to the overall AMI data footprint.*
Additional Measurements *If 'Statistics Retention Mode' configured as "Retain Historical Records".*	1 extra Measurement per Physical Meter per week	1 - 3 million extra Measurements per week (52-156 million per year) *This data is 0.2-0.5% of the size of the AMI data collected in that same week.*

Configuring Measuring Component Statistics

Measuring Component Types

In order to start generating Measuring Component Statistics, two different configurations must occur for Measuring Component Types:

- New Measuring Component Types must be configured that leverage the **Channel Statistics Type** business object. These should be setup as the periods for which Measuring Component Statistics should be calculated. For example, if you would like to be able to reference both last month, last year, as well as the past 13 months as statistics periods, then 3 different Channel Statistics Types must be setup for this. These could be setup as follows:
- – "Last Month" Channel Statistics Type:
  - – Calculation Lag Days = 0
  - – Calculation Horizon Days = 30

Copyright © 2026, Oracle and/or its affiliates.

- "Same Month Last Year" Channel Statistics Type:
- Calculation Lag Days = 365
- Calculation Horizon Days = 395
- "Prior 13 Months" Channel Statistics Type:
- Calculation Lag Days = 0
- Calculation Horizon Days = 395

NOTE: *it's recommended to set the Statistics Retention Mode to "Only Retain Most Recent Record" on the Channel Statistics Type to reduce the amount of data being stored in Oracle Utilities Meter Data Management.*

- For a Physical Measuring Component Type leveraging either "Interval Channel Type" or "Register - Auto-Read Type", the **Related Statistics Measuring Component Types** section must be filled in with the appropriate types that should be created for the Physical Measuring Component.

VEE Rule

If you have a desire to leverage Measuring Component Statistics for VEE, the Dynamic Comparison Validation VEE Rule section should be referenced. A number of detailed examples of how to implement this rule are provided in this section as well.

Related Batch Controls

There are a few batches directly involved with Measuring Component Statistics:

- **Measuring Component Historical Statistics** (D1-MCHS): this batch monitors any active Statistics Measuring Components to execute the logic for calculating a new set of historical statistics.
- **Statistics Measuring Component Creation** (D1-STMCC): this batch is used to create Statistics Measuring Components on any Physical Measuring Components that were created prior to configuring this module. This is especially useful for existing customers that want to implement Statistics Measuring Components for all of their devices.

> **Note**
>
> Refer to Understanding Measuring Component Statistics for more information.

Defining Usage Options

This section describes concepts and common tasks related to usage administration.

- The Big Picture of Usage Requests
- The Big Picture of Time of Use Mapping and Pricing
- Usage Subscription Types
- Integrating Usage Transactions
- Usage Transaction Exception Types
- Usage Calculation Groups

Copyright © 2026, Oracle and/or its affiliates.

- Usage Calculation Rules
- Usage Calculation Types
- Usage Subscription Quantity Types
- Dynamic Option Types
- Bill Cycle

The Big Picture of Usage Requests

Oracle Utilities Meter Data Management is used by some organizations who use a meter data management (mdm) system to record meter reading information. Since meter reads are not available in Oracle Utilities Customer Care and Billing to calculate consumption during the billing process, usage (or bill determinants) must be requested by Oracle Utilities Customer Care and Billing from the mdm system, Oracle Utilities Meter Data Management.

The following sections describe how the base-package usage request process works.

- Requesting Bill Determinants
- Usage Request Lifecycle
- Summary Billed Accounts and Usage Requests
- Corrected Read Notifications
- Configuring The System For Usage Request Integration
- Start and End Times for Billing

Requesting Bill Determinants

The term bill determinant request is another way to refer to a usage request. These usage requests are created during the batch billing process, and also when a user generates an online bill segment or performs cancel / rebill. The get consumption algorithm specified on the SA Type's Bill Segment Type is responsible for creating the usage request.

- Online Billing Usage Requests

Online Billing Usage Requests

- Usage requests created from an online billing request are not held in the initial state. These usage requests are transitioned through their lifecycle resulting in the request being sent to Oracle Utilities Meter Data Management (MDM) and, once a response is received, the bill segment is regenerated.
- The bill segment remains in the Freezable state until the user freezes the bill segment and completes the bill. If the freezable bill segment is regenerated, however, the system cancels the usage request and creates another usage request for the bill segment.
- In addition to the standard MDM usage request elements, the system captures the responsible user id.

Copyright © 2026, Oracle and/or its affiliates.

Usage Request Lifecycle

The usage request business object (BO) contains the rules that govern the processing of a usage request. The base product provides the BO C1-UsageRequest which serves as a parent BO and contains the following lifecycle:

- All usage requests are created in the initial Pending state. If the usage request was created from the batch billing process, the batch Usage Scheduled Monitor Process must be executed to transition the usage request.
- If there are pending sync requests for the usage request's service agreement, the usage request is held in the Awaiting Data Sync state until the sync is processed.
- Once all pending sync requests related to the usage request have been processed, the usage request transitions to the Send Request state. An enter algorithm on this transitory state is responsible for sending the usage request to Oracle Utilities Meter Data Management (MDM).
- The usage request sits in the Awaiting Bill Determinants state until a response is received from MDM, or a time out is encountered.
- When a response is received from MDM, the usage request is transitioned to either the Bill Determinants Received state, or the Error state. In the case of an error, the following occurs:
  - If the usage request was created from the batch billing process and the billing window is still open, a new usage request will be created the next time billing runs and the process is repeated. If however, the billing window is no longer open, a to do entry is created for manual follow up.
  - If the usage request was created from an online billing process, the user will be notified. The bill segment remains in the error state and the user can either delete the bill segment, or attempt to generate it again. If the bill segment is regenerated, a new usage request is created and the process is repeated.
  - The system transitions usage requests in the Bill Determinants Received state to the Bill Determinants Processed state. An enter algorithm on this final state is responsible for generating the bill segment. If the usage request was created from the batch billing process, the bill segment is frozen and the bill is completed. For online billing usage requests, the user is responsible for freezing the bill segment and completing the bill.

Summary Billed Accounts and Usage Requests

A "summary" account refers to an account with service agreements that cover more than one premise. If these premises are linked to service points that span different measurement cycle routes scheduled for reading on different dates, there's no guarantee that all the account's SAs will be billed at the same time. This is because batch billing for manually read meters is based on the meter's scheduled reading date.

For example, assume a summary account has two SAs each linked to a different premise. One is on route X, scheduled for reading every first of the month and the other is on route Y, scheduled for reading every 20th of the month. Assume that the account’s bill cycle is cycle X (which matches the schedule of premise/SP on route X). When billing runs on the first of the month, the service agreements tied to route Y won’t have any reads. And because route Y is not scheduled for reading, the system will not estimate the consumption. Instead, it will skip billing the route Y SAs until the next month. This means some SAs are always one billing cycle behind.

Copyright © 2026, Oracle and/or its affiliates.

If your implementation maintains meters and meter reads in CCB, then the base get consumption algorithm called ***Get Consumption From SP's Linked To SA*** handles the skipping of a summary billed account's service agreements based on bill cycle and measurement cycle schedules. The remainder of this section describes how this works in a CCB-MDM integrated environment.

- Requesting Bill Determinants for Summary Billed Accounts

Requesting Bill Determinants for Summary Billed Accounts

Usage requests for summary billed accounts are handled as follows:

- On the first night of the billing window usage requests are created for each SA and sent to Oracle Utilities Meter Data Management (MDM). MDM then checks what the next schedule read date is for each service point.
- Assuming Oracle Utilities Meter Data Management owns meter read schedules; when batch billing executes Oracle Utilities Customer Care and Billing (CCB) cannot determine whether a summary account's service agreement should be included on the bill or not.
- If MDM determines that an SA/usage subscription should be skipped from billing, it notifies CCB by including a skip indicator and reason on the usage response, as well as the SA's next scheduled read date. This date is captured as a ***MDM Next Scheduled Read Date*** characteristic on the SA, and the usage request is cancelled. The system still attempts to complete the bill since this might be the last outstanding usage request for the bill
- When billing next runs, the base ***Get bill segment consumption using a usage request*** algorithm first checks if the SA should be skipped as follows (Note that this check is also performed in first billing run):
- Get the ***MDM Next Scheduled Read Date*** characteristic on the SA. If this date is after the bill segment end date, we'll skip the SA from billing

If your implementation would like to wait until the bill cycle window end date before attempting to complete summary account's bills, ensure that the ***Complete Bill at End of Bill Cycle*** pre-bill completion algorithm is defined on customer class. This is useful if scheduled reads are expected in the bill cycle window, but not necessarily on the window start date.

Corrected Read Notifications

If a read that was used for bill determinant calculations is modified in the mdm system, notification is sent to Oracle Utilities Meter Data Management.. Resulting in the creation of an off cycle bill generator. The system uses the business object defined as the OCBG Corrected Read BO MO option on the Off Cycle Bill Generator maintenance object to create the OCBG.

What happens next depends on the lifecycle that your implementation has configured for the OCBG Corrected Read BO. Here are examples of what might occur:

- Create a To Do entry for manual follow up.
- Find any frozen bill segments that might be affected by the corrected read and perform cancel/rebill.

Configuring The System For Usage Request Integration

Copyright © 2026, Oracle and/or its affiliates.

Oracle Utilities Customer Care and Billing sends usage requests to Oracle Utilities Meter Data Management (MDM) in the form of an xml message. MDM sends responses back to Oracle Utilities Customer Care and Billing.

The following sections describe at a high level the data setup required to send usage requests to MDM.

Start and End Times for Billing

As you know, there is logic in billing to determine the start date and end date for a bill segment. Refer to Ways To Control The Start Date Of A Bill Segment and Ways To Control The End Date Of A Bill Segment for more information. When billing for a customer with interval data, the system also needs to know the time.

The time used by billing, referred to as the cutoff time, is stored on the service agreement. There is also a control on the service agreement called Start Day Option that determines which day to use for the start time. Billing uses the billing date, the cutoff time, and the start day option to determine the correct interval data to process.

Oracle Utilities Customer Care and Billing lacks knowledge of the type of meter installed at a service point. Thus, both interval processing period as well as scalar processing information is captured on a usage request. Oracle Utilities Meter Data Management (MDM) then uses the appropriate period to calculate bill determinants based on the type of meter installed.

MDM returns the true usage period used to calculate bill determinants on the usage response. This usage period is captured on the bill segment.

The Big Picture of Time of Use Mapping and Pricing

This section provides an overview of the concepts related to setting up your control tables to support time of use mapping and pricing with the rates.

> **Note**
>
> Time of use mapping is supported to enable low volume interval data manipulation (for example, for mapping electric vehicle charging events into time of use quantities). A meter data management application, such as Oracle Utilities Meter Data Management, must be used to manipulate high volumes of interval data and calculating bill determinants for billing purposes.

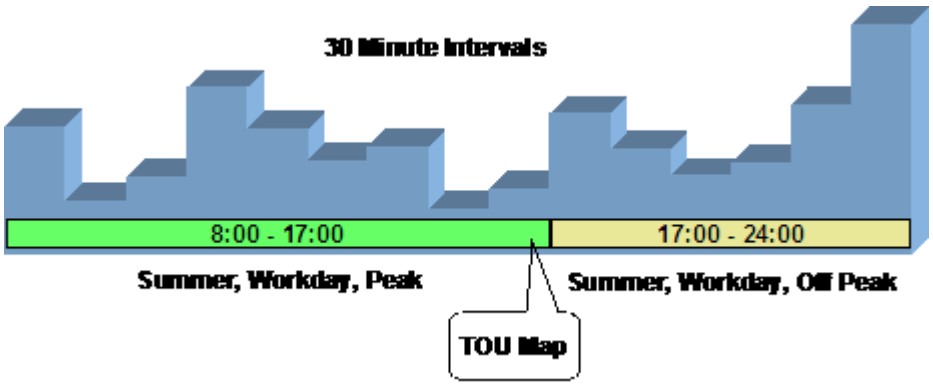
- Mapping to Time of Use (TOU) Periods
- Grouping of TOU Codes for TOU Mapping
- Time of Use (TOU) Mapping and Pricing
- Designing Your Time of Use (TOU) Mapping and Pricing Options
- Setting Up Time of Use (TOU) Mapping and Pricing Control Tables

Mapping to Time of Use (TOU) Periods

Copyright © 2026, Oracle and/or its affiliates.

Many customers choose not to price their interval data using interval prices. Customers may choose for their interval data to be mapped into time of use (TOU) periods. This option for interval data might be preferred because:

- Typically it involves fixed prices for the use periods
- It is more manageable than direct billing
- It is easier for a customer to forecast and budget



A TOU map's purpose is to define the TOU codes for a collection of time period definitions (i.e. given dates and times). The TOU Map has a TOU Map Type, which defines the interval size between TOU map data rows.

Time of use periods can (and often do) change during the year.

Map #123 (TOU Group 2)	
Effective 1 Jan 2000	
Interval Date/Time	
30/Apr/00 16:30	On Peak/Winter
30/Apr/00 16:45	On Peak/Winter
30/Apr/00 17:00	Off Peak/Winter
30/Apr/00 17:15	Off Peak/Winter
...	
01/May/00 7:45	Off Peak/Summer
02/May/00 8:00	On Peak/Summer
02/May/00 8:15	On Peak/Summer
...	

All the possible TOU codes for a given map are grouped together in a TOU group. Refer to Grouping of TOU Codes for TOU Mapping for more information.

Grouping of TOU Codes for TOU Mapping

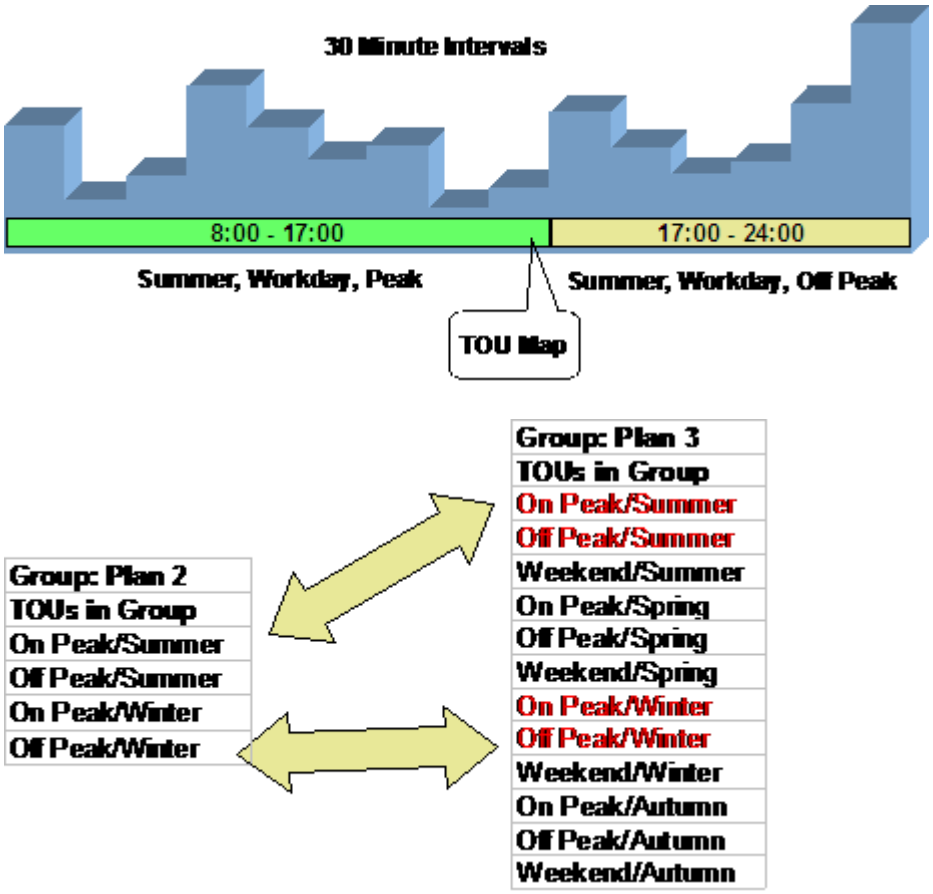
This section describes the relationship between TOU Codes and TOU Mapping.

- Overview of TOU Codes and TOU Mapping
- TOU Sequence for TOU Mapping and Pricing

Overview of TOU Codes and TOU Mapping

Copyright © 2026, Oracle and/or its affiliates.

A TOU map's purpose is to define the TOU codes for a collection of time period definitions (i.e. given dates and times).



TOU Sequence for TOU Mapping and Pricing

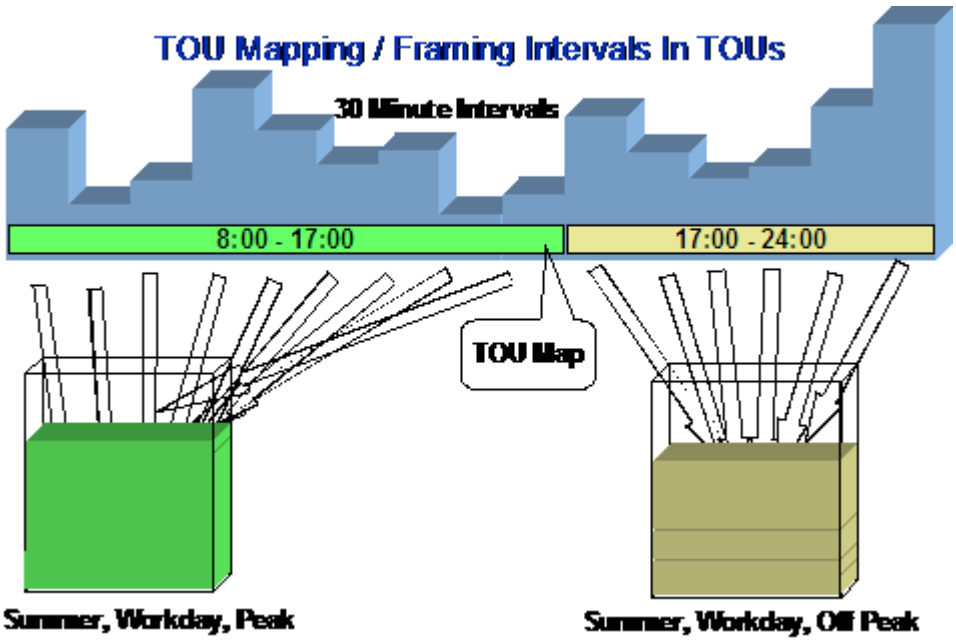
If desired, you may use sequence number to indicate the relative position or relative priority of each TOU code within a TOU group. This sequence number is not used by any system functionality, but is available for you to use in a plug-in algorithm.

Time of Use (TOU) Mapping and Pricing

The following section describes the logic used by the system to map an interval data curve to TOU periods using a TOU map, and subsequently apply prices to these mapped quantities.

At some point during billing, the system will take an interval data curve and will map the interval values to TOU periods based on a TOU Map. The interval data curve is retrieved from either a specific usage request or from the new rate engine buffer populated prior to calling the rate engine.

Copyright © 2026, Oracle and/or its affiliates.



This type of data manipulation is typically handled using a calculation rule based on the 'math' calculation rule type.

Once the interval data curve values have been mapped to TOU periods, prices can be applied to the time of use period quantities using calculation rules based on the 'service quantity' calculation rule type.

- Calculation Rules for TOU Mapping & Pricing
- TOU Map Used For Mapping

Calculation Rules for TOU Mapping & Pricing

Calculation Rules are used to perform time of use mapping and time of use pricing.

Time of Use Mapping

Time of use mapping will be typically handled using a calculation rule based on the 'math' calculation rule type. Calculation rules of this type can perform many functions. Two examples include:

- Applying a TOU map to an interval data curve to produce service quantities that will be added to the bill segment's SQ collection.
- Applying a TOU map to a derived interval data curve. For example, after deriving the power factor curve, perform TOU mapping on the resultant curve.

For further details on the 'math' calculation rule type, refer to Base Package Math Calculation Rule for more information.

Time of Use Pricing

Time of use pricing will be typically handled using calculation rules based on the 'service quantity' calculation rule type. Calculation rules of this type can perform many functions. For example:

- Add or update a service quantity in the bill segments SQ collection

Copyright © 2026, Oracle and/or its affiliates.

- Create bill calculations lines that levy charges based on some type of consumption. For example, applying a price to a specific service quantity in the bill segment's SQ collection

For further details on the 'service quantity' calculation rule type, refer to Base Package Service Quantity Calculation Rule for further information.

> **Note**
>
> Calculation rules are implemented using algorithms that contain the logic to be performed. If your implementation requires a new type of calculation rule, you can configure a new calculation rule type and reference the appropriate algorithm on it.

TOU Map Used For Mapping

An interval data curve is mapped into time-of-use periods using a TOU map. The TOU map used by the calculation rule for mapping interval data curve values to time of use periods is defined directly on the calculation rule.

Attributes of TOU Maps

TOU map types define attributes shared by TOU maps of a given type. The essential attributes of any TOU Map Type are the interval sizes between TOU map data rows, associated TOU map template(s) to use, and whether the TOU data will follow any seasonal time shifting.

Designing Your Time of Use (TOU) Mapping and Pricing Options

This section provides an overview for designing your control tables to support time of use (TOU) mapping and pricing with the new rating engine.

- Designing Your Time Of Use (TOU) Mapping and Pricing Calculation Rules
- Designing Your TOU Codes for TOU Mapping
- Designing Your TOU Groups for TOU Mapping
- Designing Your TOU Map Types for TOU Mapping
- Designing Your TOU Map Templates for TOU Mapping

Designing Your Time Of Use (TOU) Mapping and Pricing Calculation Rules

As you know from the rates chapter, the system can handle mapping of interval data curve values into different time of use periods and pricing these quantities using the 'math' and 'service quantity' calculation rule types respectively. Refer to Base Package Calculation Rule Descriptions for more information.

The following guidelines provides a high level outline on how you should go about designing a rate schedule that involves time of use mapping and pricing.

- Obtain copies of existing bills that use the rate in question. If the rate is new, then write up exactly how the information should appear on the customers' printed bills.
- Identify all the lines that represent charges for individual time of use periods.
- Determine how the quantities for the time of use periods are calculated. Which time of use map is used to define the time periods?

Copyright © 2026, Oracle and/or its affiliates.

- In order to perform time of use mapping and pricing, calculation rules need to be designed
  - A calculation rule based on the 'math' calculation rule type may be used to map an interval data curve into time of use periods. This calculation rule must know the TOU map to apply to the interval data curve to map into the relevant time of use periods. The time of use period quantities are populated in the bill segments SQ collection.
  - Calculation rules based on the 'service quantity' calculation rule type may be used to apply prices to each mapped time of use period quantity. These calculation rules each require either a price or regular bill factor for pricing each time of use period quantity. The bill factor may contain the price directly, or may indicate that the price is customer specific and can be found as contract quantities for the service agreement.

Once you have your calculation rules designed, you will be able to design the other control tables needed to set up your time of use billing customer.

Designing Your TOU Codes for TOU Mapping

The next most logical step in designing your time of use mapping controls is to define your time of use codes. To do this, look at the time of use periods to which your usage needs to be mapped. These values will likely correspond to the time of use quantities that your rate bills for. It should be noted that it's possible that will not bill for every time of use period.

Designing Your TOU Groups for TOU Mapping

To further aid in designing time of use mapping, the TOU Group enables you to group together all the time of use codes that are available to be used in a single map. You must also decide if you want to use a sequence number to define the relative order of a TOU code within a TOU group.

When TOU data is created for a TOU map, only TOU periods defined on a specified TOU group can be specified.

Designing Your TOU Map Types for TOU Mapping

Now that you have your TOU groups defined, you can begin defining TOU map types. Recall that the TOU map type defines the attributes that may be shared by TOU maps of the same type.

The essential attributes of any TOU Map Type are the interval sizes between TOU map data rows, associated TOU map template(s) to use, and whether the TOU data will follow any seasonal time shifting. Refer to Designing Your Time Options for more information.

TOU Map Type Interval Size

The interval size of a TOU map must divide evenly into the interval size of the interval data curve that uses the map (because the system joins the date/time of the interval data curve values to the date/time of the TOU data). This means that it is possible to use a 15 minute TOU map with a 60 minute interval data curve. However, it is not OK to have a 60 minute TOU map used with a 15 minute interval data curve because the join will miss 3 out of 4 interval data curve values.

Copyright © 2026, Oracle and/or its affiliates.

TOU Map Type Override Template

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support override templates. Refer to Designing Your TOU Map Templates for TOU Mapping for more information.

- A TOU map's TOU map type defines the default TOU map template that's used to generate its TOU data.
- A TOU map's type defines the TOU map templates that can be referenced on individual TOU maps to override the default template.
- An individual TOU map can have override templates. If the TOU map doesn't have an override template, the default template defined on the TOU map type is used to generate the map's TOU data.

Refer to Creating TOU Map Types for further information.

Designing Your TOU Map Templates for TOU Mapping

In order to help your users to create and maintain data for TOU maps, you may define TOU map templates, which can be used to generate data for a TOU map. The templates may be used to define standard data for a TOU map as well as data for special periods, such as holidays. TOU map types reference TOU map templates.

Every TOU map references a TOU map template that defines the rules for generating TOU data from that TOU map. Specifically, TOU map templates define:

- The TOU group (defines the valid TOU periods for the template) used for the TOU map
- The default TOU period used for periods not explicitly defined. (This means you don't have to specify dates and times for all periods. For example, if your default TOU period is "Off Peak" you only need to define dates and days and times for On Peak or other TOU periods.)
- The specific date ranges, days of the week, and time periods designated for each TOU period. The system periodically generates TOU map data for TOU maps by interpreting the rules defined in the template.

Holidays

Many utilities categorize consumption on holidays differently than on the day of week on which the holiday falls. For example, holiday consumption might be categorized as Off-Peak regardless of the day it falls on. TOU map templates can define rules for different TOU periods for holidays by specifying the following:

- A Work Calendar that defines when holidays start and end
- Either:
  - A Holiday TOU period for consumption on holiday
  - A Holiday TOU Map Template that defines the TOU codes to use for different times in the year

TOU Map Template Interval Size

TOU map templates can also specify an interval size. This value specifies the duration of the individual TOU map data records, and also controls the values allowed in the Start and End Times. For example, if a TOU map template sets the interval size at 15 minutes, Start and End times must be in units of the interval size (10:00, 10:15, 10:30, etc.).

Copyright © 2026, Oracle and/or its affiliates.

A TOU map template can be used to generate TOU map data for TOU maps whose interval size is divisible by the template's interval size. For example, a 60 minute template can be used to generate TOU data for TOU maps with an interval size of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every interval size.

Refer to Creating TOU Map Templates for further information.

Setting Up Time of Use (TOU) Mapping and Pricing Control Tables

This section provides an overview of setting up your control tables to support time of use mapping and pricing with rates.

- Creating TOU Map Templates
- Creating TOU Map Types

Creating TOU Map Templates

TOU map templates may be used to define standard data for a TOU map as well as data for special periods, such as holidays. TOU map types reference TOU map templates and used for TOU map data generation.

Prerequisites: You must define TOU groups and work calendars before you can create TOU map templates. Refer to the Oracle Utilities Application Framework online help for more information about creating work calendars.

To maintain existing TOU map templates, select **Admin**, and then **Usage**, and then **TOU Map Template**, and then **Search** then use standard actions to edit, duplicate, or delete a TOU map template.

To define a new TOU map template, follow these steps:

1. Select **Admin**, and then **Consumption**, and then **TOU Map Template**, and then **Add.** If your system supports more than one TOU map template business object, you will be prompted to select a business object for this TOU map template.
2. Enter a name and a meaningful description for the TOU map template.
3. Select the TOU group to be used by the TOU map template.
4. Select the default TOU for the TOU map template (from the TOU Group). This is the TOU used when creating TOU map data for dates not accounted for in the TOU Schedules section.
5. Select the work calendar for the TOU map template. Work calendars define the days of the week on which work is performed, and specify holidays.
6. Select the holiday TOU for the TOU map template (from the TOU Group).
7. Select the TOU map template used for holidays (if applicable).
8. Specify the interval size for TOU map data created from the map template. Interval size is designated as hours:minutes:seconds (HH:MM:SS)
9. To specify TOU schedule's date ranges and which TOUs should be used for this TOU map template, click the + or - sign in the TOU Schedule Section and enter or select the following:
    - Start and End Dates for a specific date range
    - Start and End Days of the Week: To add or remove Start and End Days of the Week pairs, click the + or - sign and select the appropriate weekdays

Copyright © 2026, Oracle and/or its affiliates.

- Start and End Times: To add or remove Start and End Times within a Start and End Days of the Week pair, click the + or - sign and enter the appropriate times
- TOU

10. Click **Save.**

Now you can use the TOU map template when creating TOU types.

Creating TOU Map Types

An interval data curve is mapped into time-of-use periods using a TOU map. The TOU map used by the calculation rule for mapping interval data curve values to time of use periods is defined directly on the calculation rule. TOU map types define attributes shared by TOU maps of a given type.

Prerequisites: You must define TOU map templates and time zones before you can create TOU map types. Refer to the Oracle Utilities Application Framework online help for more information about creating time zones.

To maintain existing TOU map types, select **Admin**, and then **Consumption**, and then **TOU Map Type**, and then **Search** then use standard actions to edit, duplicate, or delete a TOU map type.

To define a new TOU map type, follow these steps:

1. Select **Admin**, and then **Consumption**, and then **TOU Map Type**, and then **Add**

> **Note**
>
> If your system supports more than one TOU map type business object, you will be prompted to elect a business object for this TOU map type.

2. Enter a name and a meaningful description for the TOU map type.
3. Select the business object to use when creating TOU maps of this type
4. Select the time zone for the TOU map type.
5. Specify the interval size for TOU map data created from the map type. Interval size is designated as hours:minutes:seconds (HH:MM:SS)
6. Select the default TOU map template for the TOU map type
7. To add or remove override TOU map templates for this TOU type, click the + or - sign in the Override TOU Map Templates section and select the TOU map template.
8. Click **Save**.

Now you can use the TOU map type when creating TOU maps.

Usage Subscription Types

- Understanding Usage Subscription Types
- Configuring Usage Subscription Types

Understanding Usage Subscription Types

Copyright © 2026, Oracle and/or its affiliates.

Usage Subscription Types define a collection of properties defining a class of Usage Subscriptions. Usage Subscription types also control valid values for various attributes of Usage Subscriptions.

For a deeper functional understanding, refer to the About Usage Subscriptions or About Usage Calculation sections .

Configuring Usage Subscription Types

This portal is used to display and maintain a Usage Subscription Type.

Refer to Understanding Usage Subscription Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Type List**: displays all of the Usage Subscription Types so you can choose the one you want to display in more detail
- **Usage Subscription Type**: shows the specific configuration for the selected Usage Subscription Type

The Usage Subscription Type defines a number of lists for "valid" objects that can be used in conjunction with the overall usage calculation process:

- Valid Usage Recipients
- Valid Service Point Types
- Valid Usage Calculation Groups
- Valid Dynamic Option Types

Integrating Usage Transactions

- Requesting Usage Transactions
- Processing Usage Transactions
- Generating Usage Transactions in Oracle Utilities Meter Data Management

Requesting Usage Transactions

Usage transactions can be created via a request from the billing process.

To invoke a usage transaction request, the billing process invokes a Usage Transaction Request Inbound service script which invokes the Usage Transaction Seeder (D2-UsgTranSeeder) business object. This business object does the following:

- Determines the usage subscription ID based on an external usage subscription ID
- Determines the appropriate usage transaction business object to create. This uses the "How To Create Usage Subscription Related Information" processing method defined for the "Usage Transaction Creation" processing role on the service provider that represents the external system.

For a deeper functional understanding of Usage Transactions, refer to the About Usage Transactions section.

Copyright © 2026, Oracle and/or its affiliates.

Processing Usage Transactions

In Oracle Utilities Customer To Meter, usage transaction information, including bill determinants and other data is requested and “sent” to the billing process.

Usage transactions are sent when they enter the "Sent" state via a number of Enter algorithms on the base package usage transaction business object (D2-UsageTransaction) .

- The "Send Usage" (D2-SEND-USG) algorithm determines the method used to send the information, based on the Usage Recipient (service provider) defined on the usage transaction's Usage Subscription (in most cases, Customer To Meter).
- The “Send Usage Response” (X1–SENDUSGRP) algorithm sends a response to the billing process, and transitions the related usage request to its next state.
- The “Transition Related Usage Request from UT” (X1-TRNUSGREQ) algorithm transitions the related usage requests in the “Awaiting BD” state to its next state (if the usage transaction had been previously stopped at a non-final status (such as Approval In Progress).

Usage transactions can be sent via either online real-time processing or periodically via batch processing.

Online Real-Time Processing

To set up the service provider to support online real-time notification of usage transactions, do the following:

- Create one or more Outbound Message Types that reference the outbound message business object to be used to send usage transaction information to the billing system. The base package include the following business object for Usage Transaction Outbound Message (X1-UsageTranOutboundMesg).
- Define a Message Sender that will be used to send the message to the external system.
- Add the outbound message type to the service provider's External System and reference the Message sender created above.
- Add a processing method to the service provider as follows:
  - **Processing Role**: Usage Transaction Notification - Online
  - **Processing Method**: How To Send US Related Information
  - **Status**: Active
  - **Default Processing Method**:
  - **Outbound Message Type**: the outbound message type created above
  - **Override Processing Method**: outbound message types for specific usage subscription types if applicable

For a deeper understanding of integrating usage transactions with Oracle Utilities Customer Care and Billing, refer to the Configuring the Bill Determinant Interface in the Interface section.

Batch Processing

Requests from the billing process that have been indicated as "batch requests" (such as those produced by Oracle Utilities Customer Care and Billing batch billing process) accumulate in a "calculation deferred" state to be processed specially by the **Usage Transaction Calculate Defer Monitor** batch control (D2-UTCD).

Copyright © 2026, Oracle and/or its affiliates.

Requests from Oracle Utilities Customer to Meter that have been indicated as "batch requests" accumulate in a "calculation deferred" state to be processed specially by the **Usage Transaction Calculate Defer Monitor** batch control (D2-UTCD).

To set up the service provider to support periodic batch processing of usage transactions, use a periodic monitor batch control. These batch programs should invoke the business objects that will contain the usage transaction information. The base package includes the following business object for this: Usage Transaction Outbound Message (D2-UsageTranOutboundMesg).

Next, setup the following configuration:

- Add a processing method to the service provider as follows:
  - **Processing Role**: Usage Transaction Notification - Batch
  - **Processing Method**: How To Send US Related Information
  - **Status**: Active
  - **Default Processing Method**:
  - **Batch Control**: the batch control created above
  - **Override Processing Method**: batch controls for specific usage subscription types if applicable

Errors encountered when processing Usage Transactions can be reprocessed with the following batch controls:

- Reprocessing usage transaction "seeders" in error (D2-UTSED)
- Reprocessing usage transactions in error (D2-UTID)

If unexpected errors occur that leave usage transactions in an unmonitored state, the **Usage Transaction Monitor** batch control (D2-UT), or one based on this batch control with parameter values tailored to any specific requirements, can be used to process those usage transactions.

For a deeper functional understanding of Usage Transactions, refer to the About Usage Transactions section.

Generating Usage Transactions in Oracle Utilities Meter Data Management

There may be some cases where Usage Transactions should be generated in Oracle Utilities Meter Data Management rather than requested from an external system. Generation of these usage transactions are scheduled around the time when the meter installed on the service point is read.

Base product delivered a batch program that reads all pending measurement cycle schedule routes with selection date on or before the business date and creates as SP measurement cycle schedule route business object instance that stages the creation of these usage transactions.

For additional details refer to the D1-CSPSR batch control and D1-SPMsrmtCycScheduleRoute business object.

Usage Transaction Exception Types

- Understanding Usage Transaction Exception Types
- Configuring Usage Transaction Exception Types

Copyright © 2026, Oracle and/or its affiliates.

Understanding Usage Transaction Exception Types

Usage Transaction Exception Types define the groupings of exceptions for a Usage Transaction based on their functional similarity. This provides a way to define Usage Transaction Exceptions in a distinct enough way to understand the root issue that was generated from the usage calculation rule.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section.

Configuring Usage Transaction Exception Types

You ue the **Usage Transaction Exception Type** portal to display and maintain usage transaction exception types .

Refer to Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Transaction Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Transaction Exception Type List**: displays all of the Usage Transaction Exception Types so you can choose the one you want to display in more detail
- **Usage Transaction Exception Type**: shows the specific configuration for the selected Usage Transaction Exception Type

Usage Calculation Groups

- Understanding Usage Calculation Groups
- Configuring Usage Calculation Groups

Understanding Usage Calculation Groups

Usage calculation groups are collections of usage calculation rules that are used to either calculate bill determinants or validate bill determinants. During the usage transaction process, the system executes the usage calculation rules defined in the usage calculation group referenced on the usage subscription. The rules within a usage calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Usage calculation groups are associated with specific usage subscriptions and usage subscriptions types (or both). When assigned to usage subscriptions, usage calculation groups contain the usage calculation rules to be used to calculate usage / bill determinants. Usage calculation groups associated with usage subscription types are those groups considered valid for usage subscriptions of that type.

Usage calculation groups can also specify a list of device donfiguration types that are considered valid. Usage calculation groups should only be associated with usage subscriptions for service points related to device configurations of a valid device configuration type.

Usage calculation groups can also be referenced by the Execute Usage Calculation Group usage calculation rule.

Copyright © 2026, Oracle and/or its affiliates.

For a deeper functional understanding of usage calculation, refer to the About Usage Calculation section.

Configuring Usage Calculation Groups

This portal is used to display and maintain a usage calculation group.

Refer to Understanding Usage Calculation Groups for more information.

You can access the portal by selecting**Admin**, then **Usage**, then **Usage Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Group**: Defines basic information about usage calculation group
- **Usage Calculation Rules List**: lists the usage calculation rules belonging to the group
- **Referencing Usage Calculation Rules List**: lists the usage calculation rules that reference the group
- **Referencing Usage Subscription Type List**: lists the usage subscription types that reference the group
- **Referencing Usage Subscriptions List**: lists the usage subscriptions that reference the group

Usage Calculation Rules

- Understanding Usage Calculation Rules
- Configuring Usage Calculation Rules
- Detailed Configuration Examples

Understanding Usage Calculation Rules

Usage calculation rules are standard and custom rules that perform calculations on measurement data to generate bill determinants and other values used by external systems, such as billing systems, customer information systems, etc. Usage calculation rules are created for a specific usage calculation group. For example, if you were configuring two usage calculation groups and both included a specific usage calculation rule, you would need to create two instances of the usage calculation rule, one for each group.

The base package includes rules that calculate common bill determinants including:

- Scalar reads
- Time-of-use consumption (by applying a time-of-use map to an interval channel)
- Interval curves (either real or derived)
- Virtually anything else that can be calculated from the information in the system

For a deeper functional understanding of usage calculation, refer to the About Usage Calculation section.

Copyright © 2026, Oracle and/or its affiliates.

Usage Transaction Exceptions

On almost every usage calculation rule, the failure of the rule results in a usage transaction exception and the Usage Transaction Exception Type for the failure can be configured on the rule. These usage transaction exception types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the usage transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the usage transaction from being sent. Multiple "issue exceptions" can be created during usage transaction processing. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the usage calculation process to stop and the usage transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a Usage Transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full usage period should always be set to Terminate as an Exception Severity.

Rounding Usage Transaction Service Quantities

Several usage calculation rules provide a means for defining the manner in which service quantities are rounded during usage calculations. Service quantity rounding can be defined using the **SQ Rounding Details** which allow configuration of the method by which quantities are rounded (Up, Down, or Nearest) and the number of decimal positions retained after rounding.

Creating Measurement Data Snapshots

Daily Scalar Calculation Rule, Get Interval, Get Item Counts and Consumption, and Get Scalar Details usage calculation rules can be configured to create measurement data snapshots during usage transaction processing. Measurement data snapshots are used by data aggregation in Market Settlements Management.

Measurement data snapshot configuration is done in the **Extract To Measurement Data Snapshots** section, and includes the following options:

- **Measurement Data Snapshot Type**: The measurement data snapshot type for measurement data snapshots created by the usage rule.
- **Usage Transaction ID To Use**: A flag that designates the type of ID to use to identify the usage transaction on the measurement data extract. Options include:
  - **Usage Transaction ID**: The usage transaction's ID
  - **Usage Transaction External ID**: The external ID for the usage transaction. Used when the usage transaction is triggered by external system such as a CIS (or by the billing functionality in Customer To Meter)
- **Extract To Different Usage Subscription**: A flag that designates whether or not the measurement data snapshot should be extracted to a different usage subscription, such as one used by Market Settlements Management. Options include:
  - **No**: The extract will reference the usage subscription for which usage calculations are being processed. This option is used when Meter Data Management and Market Settlements Management are installed in different application environments. In this

Copyright © 2026, Oracle and/or its affiliates.

case, the extract can be exported from Meter Data Management for subsequence import into Market Settlements Management.

  - **Yes**: The extract will reference a different usage subscription such as one used by Market Settlements Management. This option is used when Meter Data Management and Market Settlements Management are installed in the same application environment. In this case, the measurement data snapshot is “exported” within the system to the Settlement Item, where it can be used in data aggregation processing. When this option is selected, the **Source Identifier Type** and **Target Identifier Type** fields must be configured to define how the system will determine the correct settlement usage subscription to use.

- **Irregular Data Flag Mapping**: Ranges of condition codes used to flag data as “irregular”. The value of the **Irregular Data Flag** (based on the IRREGULAR_DATA_FLG lookup) is used to flag data that falls with each range configured. A blank **Irregular Flag** value denotes that the condition range is considered to be "Regular".

See About Measurement Data Snapshots for more information about measurement data snapshot creation and processing.

Usage Calculation Rules and Anchor Time

Some usage calculation rules have been enhanced to support usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

Configuring Usage Calculation Rules

This portal is used to display and maintain a usage calculation rule.

Refer to Understanding Usage Calculation Rules, Understanding Usage Calculation Groups, and Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Rule**: Defines the usage calculation rule, including parameters used when executing the rule
  - Click **Show All Rules** to open the **VEE Rules** List zone, which lists all of the usage calculation rules in the usage calculation group to which the current rule belongs.
- **Eligibility Criteria List**: Lists the eligibility criteria defined for the rule

Pre-Calculation Usage Calculation Rules

The following is a list of the pre-calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose

Copyright © 2026, Oracle and/or its affiliates.

Alignment and Delay Usage Calculation Rule	Interval or Scalar	This rule can be used to handle two main needs: 1) aligns all measuring components for a Usage Subscription to the same date (whether on the same device or separate devices) 2) delays the usage transaction until the end of the retry window based on the quality of available data and other configured parameters.
Check Existence of Installed Device	Interval or Scalar	This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period.

Calculation Usage Calculation Rules

The following is a list of the calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose
Apply Math (Interval Data)	Interval	This rule is used to perform calculations on interval data and stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions). This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a usage subscription. This rule can also multiply total usage by a factor using a custom formula.

Copyright © 2026, Oracle and/or its affiliates.

Coincident Peak and Individual SP Demand	Interval	This rule is used to calculate the coincident peak demand for all of the service points on the usage subscription and the demand values for each service point at the coincident peak. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point).
Daily Scalar Usage Calculation Rule	Scalar	This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified usage period. It creates bill determinants by taking the difference between the beginning and ending reading for the bill period. This rule can also be used to provide register readings by measuring component. For consumption values, only the beginning and ending readings are exported This rule supports date breaks (the normal Get Scalar Data rule does not).
Get Interval Data	Interval	This rule is used to get interval quantities from interval measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. This rule retrieves the interval data for measuring components associated to a usage subscription by TOU, SQI and UOM. This rule converts the interval data to another interval length or unit of measure.
Get Item Counts and Consumption	Interval or Scalar	This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and Service Point, and calculates item-based consumption.

Copyright © 2026, Oracle and/or its affiliates.

Get Scalar Details	Scalar	This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period. This rule creates bill determinants by summing all scalar readings for the bill period. This rule can also be used to provide register readings by measuring component. All readings are exported by this rule. Note: This rule is used for traditional monthly read meters.
Get Subtractive Interval Details	Interval	This rule is used to get interval quantities from subtractive interval measuring components installed on the service points linked to the usage subscription for the specified 'interval' usage period. It also identifies the start and stop readings for each usage period using subtractive interval readings.
Get TOU Mapped Usage	Interval	This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period.
Interval Tier Calculation	Interval	This rule calculates the difference between a source and reference vector. This rule loops through each tier that is configured and calculates the imbalance amount associated to that tier level. This rule breaks down that difference into one to many positive or negative tiers, and create a service quantity for each tier calculated.
Profile Accumulation	Interval	This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction.

Copyright © 2026, Oracle and/or its affiliates.

Rolling Demand	Interval	This rule is used to calculate rolling demand for all of the service points on the usage subscription. Rolling demand can be based on a rolling consumption total, rolling consumption average, or coincident peak demand. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point).
Round and Adjust Usage	Interval or Scalar	This rule copies identified source and target Service Quantities and inserts these as Service Quantities that are rounded and adjusted.
Vector and Service Quantity Math	Interval	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations. Typical uses: Perform math using interval data, e.g., take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc. Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values). Support math functions: sin, cos, square root, etc. Store derived curves in memory that can be used in subsequent calculations Please note, this rule is not as efficient as other rules.

Post-Calculation Usage Calculation Rules

Below is a list of the post-calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Applicable Data Type(s)	Purpose

Copyright © 2026, Oracle and/or its affiliates.

Usage High/Low Rule	Interval or Scalar	This rule compares the Service Quantities of the Usage Transaction to historical values. If the current value is too high or too low compared to historical data then an exception is thrown.
Validate Against Tolerance	Interval or Scalar	This rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule.
Business Flag Hold	Interval or Scalar	This rule can stop a usage transaction from proceeding when there have been business flags for the applicable service points. The hold can either be indefinite or set to expire a configurable amount of time prior to the calculation window ending.

Decision-Making Usage Calculation Rules

There are usage calculation rules delivered as part of the base product that help with decision-making when executing running the usage calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Usage Calculation Rule Name	Purpose
Execute Usage Calculation Group	This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group.
Exception Handler	This rule is used to terminate execution of the usage processor if exception count criteria specified in the rule is met.

Advanced Aside: Using Factors For Variables

A situation common in some implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this can differ based on many different types of criteria, such as the location of the service point or other characteristics. This sort of calculation can be implemented as a usage calculation rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of "Number," and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).
- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.

Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value. At run time, the rule retrieves /

Copyright © 2026, Oracle and/or its affiliates.

derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value. Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute**: you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and yet a third value if the amount conserved is greater than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.
- Pre-Calculation Usage Calculation Rules
- Calculation Usage Calculation Rules
- Post-Calculation Usage Calculation Rules
- Decision-Making Usage Calculation Rules

Pre-Calculation Usage Calculation Rules

- Alignment and Delay Usage Calculation Rule
- Check Existence of Installed Device

Alignment and Delay Usage Calculation Rule

This rule attempts to delay usage calculation until high quality data for calculating bill determinants have become available. It is especially useful for usage subscriptions having multiple sources of usages such as multiple service points or multiple measuring components on the installed meter. The rule ensures that the usage calculated is based on a common date or alignment date, identified for these sources.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ALGNDELAY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-AlignmentandDelay

Check Existence of Installed Device

This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CHKEXTDVC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ChkExistenceofInstalledDvc

Calculation Usage Calculation Rules

- Apply Math (Interval Data)
- Daily Scalar Usage Calculation Rule
- Get Interval Data
- Get Item Counts and Consumption
- Get Scalar Details
- Get Subtractive Interval Details
- Get TOU Mapped Usage
- Interval Tier Calculation
- Profile Accumulation
- Profile Service Quantity
- Rolling Demand
- Round and Adjust Usage
- Vector and Service Quantity Math

Apply Math (Interval Data)

This rule is used to perform calculations on interval data and it stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-APPMATHIN Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

Copyright © 2026, Oracle and/or its affiliates.

D2-ApplyMathInt

Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

Scenario 1: Get the total KWH for the period

- **Calculation Type** = Single Variable
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum

Scenario 2: Get the higher value between the total KWH or total KVARH

- **Calculation Type** = Math Function
- **Math Function** = Max
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum
- **Variable Name** = V2, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KVARH
  - Set Function = Sum

Scenario 3: Total KWH multiplied with a factor value

- **Calculation Type** = Mathematical Expression
- **Mathematical Expression** = V1 * V2
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum
- **Variable Name** = V2, where the following is configured:
  - Variable Type = Factor
  - Factor = factor code that holds the multiplier

Daily Scalar Usage Calculation Rule

This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified 'Interval' usage period. Only automatically read scalar measuring components with value identifier that matches usage calculation rule's UOM/TOU/SQI will be processed.

Copyright © 2026, Oracle and/or its affiliates.

If configuring this rule with interval and scalar usage calculation rules under a usage calculation group, it should be executed after interval usage calculation rule and before the scalar usage calculation rule.

This rule calculates usage based on the beginning and ending reading values -- ignoring intermediate readings. This rule will not estimate data based on the usage transaction estimate flag. Frequently read scalar devices must use periodic estimation processes.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETFRESCR Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-CalFrequentlyReadScalar

Get Interval Data

This rule is used to get interval quantities from interval measuring components installed in the service points linked to the usage subscription for the specified 'Interval' usage period. Only measuring components that match the UOM/SQI defined in the usage calculation rule are processed. Measurements within the period are stored in the usage transaction's service quantities' interval data list. The SQ entry's quantity is calculated based on the **Calculate Function** defined in the usage calculation rule. This is done for every entry in the usage period list. Each service quantity recorded by this rule is linked to a service point and measuring component.

By supplying the Axis Conversion parameters on the usage calculation rule, this algorithm will convert the identified measuring component's consumption into the supplied UOM and interval size prior to storing the consumption into the service quantity's interval data list.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETINTDAT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-GetIntervalData

Get Item Counts and Consumption

This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and

Copyright © 2026, Oracle and/or its affiliates.

Service Point, and calculates item-based consumption. Once the item detail entries are created, the rule processes goes through them to create usage period SQs.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETITEMCC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-GetItemCountsConsumption

Get Scalar Details

This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period. By default all scalar measuring components are processed but if specific UOMs/TOUs/SQIs are defined in the usage calculation rule then only applicable measuring components are processed. Measurements within the period are retrieved. The usage transaction request may indicate whether or not 'Estimate' measurements are allowed. If the usage transaction request indicates that ‘Estimate’ measurements are allowed but an estimated end measurement cannot be found an Estimate initial measurement is created.

The measurement details are stored in the usage transaction's scalar details. The usage is also stored in the usage transaction's service quantities unless otherwise specified in the usage calculation rule (using Build Service Quantity indicator).

This rule can be configured to perform measurement quality assessment, which will result in the population of the measurement quality list with the quantities and corresponding conditions of those measurements (regular, estimated, etc.).

Please note: this rule should not be used with readings occurring daily or more frequently as it will retrieve all of the readings to store in the Usage Transaction. The Daily Scalar usage calculation rule may be a better option for this use case.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETSCALAR Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-GetScalar

Get Subtractive Interval Details

Copyright © 2026, Oracle and/or its affiliates.

This rule is used to calculate the usage for a subtractive interval measuring components. This differs from the Get Interval Data usage calculation rule in that it also retrieves a start and stop reading for each usage period it calculates. The start and stop readings are then used to provide entries into the usage scalar details table which are then made available in the usage transaction sent to the billing system for bill presentment.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUBINTADJ Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-GetSubtractiveInterval

Get TOU Mapped Usage

This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. Only measuring components that match the UOM/SQI defined in the usage calculation rule instance are processed.

Measurements within the period are mapped to time of use quantities based on the TOU map defined on the usage calculation rule. If dynamic options are specified in the referenced TOU map and if there are dynamic option events in effect within the usage period then the TOU map associated with the dynamic option is used for the entire dynamic option event period. This calculation is performed for every usage period requested.

The calculated time of use quantities are stored in the usage transaction's service quantities per Service Point and measuring component.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETTOUUSG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-GetTOUUsage

Interval Tier Calculation

This rule calculates the difference between a source and reference vector. It then breaks that difference into one to many positive or negative tiers. For each tier calculated it will create a service quantity with the optional ability to create additional service quantities for the total of all

Copyright © 2026, Oracle and/or its affiliates.

positive tiers and the total of all negative tiers. This rule will loop through each tier that is configured and calculate the imbalance amount associated to that tier level.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETINTIER Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-IntervalTierCalculation

Profile Accumulation

This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction.

The typical application of this rule is for customer self-service rate compare, in which a self-service user has selected a set of usage adjustments to apply to their historical consumption to assess the effect on the amount consumed.

This rule algorithm retrieves the customer's usage (the source UOM) for the usage period using one of two options:

- A channel linked to a Usage Subscription
- A usage transaction service quantity

Then it takes each entry in the usage transaction's profile factor list, finds the profile measuring component for each, retrieves the measurement data for the profile, applies axis conversion to align it to the source UOM and target interval size, and adds it to the source vector. Each profile's data is added progressively to arrive at a final vector. The final vector may then be TOU-mapped, and the vector itself may be preserved or discarded.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-DYNPRFLAC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ProfileAccumulation

Profile Service Quantity

This rule intervalizes scalar service quantity values based on a profile measuring component.

Copyright © 2026, Oracle and/or its affiliates.

This rule first retrieves scalar service quantities, identified by the rule's UOM/TOU/SQI. If multiple service quantities exist for the service point they will be summarized at the service point level. In the case of a split or broken usage period for a service point, the usage period is merged and the quantities are summarized. Each service quantity is subject to intervalization. The profile measuring component is then derived using the profile factor supplied with either the usage subscription or service point as characteristic sources. If the profile is not available, an exception will be created using the exception details on the rule.

If a dynamic option type is provided, the events associated with it are retrieved for the usage transaction period. Any intervalized quantities that coincide with the dynamic option events start/end date time are then recorded in a new usage period service quantity of usage type “Correlated SQ”.

If no dynamic option type provided, the intervalized quantities are recorded in a new usage period service quantity of usage type “Profile SQ”.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-PRFSVCQTY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ProfileServiceQuantity

Rolling Demand

This rule is used to calculate rolling demand for all of the service points on the usage subscription. Demand is calculated from consumption. In electric service markets, these are typically represented by kilowatts (kW) and kilowatt hours (kWh) respectively, where demand values are calculated by multiplying consumption interval values by the number of intervals per hour. For example, the demand associated with a fifteen (15) minute interval value of 25kWh would be 100kW (25 x 4 intervals per hour).

Interval Data Retrieval Parameters

This rule can be configured to use specific data either directly from a specific device and measuring component on the service point or calculated from more than one devices and/or measuring components on the service point. Each type of data is referred to as a "channel" of data. For example, a service point with solar panels might have separate channels for "Consumed" and "Generated" usage, and the rule could be configured to use data from either or based on calculations involving both.

Parameters used to retrieve interval data for this rule include:

- **Check Missing Intervals**: A flag that indicates if the rule should check for missing intervals during the calculation period.
- **No Meter Condition**: The measurement condition assigned to intervals inserted to fill gaps during periods in which when no meter is installed. If not set, the measurement condition defaults to "No Read - System" (100000).

Copyright © 2026, Oracle and/or its affiliates.

- **SP Device Filter**: An algorithm used to filter the service point device configuration that will be processed by the rule.
- **Common UOM**: The unit of measure that is common among the multiple channels of data used by the rule, if applicable
- **Common Interval Size**: The interval size used to align service point data of different interval sizes. For instance, if a service point has channels measuring data in both 1-hour and 15-minute intervals, setting this to 01:00:00 (1 hour) would scale the 15-minute data to hourly intervals.
- **Channels**: A list of the data channels measured at the service point. Each should be assigned a variable name and the appropriate UOM, TOU, and SQI.
- **Aggregation Method**: The method by which data from multiple channels will be aggregated. Channels can be combined using a simple formula or using a conditional formula.

To define a single channel for the rule to use, configure the "Vector 1 (IV1)" variable in the **Channels** section and reference that variable in a simple formula.

To configure the rule to use multiple channels, define each as a variable in the **Channels** section and configure either a simple or conditional formula to perform any needed evaluations or calculations on the channels.

Defining Formula Operands

Operands used in both simple and conditional formulas are defined in the Operand Lookup (D1-CoincidentPeakOperandLookup) extendable lookup. Operands used to represent channels should use the values in parentheses in the **Variables** drop-down list in the **Channels** section. For instance, the operand for "Vector 1 (IV1)" should be "IV1", the operand for "Vector 2 (IV2)" should be "IV2", and so on.

Operands can also be simple formulas based on the variables defined in the **Variables** drop-down list in the **Channels** section. For instance, a simple formula in which the values from Vector 1 are subtracted from the values in Vector2 could be defined as "IV2 - IV1".

Calculation Modes

This rule has two main modes: calculating rolling demand for a usage subscription based on either a rolling consumption total (Rolled Total) or rolling consumption average (Rolled Average), or calculating the coincident peak among all of the service points on a usage subscription based on either a rolling consumption total (Rolled Total - Coincident Peak) or rolling consumption average (Rolled Average - Coincident Peak). The coincident peak calculation can be performed on either all readings, or those designated as "on-peak" or "off-peak" via time of use mapping using the Coincident On-Peak or Coincident Off-Peak function, See **Defining Time of Use Mapping Options** for more information about time of use mapping.

The number of intervals in the rolling total or rolling average is determined via rule configuration. The **Roll Interval Size** parameter (in the **Calculation Details** section) divided by the **Common Interval Size** parameter (in the **Interval Data Retrieval Parameters** section). For example, if **Roll Interval Size** was 01:00:00 (1 hour) and the **Common Interval Size** was 00:15:00 (15 minutes), the rolling total or average would be based on sets of 4 intervals (1 hour divided by 15 minutes).

Rolled Demand Total

The "Rolled Demand Total" **Roll Calculate Function** identifies the peak demand for the usage subscription based on a rolling consumption total.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling total (calculated by multiplying the running total by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest demand for the usage subscription during the calculation period (indicated in **bold**).

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Demand
10/27/2022 12:00 PM	12	11	23		
10/27/2022 1:00 PM	10	12	22		
10/27/2022 2:00 PM	11	13	24		
10/27/2022 3:00 PM	13	13	26	95	95
10/27/2022 4:00 PM	14	14	28	100	100
10/27/2022 5:00 PM	13	13	26	104	104
10/27/2022 6:00 PM	12	13	25	105	**105**
10/27/2022 7:00 PM	11	12	23	102	102
10/27/2022 8:00 PM	12	13	25	99	99
10/27/2022 9:00 PM	13	14	27	100	100
10/27/2022 10:00 PM	12	14	26	101	101
10/27/2022 11:00 PM	12	12	24	102	102

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

Date/Time	Demand
10/27/2022 6:00 PM	105

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

Copyright © 2026, Oracle and/or its affiliates.

Rolled Demand Total - Coincident Peak / Rolled Demand Total - Coincident On Peak / Rolled Demand Total - Coincident Off Peak

The "Rolled Demand Total - Coincident Peak / Rolled Demand Total - Coincident On Peak / Rolled Demand Total - Coincident Off Peak" **Roll Calculate Function** identify the coincident peak demand for all of the service points on the usage subscription. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). The coincident peak demand is the demand based on the rolling total at the coincident peak.

The data used in the calculation is based on the specific **Roll Calculate Function** selected:

- **Total - Coincident Peak**: Evaluates all measurements matching the configured unit of measure
- **Total - Coincident On Peak**: Evaluates measurements matching the configured unit of measure that falls with the "on peak" TOU period defined in the **Results** section based on a specified TOU Map Template.
- **Total - Coincident Off Peak**: Evaluates measurements matching the configured unit of measure that falls with the "off peak" TOU period defined in the **Results** section based on a specified TOU Map Template.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling total (calculated by multiplying the running total by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest total for all of the service points within the calculation period (indicated in **bold**).

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Demand
10/27/2022 12:00 PM	12	11	23		
10/27/2022 1:00 PM	10	12	22		
10/27/2022 2:00 PM	11	13	24		
10/27/2022 3:00 PM	13	13	26	95	95
10/27/2022 4:00 PM	14	14	28	100	**100**
10/27/2022 5:00 PM	13	13	26	104	104
10/27/2022 6:00 PM	12	13	25	105	105

Copyright © 2026, Oracle and/or its affiliates.

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Demand
10/27/2022 7:00 PM	11	12	23	102	102
10/27/2022 8:00 PM	12	13	25	99	99
10/27/2022 9:00 PM	13	14	27	100	100
10/27/2022 10:00 PM	12	14	26	101	101
10/27/2022 11:00 PM	12	12	24	102	102

This rule would return the following values in the usage transaction service quantities list, which represent the demand at the coincident peak:

Date/Time	Coincident Peak Demand
10/27/2022 4:00 PM	100

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

Rolled Demand Average

The "Rolled Demand Average" **Roll Calculate Function** identifies the peak demand for the usage subscription based on a rolling average (calculated from the rolling total).

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Rolling Average**: The rolling average of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling average (calculated by multiplying the rolling average by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest demand for the usage subscription during the calculation period (indicated in **bold**).

Copyright © 2026, Oracle and/or its affiliates.

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Rolling Average	Demand
10/27/2022 12:00 PM	12	11	23			
10/27/2022 1:00 PM	10	12	22			
10/27/2022 2:00 PM	11	13	24			
10/27/2022 3:00 PM	13	13	26	95	23.75	23.75
10/27/2022 4:00 PM	14	14	28	100	25	25
10/27/2022 5:00 PM	13	13	26	104	26	26
10/27/2022 6:00 PM	12	13	25	105	26.25	**26.25**
10/27/2022 7:00 PM	11	12	23	102	25.5	25.5
10/27/2022 8:00 PM	12	13	25	99	24.75	24.75
10/27/2022 9:00 PM	13	14	27	100	25	25
10/27/2022 10:00 PM	12	14	26	101	25.25	25.25
10/27/2022 11:00 PM	12	12	24	102	25.5	25.5

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

Date/Time	Demand
10/27/2022 6:00 PM	26.25

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

Rolled Demand Average- Coincident Peak /Rolled Demand Average-Coincident On Peak / Rolled Demand Average - Coincident Off Peak

The "Average -Coincident Peak / Average - Coincident On-Peak / Average - Coincident Off-Peak"**Calculate Function**identify the coincident peak demand for all of the service points on the usage subscription. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). The coincident peak demand is the demand based on the rolling average at the coincident peak.

The data used in the calculation is based on the specific**Roll Calculate Function**selected:

Copyright © 2026, Oracle and/or its affiliates.

- **Rolled Demand Total - Coincident Peak**: Evaluates all measurements matching the configured unit of measure
- **Rolled Demand Total - Coincident On Peak**: Evaluates measurements matching the configured unit of measure that falls with the "on peak" TOU period defined in the**Results**section based on a specified TOU Map Template.
- **Rolled Demand Total - Coincident Off Peak**: Evaluates measurements matching the configured unit of measure that falls with the "off peak" TOU period defined in the**Results**section based on a specified TOU Map Template.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period.This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a**Roll Interval Size**of 01:00:00 (1 hour) and**Common Interval Size**of 00:15:00 (15 minutes))
- **Rolling Average**: The rolling average of 4 total values (based on a**Roll Interval Size**of 01:00:00 (1 hour) and**Common Interval Size**of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling average (calculated by multiplying the rolling average by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule wouldidentify the highest demand for the usage subscription during the calculation period (indicated in**bold**).

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Rolling Average	Demand
10/27/2022 12:00 PM	12	11	23			
10/27/2022 1:00 PM	10	12	22			
10/27/2022 2:00 PM	11	13	24			
10/27/2022 3:00 PM	13	13	26	95	23.75	23.75
10/27/2022 4:00 PM	14	14	28	100	25	**25**
10/27/2022 5:00 PM	13	13	26	104	26	26
10/27/2022 6:00 PM	12	13	25	105	26.25	26.25
10/27/2022 7:00 PM	11	12	23	102	25.5	25.5
10/27/2022 8:00 PM	12	13	25	99	24.75	24.75
10/27/2022 9:00 PM	13	14	27	100	25	25
10/27/2022 10:00 PM	12	14	26	101	25.25	25.25

Copyright © 2026, Oracle and/or its affiliates.

Date/Time	Service Point 1	Service Point 2	Total	Rolling Total	Rolling Average	Demand
10/27/2022 11:00 PM	12	12	24	102	25.5	25.5

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

Date/Time	Demand
10/27/2022 4:00 PM	25

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

Notes and Examples

Notes and Recommendations

- When using he "Coincident On Peak" or "Coincident Off Peak"**Roll Calculate Function**, you must specify either a TOU map (in the**TOU Map**field) or a TOU map template (in the**On Peak Off Peak Schedule**drop-down list) in the**Calculation Details**section,and specify the appropriate time of use period (On-Peak or Off-Peak, respectively) in the**TOU**field in the**Results**section.
- A recommended practice is to configure a Service Quantity Identifier (SQI) to uniquely identify coincident peak data and specify it in the**SQI**field in the**Results**section.
- Additional detail on the logic of this rule can be found in the **Detailed Description** of the D1-CALROLDEM Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-CalcRollingDemandRule

Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

Scenario 1: Calculate demand for services points based on rolling total (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    - **Variable 1**:

Copyright © 2026, Oracle and/or its affiliates.

        - **Variable Name**: Variable 1 (IV1)
        - **UOM**: Kilowatt-Hour
    - **Aggregation Method**:
        - **Formula Source**: Simple Vector Formula
        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Roll Calculate Function**: Rolled Demand Total
    - **Roll Interval Size**: 01:00:00
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**:
    - **SQI**:

Scenario 2: Calculate demand for service points based on rolling total (based on two channels)

In this scenario, the service point has two channels, one for consumed energy, and one for generated energy (from a solar panel). If the consumed value is greater than the generated value, subtract the generated value from the consumed value. If the generated value is greater than the consumed value, use zero (0).

- **Interval Data Retrieval Parameters**:
    - **Check Missing Intervals**: Yes
    - **No Meter Condition**: No Read - System
    - **Common UOM**: Kilowatt-Hour
    - **Common Interval Size**: 01:00:00
    - **Channels**:
        - **Variable 1**:
            - **Variable Name**: Variable 1 (IV1)
            - **UOM**: Kilowatt-Hour
            - **TOU**:
            - **SQI**: Consumed
        - **Variable 2**:
            - **Variable Name**: Variable 2 (IV2)
            - **UOM**: Kilowatt-Hour
            - **TOU**:
            - **SQI**: Generated
    - **Aggregation Method**:
        - **Formula Source**: Conditional Vector Formula
        - **Conditional Vector Formula**:
            - **Sequence**: 10
            - **Left Operand**: IV1

Copyright © 2026, Oracle and/or its affiliates.

        * **Criteria Operator**: >
        * **Right Operand**: IV2
        * **True Action**: Apply True Formula
        * **True Formula**: IV1-IV2
        * **False Action**: Apply False Formula
        * **False Formula**: 0
- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Total
  - **Roll Interval Size**: 01:00:00
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:

Scenario 3: Calculate demand for services points based on rolling average (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    * **Variable 1**:
      * **Variable Name**: Variable 1 (IV1)
      * **UOM**: Kilowatt-Hour
  - **Aggregation Method**:
    * **Formula Source**: Simple Vector Formula
    * **Simple Vector Formula**: IV1
- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Average
  - **Roll Interval Size**: 01:00:00
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:

Scenario 4: Total - Coincident Peak (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System

Copyright © 2026, Oracle and/or its affiliates.

- **Common UOM**: Kilowatt-Hour
- **Common Interval Size**: 00:15:00
- **Channels**:
  - **Variable 1**:
    - **Variable Name**: Variable 1 (IV1)
    - **UOM**: Kilowatt-Hour
- **Aggregation Method**:
  - **Formula Source**: Simple Vector Formula
  - **Simple Vector Formula**: IV1

- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Total, Coincident Peak
  - **Roll Interval Size**: 01:00:00
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**: Coincident Peak

Scenario 5: Average - Coincident On Peak (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    - **Variable 1**:
      - **Variable Name**: Variable 1 (IV1)
      - **UOM**: Kilowatt-Hour
  - **Aggregation Method**:
    - **Formula Source**: Simple Vector Formula
    - **Simple Vector Formula**: IV1
- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Average, Coincident, On Peak
  - **Roll Interval Size**: 01:00:00
  - **Reference TOU Map by Template**: Yes
  - **On Peak Off Peak Schedule**: <TOU map template>
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**: On Peak

Copyright © 2026, Oracle and/or its affiliates.

- **SQI**: Coincident On Peak

Round and Adjust Usage

This rule copies identified source and target Service Quantities and inserts these as period Service Quantities that are rounded and adjusted. This rule captures the source and target Service Quantity identifiers, rounding method, and Service Quantity bucket to hold the adjustment. The usage calculation rule also has an option to validate the difference between the source and target service quantities.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-RNDADJUSG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-RoundAndAdjustUsage

Vector and Service Quantity Math

This rule is designed to facilitate configuration of complex vector and scalar quantity calculations. It is based on a series of underlying services, including Axis Conversion, Apply Formula (most importantly), and Apply TOU Map. It can be configured to accept as input up to five vectors - where a single vector can represent the combination of all measuring components with linkages to the Usage Subscription with the configured UOM/SQI. It can also accept a list of scalar variables that can be taken in as factor values, usage transaction service quantities, set functions on a vector, or numeric values.

Vectors can be combined using a simple formula expression, or using condition formula expressions. Both interval values and interval conditions can be referenced. The result is interval-by-interval processing of the vector formula. The resulting final vector can be stored, TOU mapped, and/or subjected to a final set function (such as sum or max).

This rule stores service quantities of type "other" - meaning that the service quantities created by this rule do not have linkages to a specific Service Point or measuring component, given the potentially diverse sources of the data taken as input by this rule that lead to the final quantity.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-MATH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-Math

Copyright © 2026, Oracle and/or its affiliates.

Vector and Service Quantity Math Usage Calculation Rules and Anchor Time

The Vector and Service Quantity Math usage calculation rule supports usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

Post-Calculation Usage Calculation Rules

- Usage Calculation High/Low Rule
- Validate Against Tolerance
- Business Flag Hold

Usage Calculation High/Low Rule

This rule is used to validate the current usage against historical usage - previous year usage or previous usage. It ensures that any increase or decrease of the current usage relative to historical usage is within a tolerance.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-USGHIGLOW Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-UsageHighLowRule

Validate Against Tolerance

This rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-VALUSGTOL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

Copyright © 2026, Oracle and/or its affiliates.

D2-ValAgainstTol

Business Flag Hold

This rule can be used to place a hold on a usage transaction until the business flags related to the usage transactions service points can be investigated. The hold can be placed in one of two ways:

- **Calculation Window Based**: holds will remain until either a user manually bypasses the usage exception error or the current date is within a configurable tolerance of the calculation window end (aka the retry until date time).
- **Indefinite**: holds require a user to manually bypass the usage exception that is generated.

Defining which business flags should result in a usage transaction is a matter of identifying which business flag types, priorities, and confidences should be included.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-AUR-IBFH algorithm type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-BusinessFlagHold

Decision-Making Usage Calculation Rules

- Execute Usage Calculation Group
- Exception Handler

Execute Usage Calculation Group

This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. It is especially useful in cases where repeating the same rule over and over would be hard to maintain.

For example, you may want to calculate a straightforward kWh sum for every usage calculation group. Creating a separate CALC_KWH_SUM usage calculation group for this one calculation allows you to isolate the configuration points. Then the CALC_KWH_SUM usage calculation group can be referenced in every other usage calculation group that needs this sort of service quantity.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AUSGR-RFG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D1-UsageRuleReferredUsageGroup

Exception Handler

This rule evaluates the exception list that was accumulated during the execution of the calculation rules against exception criteria configured on this rule. This used to terminate the execution of usage calculation group processor if there are far too many exceptions hit during the execution.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-UREXPCHAN Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-UsageRuleExceptionHandler

Detailed Configuration Examples

Demand Calculation Options Using Interval Data

There are a number of ways to calculate a demand value from interval data when retrieving bill determinants for CIS. A few examples of demand calculations are listed below:

Scenario1: Use interval data for demand at a common block size: the Vector and Service Quantity Math usage calculation rule can be used to calculate demand from the interval data. Below is one example of configuration to handle demand calculation based on 30 minute blocks:

Usage Calculation Rule field	Value	Notes
Vector 1 // Type	Physical Channels Links to Usage Subscription	
Vector 1 // Unit of Measure	Kilowatt-Hours	Use your own UOM here - this is just as example.
Vector 1 // Service Quantity Identifier	Consumed	Use your own SQI here - this is just as example.

Copyright © 2026, Oracle and/or its affiliates.

Vector Processing // Common Interval Size	00:30:00	This configuration rolls up all channels (whether 5 minute, 15 minute, or 30 minute interval data) into a common interval size.
Vector Processing // Vector Formula Source	Simple Vector Formula	
Vector Processing // Simple Vector Formula	IV1*2	This is multiplying the common 30 minute interval size by 2 to achieve an hourly value.
Result // Unit of Measure	Kilowatts	Use your own UOM here - this is just as example.
Result // Service Quantity Identifier	Consumed	Use your own SQI here - this is just as example.
Result // Insert Primary SQ Entry	Yes	
Result // SQ Entry Quantity Source	Set Function Against Derived Vector	
Result // Set Function Against Derived Vector	Max	This function pulls the max value based on the configuration in the "Vector Processing" section.

Scenario 2: Use interval data for demand by TOU period: This example is very similar to the last and again leverages the Vector and Service Quantity Math usage calculation rule. However, instead of calculating a single demand value for the entire period it will calculate a demand value for each TOU period:

Usage Calculation Rule field	Value	Notes
Vector 1 // Type	Physical Channels Links to Usage Subscription	
Vector 1 // Unit of Measure	Kilowatt-Hours	Use your own UOM here - this is just as example.
Vector 1 // Service Quantity Identifier	Consumed	Use your own SQI here - this is just as example.
Vector Processing // Common Interval Size	00:30:00	This configuration rolls up all channels (whether 5 minute, 15 minute, or 30 minute interval data) into a common interval size.
Vector Processing // Vector Formula Source	Simple Vector Formula	
Vector Processing // Simple Vector Formula	IV1*2	This is multiplying the common 30 minute interval size by 2 to achieve an hourly value.
Result // Unit of Measure	Kilowatts	Use your own UOM here - this is just as example.
Result // Service Quantity Identifier	Consumed	Use your own SQI here - this is just as example.

Copyright © 2026, Oracle and/or its affiliates.

Result // Insert Primary SQ Entry	No	
Result // Apply TOU Map To Derived Vector	Yes	
Result // TOU Map	(your TOU Map)	Select the TOU Map you'd like to apply.
Result // Time Of Use Calculate Function	Max	This function pulls the max value based on the configuration in the "Vector Processing" section and then buckets it based on the selected TOU Map.

Usage Calculation Types

- Understanding Usage Calculation Types
- Configuring Usage Calculation Types

Understanding Usage Calculation Types

Usage calculation types define calculations executed at a customer level where there may be more than one calculation per usage subscription. For example, a customer may have a usage subscription for producing billing determinants but there may be additional calculations that will be executed on a much less frequent basis (annually, quarterly, etc). Usage calculation types provide a way to leverage a single usage subscription to drive multiple different calculations.

Usage calculation types use the following parameters:

- **Request Type**: Used to drive the creation and processing of the usage transactions for all usage subscriptions that are valid for the usage calculation type.

  > **Note**
  >
  > Request types for usage calculation types should be based on the Usage Calculation Request Type business object (D2–UsageCalculationReqType)

- **Service Point Quantity Creation**: Details of the quantities created by usage calculations of this type, including service point quantity type, and UOM, TOU, and SQI.
- **Applicable Usage Subscription Types**: A list of usage subscription types for which this usage calculation type is applicable. Each usage subscription type is defined by an effective date range, the usage transaction business object used for the usage calculation, the result UOM/TOU/SQI, and the usage calculation group used to perform the calculations.

Configuring Usage Calculation Types

You use the **Usage Calculation Type** portal to maintain and view usage calculation types.

Refer to Understanding Usage Calculation Types for more information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Calculation Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Type List**: This zone lists all usage calculation type records. Broadcast a record to display the details of the selected record.
- **Usage Calculation Type**: This zone displays details for the selected usage calculation type.

Usage Subscription Quantity Types

- Understanding Usage Subscription Quantity Types
- Configuring Usage Subscription Quantity Types

Understanding Usage Subscription Quantity Types

Usage subscription quantity types define types of quantities that can be stored for a usage subscription. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class. For example, a customer quantity may be used to store a scaling factor that describes how a given customer’s usage compares to the profiled usage for their rate class.

In a settlement implementation, usage subscription quantity types can be used to define values calculated monthly, annually, etc., such as annual peak load contribution (PLC).

Usage subscription quantity types use the following parameters:

- **Service Type**: The service type (electric, gas, water, etc.) for service point quantities of this type.
- **Usage Subscription Quantity Business Object**: The business object used for usage subscription quantities of this type.
- **Usage Subscription Quantity Identifiers**: Value identifiers related to the current usage subscription quantity type (used to provide shorthand descriptions of the various types of values measured by usage subscription quantities of this type).

Configuring Usage Subscription Quantity Types

You use the **Usage Subscription Quantity Type** portal to display and maintain usage subscription quantity types.

Refer to Understanding Usage Subscription Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Subscription Quantity Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Quantity Type List**: This zone lists all usage subscription quantity type records. Broadcast a record to display the details of the selected record.
- **Usage Subscription Quantity Type**: This zone displays details for the selected usage subscription quantity type.

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Option Types

- Understanding Dynamic Option Types
- Configuring Dynamic Option Types

Understanding Dynamic Option Types

Dynamic options specify terms that override how usage is normally calculated, such as a critical peak period that affects the TOU mapping of interval consumption. Dynamic option types store information common to dynamic options of a specific type.

There are two classes of dynamic options:

- **Interval Set**: Used to represent dynamic option types that provide a set of intervals that will be included in a calculation.
- **Usage Event**: Used to represent dynamic option types that provide a specific period of time during which a dynamic option is applicable.

Interval Set dynamic option types also define key information about the interval data they represent, including:

- **Unit of Measure**: Identifies the quantity that Interval Set dynamic options and Interval Set dynamic option events based on this type are measuring.
- **Interval Size**: Defines the size of the intervals for Interval Set dynamic option events of this type, represented as hours:minutes:seconds (HH:MI:SS).

Configuring Dynamic Option Types

This portal is used to display and maintain a Dynamic Option Type.

Refer to Understanding Dynamic Option Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Dynamic Option Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Dynamic Option Type List**: displays all of the Dynamic Option Types so you can choose the one you want to display in more detail
- **Dynamic Option Type**: shows the specific configuration for the selected Dynamic Option Type

Bill Cycle

- Understanding Bill Cycles
- Configuring Bill Cycles

Understanding Bill Cycles

Copyright © 2026, Oracle and/or its affiliates.

Bill cycles identify when a customer is going to be billed and when bill determinants are going to be calculated.

> **Note**
>
> Meter Data Management bill cycles are created from Customer Care and Billing bill cycles via Customer To Meter Master Data Synchronization.

Configuring Bill Cycles

This portal is used to display and maintain a Bill Cycle.

Refer to Understanding Bill Cycles for more information.

You can access the portal from other objects such as Measurement Cycles.

The following zones may appear as part of the portal's **Main** tab page:

- **Bill Cycle List**: displays all of the Bill Cycles so you can choose the one you want to display in more detail.
- **Bill Cycle**: shows the specific configuration for the selected Bill Cycle.
- **Bill Cycle List**: lists the bill cycle schedules of the current bill cycle.

> **Note**
>
> Meter Data Management bill cycles are created from Customer Care and Billing bill cycles via Customer To Meter Master Data Synchronization. Bill cycles should only be updated via the Customer Care and Billing Bill Cycle portal.

Defining Time of Use Options

- Time of Use Groups
- Time of Use Map Templates
- Time of Use Map Types

Time of Use Groups

- Understanding Time of Use Groups
- Configuring Time of Use Groups

Understanding Time of Use Groups

Time of Use (TOU) Groups are groups of TOUs that are associated to TOU templates that limit the TOUs available for use in a TOU schedule. These TOUs will be available as the default, holiday, and schedule TOUs within the template.

Copyright © 2026, Oracle and/or its affiliates.

Each TOU in the group is given a priority. The priority is important in a few ways:

- The TOU Overlay 360 Degree zone uses these priorities to decide which TOUs will be rendered as a unique shade and which will fall into a category of "other" when the maximum number of distinct TOUs to graph has been reached.
- These priorities are available for use in customized logic.

Configuring Time of Use Groups

This portal is used to display and maintain a TOU Group.

Refer to Understanding Time of Use Groups for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Group**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Group List:** This zone lists all TOU Group records. Broadcast a record to display the details of the selected record.
- **TOU Group:** This zone provides information about the selected TOU Group.

Time of Use Map Templates

- Understanding Time of Use Map Templates
- Configuring Time of Use Map Templates

Understanding Time of Use Map Templates

Every TOU map references a TOU map template that defines the rules for generating TOU data from that TOU map. Specifically, TOU map templates define:

- The TOU group (defines the valid TOU periods for the template) used for the TOU map
- The default TOU period used for periods not explicitly defined. (This means you don't have to specify dates and times for all periods. For example, if your default TOU period is "Off Peak" you only need to define dates and days and times for On Peak or other TOU periods.)
- The specific date ranges, days of the week, and time periods designated for each TOU period.

The system periodically generates TOU map data for TOU maps by interpreting the rules defined template.

Attributes used to define TOU map templates include the following:

- **TOU Group**: the TOU group used by the map template Default TOU: the default TOU for the map template (from the TOU Group). This is the TOU used when creating TOU map data for dates not accounted for in the TOU Schedules section.
- **Work Calendar**: the work calendar used to identify holidays. On each holiday the Holiday TOU will be used. For additional information see Defining Work Calendar in the Oracle Utilities Application Framework *Administrative User Guide*.
- **Holiday TOU**: the TOU used for holidays (from the TOU Group)
- **Holiday Template**: the TOU map template used for holidays (if applicable)

Copyright © 2026, Oracle and/or its affiliates.

- **Interval Size**: the size of the intervals for TOU map data created from the map template, represented as hours:minutes:seconds (HH:MI:SS).
- **TOU Schedules**: date ranges (including month, day, and time ranges) and which TOU periods apply to each

Refer to Understanding Referencing Master Data by Identifiers for information on how admin configuration can reference a TOU map by a TOU template to ease migration of data between environments.

TOU Map Template Interval Size

TOU map templates can also specify an interval size (in seconds-per-interval, or SPI). This value specifies the duration of the individual TOU map data records, and also controls the values allowed in the Start and End Times. For example, if a TOU map template sets the interval size at 15 minutes, Start and End times must be in units of the interval size (10:00, 10:15, 10:30, etc.).

A TOU map template can be used to generate TOU map data for TOU maps whose SPI is divisible by the template's SPI. For example, a 60 minute template can be used to generate TOU data for TOU maps with SPIs of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every SPI.

Holidays

Many utilities categorize consumption on holidays differently than on the day of week on which the holiday falls. For example, holiday consumption might be categorized as Off-Peak regardless of the day it falls on. TOU map templates can define rules for different TOU periods for holidays in two ways. Both options require that the template references a Work Calendar that identifies each of the holidays throughout the year. In addition they require either:

- A Holiday TOU that will be used for each holiday for the duration of that holiday (e.g. Off Peak)
- A Holiday TOU Map Template that defines the TOUs that should be used for holidays that fall during different seasons within the year (e.g. Off Peak Summer, Off Peak Winter)

Important Time of Use Template System Events

The time of use template supports the following business object algorithm system events:

- **Derive TOU**: receives a date time and determines the TOU code for that date time based on the configuration of the time of use template schedule. See the Derive Time Of Use For Date Time (D2-DERTOU-DT) algorithm type as an example.

Configuring Time of Use Map Templates

This portal is used to display and maintain a TOU Map Templates.

Refer to Understanding Time of Use Map Templates for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Map Template**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map Template List:** This zone lists all TOU Map Template records. Broadcast a record to display the details of the selected record.
- **TOU Map Template:** This zone provides information about the selected TOU Map Template.

Copyright © 2026, Oracle and/or its affiliates.

Time of Use Map Types

- Understanding Time of Use Map Types
- Configuring Time of Use Map Types

Understanding Time of Use Map Types

TOU Map Types define important properties of TOU maps of the type, including the interval size and the valid TOU map templates.

Attributes used to define TOU map types include the following:

- **Time Zone**: the time zone to be used when generating the TOU map data. Refer to Multiple Time Zone Support for more information about how time zones impact TOU map data in multiple time zone environments.
- **Interval Size**: the size of the intervals for TOU map data created from maps of this type, represented as hours:minutes:seconds (HH:MI:SS). The interval size cannot be larger than the interval size defined on the Default TOU Map Template or any of the Override TOU Map Templates.
- **Default TOU Map Template**: the default TOU map template used by maps of this type
- **Override TOU Map Templates**: one or more TOU map templates that can be used as an override on TOU maps of this type.

TOU Map Type Interval Size

The SPI of a TOU map must divide evenly into the SPI of any measuring component that uses the map (because the system joins the date/time of the measurement to the date/time of the TOU data). This means that it is possible to use a 15 minute TOU map with a 60 minute measuring component. However, it is not OK to have a 60 minute TOU map used with a 15 minute measuring component because the join will miss 3 out of 4 measurements.

However, it is important to note that the TOU mapping process is at its most efficient when the measurement data that is being mapped is of the same interval size as the TOU schedule it is being mapped against. When there are differences in the interval size the process must first convert the measurement data into the appropriate interval size for the TOU Map prior to applying the TOU Map.

This means that for each TOU Map Template you should have sufficient TOU Map Types to cover the various interval sizes that will be supported by your measurement data. For example, if you have measuring components with interval sizes of 15 minutes, 30 minutes, and 60 minutes then for each TOU Map Template there should be TOU Map Types with interval sizes of 15 minutes, 30 minutes, and 60 minutes.

Default and Override TOU Map Templates

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support a fallback/override pattern used in other areas of the system.

- A TOU map's TOU map type defines the default (or "fallback") TOU map template that's used to generate its TOU data.
- A TOU map's type defines the TOU map templates that can be referenced on individual TOU maps to override the "fallback" template.

Copyright © 2026, Oracle and/or its affiliates.

- An individual TOU map can have an override template. If the TOU map doesn't have an override template, the fallback template defined on the TOU map type is used to generate the map's TOU data.

Important Time of Use Map System Events

The TOU Map business object that is associated to a given TOU map type supports a special system event that is used in the generation of TOU map data:

- **Create TOU Map Data**: receives a date range and for that date range it will create the appropriate TOU map data for the TOU map. See the Create TOU Map Data (D2-CRETMD-CT) algorithm type as an example.

Configuring Time of Use Map Types

This portal is used to display and maintain a TOU Map Types.

Refer to Understanding Time of Use Map Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Map Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map Type List:** This zone lists all TOU Type records. Broadcast a record to display the details of the selected record.
- **TOU Map Type:** This zone provides information about the selected TOU Map Type.

Defining Premise and Service Point Options

A premise is where a customer consumes the services supplied by your company. Located at a premise are the various devices that consume energy and measure consumption. Before you can define premises and devices, you must set up the control tables defined in this section.

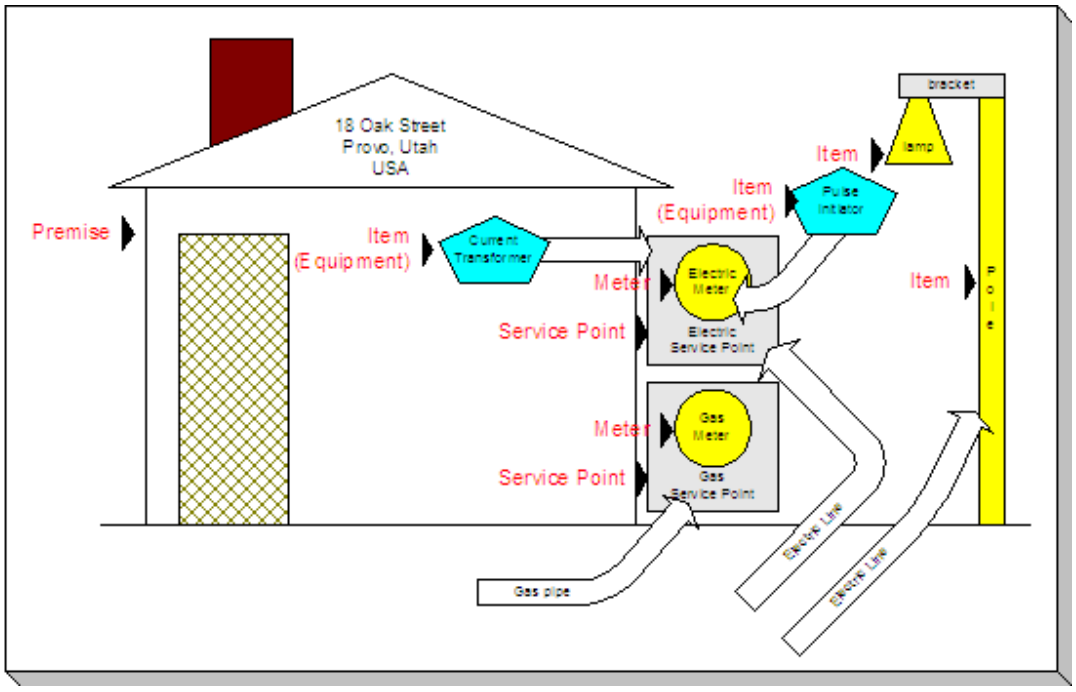
> **Warning**
>
> The topics in this section do not describe every table that must exist in order to set up premises and service points. Many premise and service point control tables are described in Defining Device Options, and Defining Service Order Options. The tables described in this section are those that must be set up regardless of the type of service.

- An Illustration Of A Premise
- Setting Up Premise Options
- Setting Up Generic Service Point Options
- Setting Up Premise & Service Point Postal Defaults
- Designing SP Types

Copyright © 2026, Oracle and/or its affiliates.

An Illustration Of A Premise

The following picture illustrates a premise with 2 service points, 2 meters, and 2 badged items:



The following concepts are illustrated above:

Premise A premise describes a location at which your company supplies some type of service. In addition to the obvious address information, a premise also contains geographic coordinates, meter read instructions, and taxation jurisdiction information.

> **NOT_SUPPORTED**
>
> For more information about the control tables that must be set up before you can define a premise see Setting Up Premise Options.

Service Point A service point (SP) is a geographic location at which service(s) are delivered to a premise. The SP record maintains information about the type of service, the measurement cycle (if the service is metered), the field office responsible for maintaining the service, the distribution company that supplies the service, etc.

There are three major categories of service points:

- Those where the rate of consumption and the total amount of consumption is measured (e.g., electricity, gas, water) by a meter. You can think of this type of service point as a "socket" into which a meter can be plugged. Over time, many meters may be plugged into the socket. We refer to these types of service points as metered.
- Those that hold badged items. A badged item is a physical device with a unique identity (e.g., a specific street light, a specific hydrant). You can think of this type of service point as

Copyright © 2026, Oracle and/or its affiliates.

a "socket" into which a badged item can be plugged. Over time, many items may be plugged into the socket. We refer to these types of service points as item-based.

- Those used to hold one or more non-badged items. For example, if your organization doesn't badge street lamps, you can use a single service point to hold an infinite number of lamps. We refer to these types of service points as non-badged.

> **NOT_SUPPORTED**
>
> Refer to Service Points (SPs) for more information about non-badged items.

An unlimited number of SPs may exist at a premise. However in reality, the number of SP's is related to the number of services supplied by your company. For example, an electric and gas company will typically have two SPs per premise.

> **NOT_SUPPORTED**
>
> For more information about the control tables that must be set up before you can define service points refer to Setting Up Generic Service Point Options, Defining Device Options.

Field activities may be dispatched to all types of service points.

Meter A meter is a physical device used to measure the amount of gas, water, or electricity used by a customer. While most meters measure consumption in a single unit of measure (e.g., gallons, cubic feet, kilowatt-hours), some electric meters are extremely sophisticated and measure several different factors. For example, some electric meters measure how much was used, when it was used, the efficiency of consumption, the maximum amount used, and a few other unusual things.

> **NOT_SUPPORTED**
>
> For more information about the control tables that must be set up before you can define a meter, refer to Defining Device Options, and Setting Up Consumption Estimation Parameters.

Item An item that is not consider to be "equipment" is a physical device that does NOT measure consumption, but impacts billing in some way (i.e., there are charges in your rates based on the number and type of items installed at a service point). Examples include street lights, light poles, and security cameras. Items are related to service points and a service point can have one or more items linked to it.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about the control tables that must be set up before you can define an item, refer to Configuring Device Types.

For more information about premises and service points, refer to Understanding The "V".

Setting Up Premise Options

This section describes tables that must be set up before you can define premises.

- Defining Premise Types
- Implementing Address Validation

Defining Premise Types

Open **Admin**, and then **Geographic**, and then **Premise Type** to define the premise types used to categorize your premises.

Description of Page

Enter a unique **Premise Type** and a **Description** for every premise type.

Allow CIS Division controls whether premises of this type can use the CIS Division on the premise page. Valid values are Not Allowed, Optional, and Required.

If your implementation requires restricting data by CIS Division, it is recommended that Allow CIS Division be set to Required.

Implementing Address Validation

In order to set up address validation, you must ensure that the following Option Types are defined with their corresponding values on the General System Configuration – Feature Configuration page:

Option Type	Description
Address Validation Script	Indicates the name of the BPA script to invoke when a user clicks on the validate address button. This option is required if the Allow Address Validation option is set to Y. The base product includes sample script C1-ValAddr that uses Oracle Spatial procedures and functions to geocode an address and return its matching quality code.
Allow Address Validation	Indicates whether an integration with address validation software is implemented. If set to Y, the system will render a validate button along with the address fields displayed throughout the application to trigger address validation. Valid values are 'Y' and 'N'. If set to Y, the Address Validation Script to invoke must be specified.

Copyright © 2026, Oracle and/or its affiliates.

As described in the above table, the base product includes sample script C1-ValAddr uses Oracle Spatial procedures and functions to geocode an address and return a matching quality code. To use this script, ensure that the Oracle Spatial Geocoding (F1-ORAGEOCD) algorithm is plugged in on installation options for the geocoding service system event. A data source containing the geocoding data must also exist on the database.

If your implementation uses something other than Oracle Spatial for address validation, you'll need to create a new address validation BPA script to specify on the General System Configuration feature configuration. For additional information about creating this script, see the steps descriptions for the C1-ValAddr script in the application. The new script should contain the following move step to update the page:

- If the calling page is an old style page, the move from the BPA to the page can be achieved by specifying a move step from a source field to a User Interface Field.
- If the calling page is a display map, the move from the BPA to the page can be achieved by specifying a step to read the Business Object, modify the values as returned by the validation software, and then update the BO.

Setting Up Generic Service Point Options

This section describes tables that must be set up before you can define service points.

- Setting Up Service Point Types

Setting Up Service Point Types

Every service point must reference a service point (SP) type. The SP type controls almost all aspects of the service point's behavior (e.g., the type of field activity that may be dispatched to it, the type of service agreement that may be linked to it, the type of device that may be installed at it).

The topics in this section describe the windows used to set up your SP types.

> ⚠ **Warning**
>
> Setting up SP types requires careful analysis of your company's SA types, field activities, and its consumption estimation philosophy. Refer to Designing SP Types for more information about this design process.

Setting Up Premise & Service Point Postal Defaults

You set up postal defaults if your company is able to default field values onto new premises and service points based on the premise's postal code. The topics in this section describe how to maintain postal defaults.

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> For more information about where these default values are used, refer to Maintaining Premises and Maintaining Service Points.

- Postal Defaults - Main
- Postal Defaults - Service Default

Postal Defaults - Main

To define premise postal defaults, open **Admin**, and then **Geographic**, and then **Postal Code Default**, and then **Add.**

Description of Page

Enter the **Country Code** and range of postal codes to which the default values apply using the **From Postal** and **To Postal**.

> **Note**
>
> You may not have postal defaults whose from / to postal codes overlap.

Trend Area is not applicable for Customer to Meter.

Enter the **County** to be defaulted onto new premises located in this postal code range.

Enter the **City** to be defaulted onto new premises located in this postal code range.

Enter the **CIS Division** to be defaulted onto new premises located in this postal code range.

Enter the **State** to be defaulted onto new premises located in this postal code range.

Enter the **Time Zone** to be defaulted onto new premises located in this postal code range.

Use the **Characteristic Types and Values** collection to define the **Characteristic Type**s and their respective **Characteristic Value**s to be defaulted on premises located in this postal code range. In addition to providing interesting information, these characteristics may also determine the prices and tax rates on the bills generated for the services consumed at a premise.

> **NOT_SUPPORTED**
>
> For more information about characteristics, see Setting Up Characteristic Types & Their Values and An Illustration Of A Bill Factor And Its Characteristics .

Use the **Geographic Types and Values** collection to define the **Geographic Type**s and their respective **Value**s to be defaulted on premises located in this postal code range. In addition to providing interesting information, these values may be used to sort field activities in geographic value order.

Where Used

Copyright © 2026, Oracle and/or its affiliates.

This information is defaulted when a new Premise is added. Characteristics and geographic values are also defaulted when the postal code for a Premise is changed. Refer to Maintaining Premises for more information.

Postal Defaults - Service Default

To define values to be defaulted for a service point located in a postal range, open **Admin**, and then **Geographic**, and then **Postal Code Default**, and then **Search** and navigate to the **Service Defaults** page.

> **NOT_SUPPORTED**
>
> For more information about where these default values are used, refer to Maintaining Service Points.

Description of Page

Use the **Service Defaults** collection to define values to be defaulted on service points located in a given postal code range belonging to a given **Service Type** (note, a service point's Service Type comes from its SP Type).

Stock Location is not applicable for Customer to Meter.

- Use the **Characteristic Types and Values** collection to define the **Characteristic Type**s and their respective **Characteristic Value**s to be defaulted on new service points located in this postal code range. In addition to providing interesting information, these characteristics may also determine the prices and tax rates on the bills generated for the services consumed at a service point.
- Use the **Geographic Types and Values** collection to define the **Geographic Type**s and their respective **Value**s to be defaulted on new service points located in this postal code range. In addition to providing interesting information, these values may be used to sort field activities in geographic value order.

Designing SP Types

Every service point must reference an SP type. When you set up an SP type, you define how the system manages many aspects of its service points' behavior.

> **Note**
>
> **Perfect foresight.** In a perfect world, the other control tables would have been set up with perfect foresight of setting up your SP types. In reality, setting up your SP types may invalidate some of your earlier decisions. Don't feel bad if this happens, some amount of iteration is natural.

Copyright © 2026, Oracle and/or its affiliates.

Designing your SP types is an iterative process. To minimize the number of iterations, we recommend using the steps outlined in this section to complete the following table. When the table is complete, you're ready to set up your SP types.

SP Type	Service Type

- Service Segmentation
- Device Segmentation
- SA Type Segmentation

Service Segmentation

At a minimum, you will have one SP type for every different type of service that exists at your premises. If we assume that your organization sells gas, water, waste water and electricity services, you will need four SP types.

SP Type	Service Type
GAS	Gas service
WATER	Water service
WASTE WATER	Waste water service
ELECTRIC	Electric service

Device Segmentation

For each service, determine if there exist meters, badged items, or non-badged items at the service points. For example, if we assume:

- Electric service has service points with meters, badged lamps, and parking lots.
- Gas and water services just use meters.
- Waste water service doesn't actually have a meter installed at it (it uses the water consumption measured by the water service point's meter)

Then your SP types will be:

SP Type	Service Type	SP Sub Type
GAS - METERED	Gas service	Meter
WATER - METERED	Water service	Meter
WASTE WATER	Waste water service	Unbadged
ELECTRIC - METERED	Electric service	Meter
ELECTRIC - BADGED LAMP	Electric service	Item
ELECTRIC - PARKING LOT	Electric service	Unbadged

Copyright © 2026, Oracle and/or its affiliates.

SA Type Segmentation

Every SP Type whose service points can be linked to a service agreement has one or more SA types. These define which type of service agreements can pay for the service point's service. If different service points have different valid SA types, you will need to split the SP types accordingly.

The SA Type segmentation of SP Types is the most complicated design decision you'll have to make. Unfortunately, the decision making process is subjective and iterative. The iterations are caused by the fact that the number of SA types is dependent on the number of SP types (and vice versa). We recommend the following to help work your way through this conundrum:

- Design your SA types using the information in Defining Service Agreement Types.
- Return to your SP types and determine if, given the proposed SA types, you can define a list of valid SA types for each SP type. If you find the population of SA types (and their valid rates) could result in invalid rates paying for service at a service point, divide your SA types further.

Defining Bill and Measurement Cycles

This chapter is dedicated to issues related to defining cycles, routes and schedules in the system.

- Defining Bill and Measurement Cycles
- Defining Statement Cycles

Defining Bill and Measurement Cycles

Every account references a bill cycle. An account's bill cycle controls when it is billed.

Every manually read metered service point references a measurement cycle. A service point's measurement cycle controls when a service point's meter is measured.

The design of your measurement and bill cycles is inextricably linked because you probably want to bill your customers shortly after their meter devices are measured.

In this section, we describe how to design and set up these cycles. In addition, we discuss how to set up bill period schedules. These are used to define the bill segment end date for special types of non-metered service agreements.

> **Note**
>
> **Recommendation.** We recommend reviewing Bill Frequency - Bill Cycle vs Bill Segment Duration before setting up this information.

- The Big Picture of Bill Cycles, Measurement Cycles, and Bill Periods

Copyright © 2026, Oracle and/or its affiliates.

- Bill Cycles
- Designing Bill Periods
- Setting Up Bill Cycles
- Setting Up Service Cycles And Measurement Cycle Schedules
- Setting Up Measurement Cycle Schedules
- Setting Up Bill Periods

The Big Picture of Bill Cycles, Measurement Cycles, and Bill Periods

The topics in this section provide background information about a variety of bill cycle, measurement cycle, and bill period issues.

- Designing Cycles for Metered Services

Designing Cycles for Metered Services

The topics in this section provide background information about a variety of measurement cycle issues.

- A Description of the Cyclical Meter Read Process
- Designing Measurement Cycles, Routes, And Schedules

A Description of the Cyclical Meter Read Process

Manual meter readers using handheld devices record most meter reading measurements in the field. These meter reads are uploaded into the system for use by Oracle Utilities Meter Data Management.

A service point's installed meter device is measured due to the following data relationships:

- Every metered service point references a schedule route.
- A route references a measurement cycle.
- A measurement cycle has measurement cycle schedules that define when the service points in the cycle are measured.
- Every measurement cycle schedule contains two dates:
  - **Scheduled selection date.** This is the date Meter Data Management could use to select the service points in the cycle for download to your handheld software via custom algorithms to create handheld activities when pending measurement cycle schedules are proposed (refer to Measurement Cycle and Bill Determinants for more information).
  - **Expected work date.** This is the date the meter is expected to be read.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Overriding a route within a specific schedule.** Rather than downloading all routes within a cycle, you can set up the system so that only specific routes are downloaded on any given date. This is a very powerful feature. You can use it, for example, to estimate specific cycles every other month or indicate the customer reads the meter every third month. Refer to Designing Measurement Cycles, Routes, And Schedules for more information.

- On the scheduled selection date, Meter Data Management could create a download file, via custom algorithms, containing information to be sent to your handheld software. This download file should contain information about every register or measuring component on every meter in the routes being downloaded.
- Your handheld software distributes this information to the handheld devices and then your meter readers do their job. When they finish, the resultant measurements are uploaded into the system for subsequent use by other processes such as usage transactions for billing.

> **Warning**
>
> It is very important to create a measurement cycle schedule for every expected read date regardless of whether the cycle's routes are downloaded. Why do you have to do this? Because Meter Data Management uses the expected work date on the measurement cycle to know when to look for a measurement during usage calculations. If it can't find a reading on or near this date, Oracle Utilities Meter Data Management may estimate consumption. Without a measurement cycle schedule, MDM wouldn't know when to look for readings. So, for example, if you estimate a given cycle's consumption EVERY OTHER MONTH, you must create a measurement cycle schedule for EVERY month. On each month's schedule, you must define if the routes should be downloaded.

> **NOT_SUPPORTED**
>
> For more information about the how to control when a cycle is read, refer to Setting Up Measurement Cycle Schedules.

Designing Measurement Cycles, Routes, And Schedules

The topics in this section provide information describing how to design your measurement cycles, routes, and schedules.

- Designing Measurement Cycles For Meter Reading
- Designing Schedule Routes For Meter Reading
- Designing Metered Measurement Cycle Schedules

Designing Measurement Cycles for Meter Reading

Copyright © 2026, Oracle and/or its affiliates.

The criterion that affects the number of measurement cycles has nothing to do with when meter readers physically read your meter devices. Rather, the frequency that you bill the meter's consumption (real or estimated) is what controls the number of measurement cycles.

So, for example, if you bill every month, but read every OTHER month, you'll have 20 measurement cycles - one for each bill day during the month. If you bill bimonthly, you'll have 40 measurement cycles. If you bill quarterly, you'll have 60 measurement cycles. Etc.

> **Note**
>
> **Different billing frequencies are possible for different service points.** If you have different billing frequencies for your different types of metered service, you'll need a different set of cycles for each billing frequency. For example, if you bill water quarterly and electricity monthly, you'll have 20 bill cycles (one for each bill day during a month), but you'd need 60 quarterly measurement cycles for your water service points, and 20 monthly measurement cycles for your electric service points. This would result in a customer getting billed every month. However, four times a year, their bill would contain a charge for both electricity and water.

Top of the Page

Designing Service Routes For Meter Reading

The following points describe the relationship between a meter read, a route and a measurement cycle:

- A measurement cycle contains one or more routes.
- A route has one or more service points.
- A service point holds an installed meter device.
- And a device is what is measured.
- Therefore, the number of devices a person can read in a day limits the number of service points in a route.

> **Warning**
>
> If your company supplies electric service and uses MV90's, you will need to take advantage of "notional" routes. A "notional" route's service points are never actually read by a human. Rather, the service points' consumption is fed to the system by a sophisticated device (e.g., an MV90). These service points still need to be linked to a route because usage calculation is dependent on the route's cycle to determine the expected meter read date.

Top of the Page

Designing Metered Measurement Cycle Schedules

The process of designing your measurement cycle schedules is either easy or complicated. It will be easy if all routes within a cycle are downloaded when the measurement cycle is scheduled for download. It will be complicated if you download a subset of routes within a cycle on any given download date. We'll provide a few examples to help explain why.

Copyright © 2026, Oracle and/or its affiliates.

If you download all cycles in a route whenever the cycle is downloaded, your measurement cycle schedule will look as follows.

> **Note**
>
> **Bill cycles.** We've included each measurement cycle's related bill cycle to help you understand when the measurement cycle's consumption will be billed. Bill cycles are discussed in Bill Cycles.

Measurement Cycle	Download Date	Expected Work Date	Which Routes To Download	Bill Cycle	Bill Window	Estimation Date
1	28-May-99	31-May-99	All	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	All	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	All	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	All	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc. to 20						

Now let's complicate life. If we assume you physically read your routes every other month (but bill monthly using estimated consumption), then you'll need the following measurement cycle schedule.

Measurement Cycle	Download Date	Expected Work Date	Which Routes	Bill Cycle	Bill Window	Estimation Date
1	30-May-99	31-May-99	1, 2, 3 - Download 4, 5, 6 - Estimate	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	1, 2, 3 - Download 4, 5, 6 - Estimate	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc. to 20						
1	29-Jun-99	30-Jun-99	1, 2, 3 - Estimate 4, 5, 6 - Download	1	30-Jun-99 to 2-Jul-99	2-July-99

Copyright © 2026, Oracle and/or its affiliates.

2	30-Jun-99	1-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download	2	1-July-99 to 3-Jul-99	3-July-99
3	1-July-99	2-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download	3	2-July-99 to 4-July-99	4-July-99
4	2-July-99	3-July-99	1, 2, 3 - Estimate 4, 5, 6 - Download	4	3-July-99 to 7-July-99	7-July-99
Etc. to 20						

Notice the following:

- You still have 20 measurement cycles even though you only read the meters every other month. Why? Because Oracle Utilities Meter Data Management uses the expected work date on the measurement cycle to know when to look for a measurement. If it can't find a reading on this date, Meter Data Management estimates consumption. Without a measurement cycle schedule, Meter Data Management wouldn't know when to look for readings.
- Every other month you download half the routes in each cycle and estimate consumption for the other half.
- If you don't download all routes when a measurement cycle is scheduled, you have to indicate how to handle every route in the cycle.

The above design is infinitely flexible. You can use it to handle any number of requirements:

- Estimate consumption every seventh month.
- Bill every month, but only read once a quarter.
- Etc.

Bill Cycles

The topics in this section provide background information about a variety of bill cycle features.

- The Cyclical Billing Process & Window Billing
- Designing Your Bill Cycle
- The Relationship Between Metered Measurement Cycles and Bill Cycles
- How Does An Account Get Its Bill Cycle?
- Protecting An Account's Bill Cycle
- What Happens If An Account Has Service Agreements Spanning Metered Measurement Cycles?

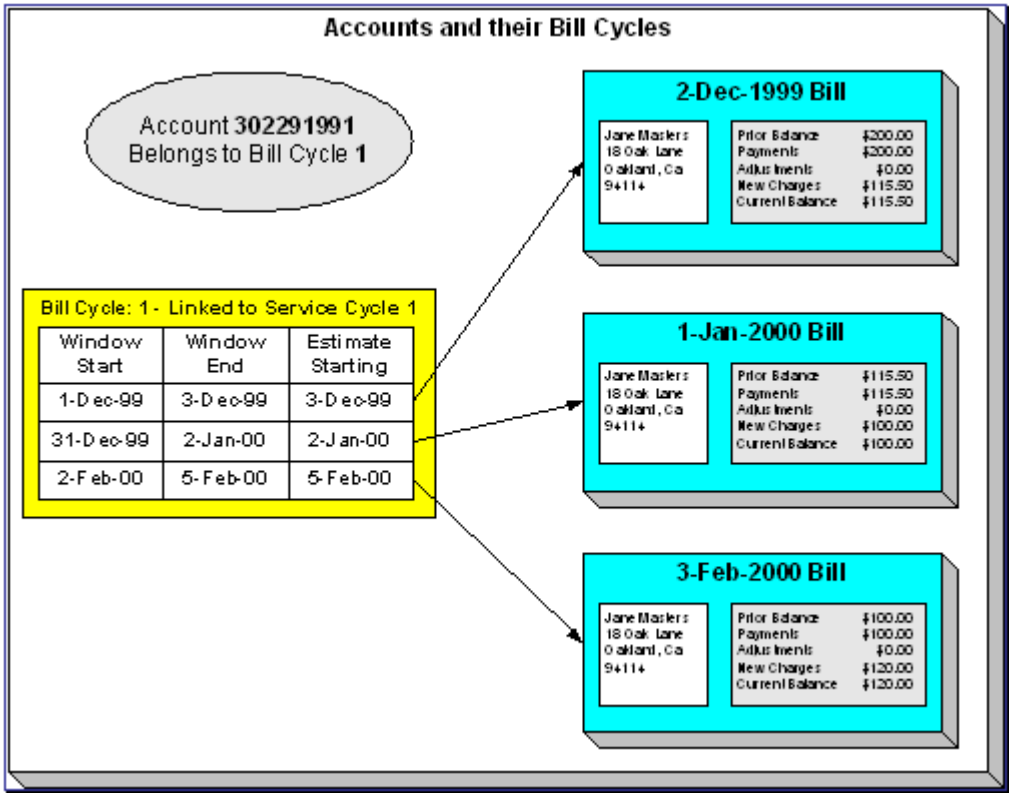
The Cyclical Billing Process & Window Billing

Copyright © 2026, Oracle and/or its affiliates.

The cyclical bill creation process creates most bills. This process works as follows:

- Every account references a bill cycle. The bill cycle's schedule controls WHEN the system attempts to create bills for the account.
- Every bill cycle has a bill cycle schedule that defines the dates when a cycle's accounts are to be billed. Rather than attempt to create bills on one evening, most bill cycles use a concept of "window billing" where the system attempts to produce bills for accounts over a few nights. The first night (i.e., the day the window opens) should be the earliest day on which meter reads for the account can enter the system. The last night (i.e., the day the window closes) should be the last possible day in which a meter read can enter the system. This concept of window billing allows you to start billing accounts on the earliest possible day and then bill the stragglers over successive evenings. This results in much better cash flow.
- When the bill cycle creation process runs, it looks for bill cycles with open bill windows. It then attempts to create bills for the accounts in each such cycle. If a bill is created, it will send it out that evening. If a bill cannot be created, the system will create a bill in the "error" state with a message that can be analyzed by your billing staff. The next day, your staff can either correct the problem or not. When the bill cycle creation process next runs, it deletes all "error" bills and attempts to recreate them. It continues this throughout the bill window. If bills are in error at the end of the window, they will remain in this state until a user fixes them. If the bills are still in error when the cycle's next window opens, a billing error will be generated.

The following diagram should help clarify the above.



Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about the how to control when bills are produced for a cycle, refer to Bill Cycle - Bill Cycle Schedule.

Designing Your Bill Cycle

The number of bill cycles is determined by the frequency that you bill your customers. So, for example, if you bill every month, you'll have 20 bill cycles - one for each bill day during the month. If you bill bimonthly, you'll have 40 bill cycles. If you bill quarterly, you'll have 60 bill cycles. Etc.

Keep in mind the following:

- You may need additional bill cycles if you allow customers to be billed off-cycle. For example, you could create a bill cycle called "Seniors" and link this to every senior citizen. You would set up this bill cycle's schedule to create bills shortly after social security checks are issued.
- You may need other bill cycles for customers with non-metered services (e.g., land leases, one time invoices).

Important

An account's bill cycle should attempt to create bill segments at least as often as the shortest service agreement duration. For example, if an account has both monthly and quarterly service agreements, the account should be placed on a monthly bill cycle. Refer to Bill Frequency - Bill Cycle vs Bill Segment Duration for more information.

The Relationship Between Metered Measurement Cycles and Bill Cycles

As you can deduce, a service point's measurement cycle is related to an account's bill cycle.

The following table is an example of how you would set up the various dates on the various schedules that control meter reading and billing.

Measurement Cycle	Download Date	Expected Work Date	Bill Cycle	Bill Window	Estimation Date
1	28-May-99	31-May-99	1	31-May-99 to 2-Jun-99	2-Jun-99
2	31-May-99	1-Jun-99	2	1-Jun-99 to 3-Jun-99	3-Jun-99
3	1-Jun-99	2-Jun-99	3	2-Jun-99 to 4-Jun-99	4-Jun-99
4	2-Jun-99	3-Jun-99	4	3-Jun-99 to 7-Jun-99	7-Jun-99
Etc.					

Copyright © 2026, Oracle and/or its affiliates.

Notice the following about this example:

- The bill cycle code is the same as the measurement cycle. This is not necessary, it's just good practice.
- The bill window starts on the first date on which a meter read could be uploaded.
- The bill window ends on the day after the last possible day a read could be uploaded for the bill cycle.
- Oracle Utilities Meter Data Management only estimates consumption on the last day of the bill window.

NOT_SUPPORTED

For more information about the bill cycle schedule, refer to Setting Up Bill Cycles.

How Does An Account Get Its Bill Cycle?

Most accounts are created behind-the-scenes when a user uses the "add account" option on Person - Main Information. An account created like this doesn't have a bill cycle. Rather, it sits in limbo awaiting the activation of its first service agreement. When a service agreement is activated, the system populates the account's bill cycle using the following algorithm:

- If an account's bill cycle is protected, the activation of a service agreement will not affect an account's bill cycle. Refer to Protecting An Account's Bill Cycle for more information.
- If the service agreement being activated is for metered service, the account is given a bill cycle that will generate bills shortly after the service point is read. The route the system follows to get this bill cycle is a bit indirect:
  - A metered service agreement references one or more metered service points.
  - Every metered service point references a measurement cycle (the measurement cycle controls when the meter at the service point is read).
  - Every measurement cycle references a default bill cycle. It is this bill cycle that is populated on the account paying for service at the service point.
- If the service agreement being activated is for metered service and the associated service point is externally managed along with its measurement cycle and bill cycle, an external call is made to obtain the bill cycle from Meter Data Management. This bill cycle is assigned to the service agreement’s account.
- If the service agreement being activated is not metered, the system cannot populate the account's bill cycle because there is no measurement cycle from which to default the bill cycle. This means the account's bill cycle will be blank until a user specifies a bill cycle for the account (using the Account page).

Note

A To Do entry highlights accounts without a bill cycle. A To Do entry highlights those accounts that require a user to specify a bill cycle. This entry is generated for accounts without a bill cycle that have active service agreements.

Copyright © 2026, Oracle and/or its affiliates.

When the last service agreement linked to an account is stopped, the account's bill cycle will be changed so that the account will be final billed when billing next executes. Refer to What Happens At Finalization Time? for more information.

When a service point's measurement cycle is updated, and the account's bill cycle is not protected, the system automatically updates the account's bill cycle to be in sync with the measurement cycle. Note that this will only take place if the Allow Bill Cycle Synchronization **Option Type** on the General System Options Feature Configuration is set to Y.

When an externally managed service point's measurement cycle is updated that results to a change in the bill cycle, a change notification is sent by Meter Data Management to trigger an corresponding update of the associated account's bill cycle in Customer Care and Billing. Note that this will only take place if the Allow Bill Cycle Synchronization **Option Type** on the General System Options Feature Configuration is set to Y.

Protecting An Account's Bill Cycle

Over time, as a customer moves from premise to premise, their bill cycle is changed behind-the-scenes to be in sync with the latest service point's measurement cycle (as described in the previous section). If you do not want the system to change an account's bill cycle when a metered service agreement is activated, you need to turn on the account's **protect bill cycle** flag. You would do this if a customer liked to be billed at the start of the month regardless of when their device was read.

When the last service agreement for an account is stopped, the protect bill cycle flag is reset. This is to ensure that if the customer returns to start new service again, the bill cycle can be set based on How Does An Account Get Its Bill Cycle.

What Happens If An Account Has Service Agreements Spanning Metered Measurement Cycles?

A single account can have service agreements that are in several measurement cycles. The bill cycle on such an account will default based on the last activated metered service agreement.

It's important to be aware that an account will only have bills created when its bill cycle schedule so indicates. This means that the consumption for some service points could remain unbilled for a few weeks.

Designing Bill Periods

Bill periods are used by non-metered service agreements whose bill end dates need to fall on strict dates. You need only set up this information if you have this type of service agreement.

NOT_SUPPORTED

Refer to Ways To Control The End Date Of A Bill Segment for more information.

Copyright © 2026, Oracle and/or its affiliates.

Every bill period has a calendar that defines when bill segments are created for service agreements that use the bill period. Examples of bill periods include:

- Quarterly Bill - Last Day Of Quarter
- Quarterly Future Bill - Last Day Of Quarter
- Monthly - 15th Day Of Month.
- Monthly Future Bill - Last Day Of Month.

The Quarterly Bill - Last Day Of Quarter bill period would have a schedule that looked as follows:

Earliest Date On Which To Create A Bill Segment	Bill End Date
1-Oct-1998	30-Sep-1998
1-Jan-1999	31-Dec-1998
1-Apr-1999	31-Mar-1999
1-Jul-1999	30-Jun-1999
...	

The Quarterly Future Bill - Last Day Of Quarter bill period would have a schedule that looked as follows:

Earliest Date On Which To Create A Bill Segment	Bill End Date
15-Dec-1998	31-Mar-1999
15-Mar-1999	30-Jun-1999
...	

The remainder of this section provides examples using the above calendars.

The following example assumes the following:

- The service agreement starts on 18-Dec-1998.
- The service agreement's SA type references the Quarterly Future Bill - Last Day Of Quarter bill period.

The following table shows when bill segments will be produced (assuming the account's bill cycle attempt to create segments as soon as possible) for several bill periods:

Earliest Date Segment Will Be Produced	Bill Period
18-Dec-1998	18-Dec-1998 thru 31-Mar-1999
15-Mar-1999	1-Apr-1999 thru 30-Jun-1999
...	

The following example assumes the following:

- The service agreement starts on 18-Dec-1998.
- The service agreement's SA type references the Quarterly Bill - Last Day Of Quarter bill period.

The following table shows when bill segments will be produced (assuming the account's bill cycle attempts to create segments as soon as possible) for several bill periods:

Copyright © 2026, Oracle and/or its affiliates.

Earliest Date Segment Will Be Produced	Bill Period
1-Jan-1999	18-Dec-1998 thru 31-Dec-1998
1-Apr-1999	1-Jan-1999 thru 31-Mar-1999
...	

NOT_SUPPORTED

Refer to Setting Up Bill Periods for information about how to define this information.

Setting Up Bill Cycles

An account references a bill cycle. The bill cycle defines when the account is billed and when the account's debt is checked to determine if it's overdue. To define a bill cycle and its bill cycle schedule, open **Admin**, and then **Billing**, and then **Bill Cycle**, and then **Add.**

Description of Page

Enter a unique **Bill Cycle** and **Description** for every bill cycle.

If your implementation requires restricting bill cycles to accounts by CIS Division, specify a **CIS Division** on the bill cycle.

Use the **Bill Cycle Schedule** collection to define when bills are produced for the accounts in a given bill cycle. The following fields are required for each instance:

Window Start Date Specify the date on which the system should start trying to create bills for accounts in the cycle.

Window End Date Specify the last day on which the system will create bills for accounts in the cycle. This should be the last possible date on which a meter read could be loaded into Oracle Utilities Meter Data Management for the account.

Accounting Date Specify the financial date associated with the bills' financial transaction. The accounting date defines the financial period(s) to which the bills will be booked in your general ledger.

Estimate Date The date on which Oracle Utilities Meter Data Management may start estimating consumption if a real measurement cannot be found. When specified, this date is typically on or shortly before the window end date.

Freeze and Complete Turn on this switch if the system should freeze and complete any bill that is created without errors. If this switch is turned off, all bills created by the billing process will be left in the unfinished state. You would only turn this switch off if you want to verify an entire bill run prior to freezing it (e.g., if you are introducing a new version of a rate). If you turn this off, you will need to return to this page after verifying a bill run and turn it back on for the customers to receive bills. When the system next runs, it deletes all unfrozen bills and recreates them as per the instructions on the bill cycle schedule.

The batch bill creation process uses this schedule to define the bill cycles for which it should create bills.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Don't forget.** After you set up the bill cycles that correspond with measurement cycles, go to Setting Up Measurement Cycle Schedules and update the measurement cycles accordingly.

Setting Up Service Cycles And Measurement Cycle Schedules

Refer to Measurement Cycles, Measurement Cycle Processing, and Setting Up Measurement Cycle Schedules sections for more information

Setting Up Measurement Cycle Schedules

> **NOT_SUPPORTED**
>
> Refer to Designing Cycles for Metered Services for more information about measurement cycle. Refer to The Relationship Between Metered Service Cycles and Bill Cycles for more information about how measurement cycles are linked to bill cycles.

Description of Page

When you want to add a new measurement cycle schedule, you must specify the following information:

Service Cycle Enter the measurement cycle ID of the cycle whose routes will be downloaded.

Scheduled Selection Date Specify the date on which the system is meant to download information about the cycle's meters. This date should be a day or two before the scheduled work date.

Define the date the meters in the route are scheduled to be read using **Scheduled Work Date**. This date is extremely important as billing uses it when it looks for meter reads for service points in this cycle. If billing can't find a reading, consumption will be estimated as of this date (assuming the service agreement allows estimation).

Click **Pre-Generate Routes** if you need to finesse the cycle's routes on this Scheduled Selection Date (e.g., because some of the routes shouldn't be downloaded on a given date because the system is meant to estimate consumption). If you don't click this button, the system will create the routes on the scheduled selection date. It does this by creating a route extract for each downloadable route within the cycle (as defined by the route type). If you click this button, the system shows all routes within the cycle in the grid below. You must then define how the system is supposed to download each route on the scheduled selection date.

Click **Delete Routes** if you have pre-generated the routes and you was to remove them and allow the system to create the routes on the scheduled selection date.

Use the **Service Schedule Routes** to define how individual routes within the cycle should be handled during the download. The following fields are required for each schedule read date:

Copyright © 2026, Oracle and/or its affiliates.

Service Route The unique ID of the service route. The route's route type is displayed adjacent.

Schedule Type This defines if and how the system is supposed to download the route's meter reads. Valid values are: Cust Read, Download, Estimate. Only those routes defined as Download will have meter reads downloaded. The other values are used to document why the route won't be downloaded.

Schedule Status This defines the download status of the route. Valid values are Pending and Complete. This value is only displayed for routes with a schedule type of Download.

Where Used

This information is used by the meter read download process to determine which meter reads to download.

This information is used by the billing process to determine the date on which it expects to find a read. If it cannot find a read on or around this date, and it's OK to estimate consumption, consumption will be estimated as of the scheduled work date. The system uses the service agreement's rate schedule's frequency to determine the period of time around the scheduled work date in which it looks for a read. Refer to Defining Frequency Codes for more information.

Setting Up Bill Periods

Some SA types reference a bill period. The bill period defines when the service agreement's bill segments are produced and the respective end date of each bill segment.

> **NOT_SUPPORTED**
>
> Refer to Designing Bill Periods for more information.

To define a bill period and the bill period schedule, open **Admin**, and then **Billing**, and then **Bill Period**, and then **Add.**

Description of Page

Enter a unique **Bill Period** and a **Description** for every bill period.

Use the **Bill Period Schedules** collection when the system should create bill segments for service agreements that use a given bill period. It also defines the end date of each respective bill segment. The following fields are required:

Bill Date Specify the earliest date on which the system is allowed to create a bill segment for service agreements using this bill period.

Bill Seg End Date Specify the end date of the bill segment. For future bills, this will be after the bill date. For retro bills, this will be before the bill date.

This information is used by the bill segment creation process to determine the end date of service agreements that use a bill period.

Defining Statement Cycles

Copyright © 2026, Oracle and/or its affiliates.

If you have persons set up in the system to receive statements with financial information, you will need to assign them to a statement cycle and define a schedule for the statement cycle.

NOT_SUPPORTED

Refer to The Big Picture of Complex Statements for more information about statements.

- The Big Picture Of Statement Cycles
- Setting Up Statement Cycles

The Big Picture Of Statement Cycles

A statement cycle has a similar purpose to that of a bill cycle. It controls when statements will be produced.

- The Cyclical Statement Process
- Designing Your Statement Cycles
- Lifecycle of a Statement Cycle Schedule

The Cyclical Statement Process

Persons who wish to receive statements will work with you to determine how often these statements should be produced. Some persons may want a monthly statement, some a quarterly and some annually. For each unique schedule that is designed for your various statement persons, you will set up a Statement Cycle and its schedule.

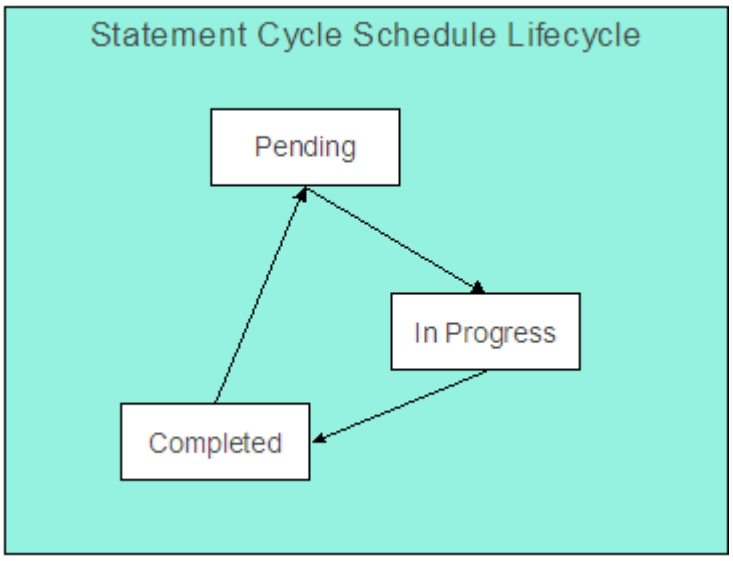
Designing Your Statement Cycles

The number of statement cycles is determined by a combination of the frequency that you will send statements to the statement persons and how many statement cycles you wish to manage within the same frequency.

So, for example, for all the statement persons who wish to receive a monthly statement, will you create only one monthly Statement Cycle so that all monthly statements are produced the same day? Or will you have several monthly statement cycles scheduled throughout the month? The answer will depend on the volume of statements being produced and on how you want to manage the statement production.

Lifecycle of a Statement Cycle Schedule

Copyright © 2026, Oracle and/or its affiliates.



Pending The statement cycle schedule is added in this state. The Create Statements background process find records in this state to process on the appropriate date.

NOT_SUPPORTED

Refer to Create Statements Background Process for more information.

In Progress Records in this state are currently being processed by the Create Statements background process.

Completed Records in this state have already been processed by the Create Statements background process. If a problem occurs with the Create Statements background process and it needs to be rerun, simply change the status back to pending and rerun the process.

Setting Up Statement Cycles

A Statement Construct references a statement cycle. The statement cycle defines when the statement person will receive statements with financial information related to the accounts and service agreements linked to the statement construct. To define a statement cycle and its statement cycle schedule, open **Admin**, and then **Billing**, and then **Statement Cycle**, and then **Add.**

Description of Page

Enter a unique **Statement Cycle** and **Description** for every statement cycle.

Use the **Statement Cycle Schedule** collection to define when statements are produced for the persons with statement construct records in the given statement cycle. The following fields are required for each instance:

Processing Date Specify the date on which the system should create statements for persons with statement construct records in the cycle.

Copyright © 2026, Oracle and/or its affiliates.

Status Indicates the status of the cycle schedule. Refer to Lifecycle of a Statement Cycle Schedule for more information.

The batch statement creation process uses this schedule to determine which statement cycles for it should create statements for the statement construct records. Refer to Create Statements Background Process for more information.

Defining Service Agreement Type (SA Types)

Every service agreement must reference a SA type. The SA type defines what you sell, how much you sell it for, to whom you sell it, how overdue debt is collected, and how sales will be booked in your general ledger.

> **Note**
>
> **Perfect foresight.** In a perfect world, the other control tables would have been set up with perfect foresight of setting up your SA types. In reality, setting up your SA types may invalidate some of your earlier decisions. Don't feel bad if this happens, some amount of iteration is natural.

- Designing SA Types For Service Agreements With Service Points
- Designing SA Types For SAs Without Service Points
- Designing SA Type For Other Segmentations
- Designing Prepayment Billing Options
- SA Types And The Financial Design
- Setting Up SA Types

Designing SA Types For Service Agreements With Service Points

Designing your SA types is an iterative process. To minimize some iterations, we recommend using the steps outlined in this section to complete the following table. When the table is complete, you're ready to set up your SA types.

Division/SA Type	Service Type

The topics in this section provide guidelines describing how to fill in this table for SA types associated with service agreements that charge for service point-oriented services.

- CIS Division Segmentation
- Service Segmentation
- Receivable Segmentation

Copyright © 2026, Oracle and/or its affiliates.

- Revenue Segmentation
- Rate Segmentation
- Service Point (SP) Type Segmentation
- Company Usage Segmentation
- Debt Class Segmentation
- Budget Billing Segmentation

CIS Division Segmentation

A CIS division is typically associated with a jurisdiction. The definition of a jurisdiction is a geographic-oriented entity with unique business rules. For example, if you conduct business in California and Nevada, and each state has different collection rules, you will need a separate jurisdiction for each state. You must set up a CIS division for each jurisdiction in which you conduct business.

If we assume that you are located in a single jurisdiction - say California - we will need a single CIS division for all of our SA types.

CIS Division/SA Type
CA

Service Segmentation

At a minimum, you will have one SA type for every different type of utility service offered by your organization. If we assume you sell electricity, gas, water, wastewater and cable; your SA Types will be as follows:

CIS Division/SA Type	Service Type
CA/G	Gas service
CA/W	Water service
CA/E	Electric service
CA/WW	Waste water service
CA/CABLE	Cable

Receivable Segmentation

Many organizations segregate their receivable balances in the general ledger. For example, the receivable amount associated with gas and water service may be maintained in separate GL accounts.

If your organization does this, you will probably have at least one SA type for each such receivable account because each SA type references a distribution code that typically contains the receivable account.

Copyright © 2026, Oracle and/or its affiliates.

- The word *probably* is underlined because this is a rule of thumb. There are situations where the number of receivable accounts isn't directly related to the number of SA types. This happens when an organization maintains very detailed receivable accounts in the general ledger and maintaining a one-to-one relationship between SA types and distribution codes would lead to a massive proliferation of SA types (and you don't want this!). If your organization maintains very detailed receivable accounts, please speak to your implementers, they should be able to introduce a small customization to generate the appropriate receivable account rather than extract it from the distribution code.
- The word *typically* is underlined because there are several SA types that don't book to a receivable account when bill segments are generated. For example, company usage and charitable contributions. Refer to Company Usage Segmentation and Charitable Contribution Segmentation for examples of SA types that don't book to receivable accounts.

We'll simplify our example and assume your organization has one receivable account for all types of utility service. Given this, we won't need additional SA types to support receivable segmentation:

CIS Division/ SA Type	Service Type	Distribution Code
CA/G	Gas service	A/R-UTIL
CA/W	Water service	A/R-UTIL
CA/E	Electric service	A/R-UTIL
CA/WW	Waste water service	A/R-UTIL
CA/CABLE	Cable service	A/R-UTIL

Revenue Segmentation

Look at your rates and determine which rates can be used by each SA type. The following table shows the sample rates that can be used for each service:

CIS Division/ SA Type	Service Type	Distribution Code	Rates
CA/G	Gas service	A/R-UTIL	GALL-1
CA/W	Water service	A/R-UTIL	WALL-1
CA/E	Electric service	A/R-UTIL	ERES-1, ERES-2, ECOM-1, EIND-1, ELAMP-1
CA/WW	Waste water service	A/R-UTIL	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	CABLE

Now, look at the rates' calculation rule GL Distribution window. You're looking for calculation rules whose GL distribution is affected by revenue class. If there are no revenue classes referenced on the calculation rules, this means that the revenue associated with the rate will be booked to a single GL account regardless of the type of customer. If you see revenue classes, this means that the revenue account associated with the calculation rule(s) differs depending on the SA type's revenue class. If revenue classes are used in the rates, you must create a different SA Type for every revenue class.

Copyright © 2026, Oracle and/or its affiliates.

Let's assume the following:

- The gas rate (GALL-1) references the RESIDENTIAL, COMMERCIAL and INDUSTRIAL revenue classes in order to differentiate revenue based on the type of customer.
- None of the other rates differentiate revenue based on customer class.

Our SA types will now look as follows:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1
CA/W	Water service	A/R-UTIL	N/A	WALL-1
CA/E	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2, ECOM-1, EIND-1, ELAMP-1
CA/WW	Waste water service	A/R-UTIL	N/A	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE

Notice that we created new SA types for gas in order to specify the respective revenue class. We didn't do this for the other services because it isn't necessary. However, you should feel free to do this if it feels right or if you need it for reporting purposes. For example, if you want to report on all *residential* service agreements, but you differentiate only gas by residential versus commercial, you'll be in trouble.

Rate Segmentation

Every SA Type whose service agreements have their charges calculated with a rate must have one or more rates linked to it. These define which rates can be linked to the SA type's service agreements. If different service agreements have different valid rate combinations, you will need to split the SA types accordingly.

For example, look at the electric rates. If we assume:

- the RES rates can only be used for residential customer,
- the IND rates can only be used for industrial customers,
- the COM rates can only be used for commercial customers,
- the LAMP rates can only be used for lamp customers

Then you might want to set up a new SA type to indicate such. The advantage of doing this is that you get more control over which rates can be used on a given SA type. The disadvantage is that you proliferate SA types. Unfortunately, it's really a question of taste.

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1

Copyright © 2026, Oracle and/or its affiliates.

CA/W	Water service	A/R-UTIL	N/A	WALL-1
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2
CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1
CA/LAMP	Electric service	A/R-UTIL	N/A	ELAMP-1
CA/WW	Waste water service	A/R-UTIL	N/A	WWALL-1
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE

Warning

Don't be too specific in your analysis in respect of rate segmentation because you could end up with a separate SA type for every rate (and you don't want this). We understand this is a very subjective warning, but we recommend that you start out with broad bands of rates that can be used on a SA type and narrow it down if you end up unhappy with the results. For example, you don't have to set up a separate SA type for low-income residential gas customers just because they have a special rate. Rather, you can leave your SA types as they are and treat low-income gas customers as a subset of your residential gas customers. **For more information**, refer to Setting Up Start Options.

Note

Rate override. If a service agreement of this type may be linked to a terms of service record, you must also consider whether or not the rate schedule could be overridden by a template SA linked to the terms of service record. Refer to SA Type - Rate for information about the possible values for this field.

Service Point (SP) Type Segmentation

Every SA Type whose service agreements exist to bill for service point-oriented service has one or more SP types. These define which type of service points can be linked to the SA type's service agreements. If different service agreements have different valid SP types, you will need to split the SA types accordingly.

For each service point-oriented SA type, determine if there are any restrictions in respect of the types of service points that can use the SA type's rates. For example, if we assume that only commercial SP types can be used by commercial customers, industrial SP types by industrial customers, residential SP types by residential customers, your SA types will be:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates	SP Type
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1	G-RES
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1	G-COM

Copyright © 2026, Oracle and/or its affiliates.

CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1	G-IND
CA/W-RES	Water service	A/R-UTIL	N/A	WALL-1	W-RES
CA/W-COM	Water service	A/R-UTIL	N/A	WALL-1	W-COM
CA/W-IND	Water service	A/R-UTIL	N/A	WALL-1	W-IND
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES
CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1	E-COM
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1	E-IND
CA/WW-RES	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES
CA/WW-COM	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM
CA/WW-IND	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE	CABLE

Notice the wastewater SA types reference both water and waste water service points. This is intentional as wastewater service uses the consumption from the water service to calculate some part of the wastewater charge.

Company Usage Segmentation

Up to now, we've discussed SA types associated with service agreements linked to your customers. The system has also been designed to keep track of the expenses associated with your company's use of power. If you want the system to do this, you must create at least one SA type for each service consumed by your organization.

For example, if we assume your organization consumes electric, gas, and water service; your SA types will now be as follows:

CIS Division/ SA Type	Service Type	Distribution Code	Revenue Class	Rates	SP Type	Bill Seg Type
CA/G-RES	Gas service	A/R-UTIL	R - residential	GALL-1	G-RES	BD-RATED
CA/G-COM	Gas service	A/R-UTIL	C - commercial	GALL-1	G-COM	BD-RATED
CA/G-IND	Gas service	A/R-UTIL	I - industrial	GALL-1	G-IND	BD-RATED
CA/W-RES	Water service	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED
CA/W-COM	Water service	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED
CA/W-IND	Water service	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED
CA/E-RES	Electric service	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED

Copyright © 2026, Oracle and/or its affiliates.

CA/E-COM	Electric service	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED
CA/E-IND	Electric service	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED
CA/WW-RES	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED
CA/WW-COM	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED
CA/WW-IND	Wastewater service	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED
CA/CABLE	Cable service	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED
CA/E-COY	Electric service	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG
CA/G-COY	Gas service	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG
CA/W-COY	Water service	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG

Notice the three company usage SA types do not reference an A/R account as their distribution code. This is because when bill segments are created for these types of service agreements, the system must debit an expense account (or contra-revenue account) rather than a receivable account.

Also notice we introduced a new column - Bill Segment Type. Notice that the customer-oriented SA types use the BD-RATED bill segment type and the company usage SA types use the COMPUSAG bill segment type. Different bill segment types are necessary because company usage SA types use a different algorithm to calculate their bill segment's financial transaction algorithm because they don't affect either payoff or current balance.

NOT_SUPPORTED

For more information, refer to Designing and Defining Bill Segment Types.

Debt Class Segmentation

Every SA Type has a debt class. The debt class is used to categorize a service agreement's debt for the purpose of credit and collections (C&C) analysis. If a given SA Type has different categories of debt from C&C's perspective, you will have to split the SA Type.

NOT_SUPPORTED

For more information about debt class, refer to Designing Your Collection Procedures.

If we assume that your residential services are regulated and your commercial and industrial services are deregulated, we won't have to introduce additional SA types.

Copyright © 2026, Oracle and/or its affiliates.

CIS Division/ SA Type	Distribution Code	Revenue Class	Rates	SP Type	Bill Seg Type	Debt Class
CA/G-RES	A/R-UTIL	R - residential	GALL-1	G-RES	BD-RATED	REGU
CA/G-COM	A/R-UTIL	C - commercial	GALL-1	G-COM	BD-RATED	UNRE
CA/G-IND	A/R-UTIL	I - industrial	GALL-1	G-IND	BD-RATED	UNRE
CA/W-RES	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED	REGU
CA/W-COM	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED	UNRE
CA/W-IND	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED	UNRE
CA/E-RES	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED	REGU
CA/E-COM	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED	UNRE
CA/E-IND	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED	UNRE
CA/WW-RES	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED	REGU
CA/WW-COM	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED	UNRE
CA/WW-IND	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED	UNRE
CA/CABLE	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED	UNRE
CA/E-COY	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG	No debt
CA/G-COY	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG	No debt
CA/W-COY	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG	No debt

Notice the three company usage SA types do not have a debt class. This is because their bill segment type's FT algorithm doesn't cause debt to be created and therefore there is no reason to have a debt class. However, you'll need to create a "dummy" debt class - call it N/A - for these SA types because every SA type must reference a debt class.

Budget Billing Segmentation

Many utilities offer their customers levelized payment plans to smooth out the seasonal bill variations. We call this levelized amount the *budget amount*.

NOT_SUPPORTED

Refer to Budget Billing for more information about budgets in general. Refer to Billing - Current Balance versus Payoff Balance for an example of budget billing accounting.

If we assume that you only allow budget billing on the electric and gas residential services, then you'll need to update your CA/G-RES and CA/E-RES SA types:

Copyright © 2026, Oracle and/or its affiliates.

CIS Division/ SA Type	Dist Code	Rev Class	Rates	SP Type	Bill Seg Type	Debt Class	Recurring Charge Control Info
CA/G-RES	A/R-UTIL	R	GALL-1	G-RES	BD-RATED	REGU	Amount to bill is Not Allowed Amount is Optional Frequency is Monthly Recurring Amount Label is Budget Amount:
CA/G-COM	A/R-UTIL	C	GALL-1	G-COM	BD-RATED	UNRE	
CA/G-IND	A/R-UTIL	I	GALL-1	G-IND	BD-RATED	UNRE	
CA/W-RES	A/R-UTIL	N/A	WALL-1	W-RES	BD-RATED	REGU	
CA/W-COM	A/R-UTIL	N/A	WALL-1	W-COM	BD-RATED	UNRE	
CA/W-IND	A/R-UTIL	N/A	WALL-1	W-IND	BD-RATED	UNRE	
CA/E-RES	A/R-UTIL	N/A	ERES-1, ERES-2	E-RES	BD-RATED	REGU	Amount to bill is Not Allowed Amount is Optional Frequency is Monthly Recurring Amount Label is Budget Amount:
CA/E-COM	A/R-UTIL	N/A	ECOM-1	E-COM	BD-RATED	UNRE	
CA/E-IND	A/R-UTIL	N/A	EIND-1	E-IND	BD-RATED	UNRE	
CA/WW-RES	A/R-UTIL	N/A	WWALL-1	W-RES, WW-RES	BD-RATED	REGU	
CA/WW-COM	A/R-UTIL	N/A	WWALL-1	W-COM, WW-COM	BD-RATED	UNRE	
CA/WW-IND	A/R-UTIL	N/A	WWALL-1	W-IND, WW-IND	BD-RATED	UNRE	
CA/CABLE	A/R-UTIL	N/A	CABLE	CABLE	BD-RATED	UNRE	
CA/E-COY	EXP-COMP	N/A	E CO USE	E-CO USE	COMPUSAG	No debt	
CA/G-COY	EXP-COMP	N/A	G CO USE	G-CO USE	COMPUSAG	No debt	
CA/W-COY	EXP-COMP	N/A	W CO USE	W-CO USE	COMPUSAG	No debt	

Notice the following:

Copyright © 2026, Oracle and/or its affiliates.

- We updated the two SA types to allow recurring charge information.
- The Recurring Charge Amount is Optional. Why? Because those customers who aren't on a budget plan won't have a recurring charge amount. Those customers on a budget will have a recurring charge amount.

> **Note**
>
> **Turn on Eligible for Budget.** Besides indicating that these SA types use recurring charges, you also need to turn on the Eligible for Budget switch on the SA type to indicate that this type of SA participates in budget processing.
>
> **Override Budget Eligibility**. You may plug in an override budget eligibility algorithm on an SA type that is configured to be Eligible for Budget if certain service agreements of this type are not eligible.

> **NOT_SUPPORTED**
>
> Refer to the Description of Page under SA Type - Billing for the definition of the recurring charge attributes.

Designing SA Types For SAs Without Service Points

The topics in this section provide guidelines describing how to design the SA types associated with your service agreements that don't have service points.

- Overpayment Segmentation
- Write Off Segmentation
- Connection Charge Segmentation
- Charitable Contribution Segmentation
- Payment Arrangement Segmentation
- Merchandise Segmentation - Installment Billing
- Deposit Segmentation - Installment Billing
- Billable Charge Segmentation
- Over/Under Cash Drawer Segmentation
- Payment Upload Error Segmentation
- CIAC Segmentation
- Loan Segmentation
- Non-Billed Budget Segmentation

Overpayment Segmentation

Copyright © 2026, Oracle and/or its affiliates.

When a customer pays more than they owe, you must decide what to do with the excess money. The following points describe two possibilities:

- You could create a new service agreement to hold the excess (let's call it an overpayment SA). The credit would be transferred from this service agreement to the billable service agreements when the next bill is completed. This means that all billable service agreements have the same opportunity to receive the overpayment when they are billed in the future.
- You could amalgamate the excess payment on one of the existing, billable service agreements. For example, if a customer has both electric and gas service, the excess funds could be kept on either the gas or the electric SA. This would result in the following:
  - The service agreements that do NOT receive the overpayment will have debt when they are next billed.
  - The service agreement that receives the overpayment could have its future debt offset by the overpayment (meaning that it could have a credit balance until the service agreement's future bill segments offset the overpayment amount).
- The above situation is not desirable unless the customer intentionally overpaid one service agreement. The first method (keeping the overpayment on a separate service agreement) obviates this potential problem. Obviously, if your organization sells a single service (and therefore your customers have a single service agreement) you would choose the second method.

You control which method is used by plugging in the appropriate Overpayment Distribution algorithm on each Customer Class (i.e., you can choose a different method for different customer classes). If you choose to hold overpayments on a separate SA, then you must set up an SA Type as described in the following table:

CIS Division/ SA Type	Service Type	Distrib. Code	Eligible for Billing	Debt Class	Pay Seg Type	Do Not Overpay	One-time
CA/ OVERPAY	Other	A/P - OVER	Not billed	N/A	Normal	No	Yes

Notice the following about the new overpayment SA type:

- It has an interesting distribution code. This is because when a payment segment is created for this type of service agreements, the system must credit a liability (an overpayment is a liability).
- It's important to indicate that the overpayment SA is a one-time service agreement. Why? Because this means that the system will automatically close the SA when it's balance falls to zero (i.e., when all of the overpayment has been used to satisfy future bills).
- A bill segment type is not needed because the system never creates bill segments for such service agreements (they exist only to hold excess credits).
- You may also want to turn on the alert message
- You must plug-in a bill completion algorithm on this SA type. This bill completion algorithm will transfer the credit balance to the account's other service agreements when the bill is completed. Refer to The Credit Transfer Algorithm for more information about this algorithm.
- You must also reference this overpayment SA type as the parameter value on your overpayment algorithm (this algorithm is plugged in on the desired customer classes). Refer to Overpayment Algorithm for more information about this algorithm.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **If overpayment means charitable contribution.** Some organizations sponsor a program that works as follows - if a customer overpays a bill by a given amount (say $5), this amount is assumed to be a charitable contribution. If you have this requirement, you will create another SA type to hold a customer's charitable contributions. This SA type will look similar to the one described below (see Charitable Contribution Segmentation) except it is not billable. The funds will be credited to this service agreement by creating a new overpayment algorithm that is similar to the base package Overpayment Algorithm. This new algorithm will be very similar to the existing algorithm. The main difference will be that it will have to check if the overpayment amount is an exact value (say $5). If so, it will create a payment segment for the charitable contribution SA type; otherwise it will create a payment segment for the overpayment SA.

Write Off Segmentation

When you write off non-collectable debt, you transfer the receivable from a "normal" service agreement onto one or more write-off service agreements. When the debt is transferred to a write-off service agreement, the distribution code on the "normal" service agreement is credited (typically an A/R GL account), and the distribution code on the write-off service agreement is debited.

You will almost always need a write-off service agreement whose distribution code is the write-off expense. However, you probably don't book all of the write-off amount to a write-off expense account. Why? Because the debt that you're writing off typically contains both revenue and liabilities. At write-off time, you want to book the written off revenue to a write-off expense account and you want to reduce the liabilities (you don't owe the liability if you don't get paid). This means you'll need another SA type for the liabilities. Refer to The Ramifications of Write Offs in the General Ledger for a complete explanation.

The following table contains the minimum number of SA Types that you'll need to hold your write-offs.

CIS Division/ SA Type	Service Type	Distrib. Code	Bill Seg Type	Debt Class	Pay Seg Type	Do Not Overpay
CA/WO-STD	Other	EXP-W/O	Not billed	WO	Normal	Yes
CA/WO-LIA	Other	LIA-General	Not billed	WO	Normal	Yes

Notice the following about the new write-off SA types:

- They have interesting distribution codes. This is because when debt is transferred to these types of service agreements, the system must debit either an expense account (i.e., write-off expense) or a liability account. It's important to note that in The Ramifications of Write Offs in the General Ledger we explain how this liability account may be overwritten with the liability account that was originally booked.
- Neither needs a bill segment type because the system never creates bill segments for such service agreements (they exist only to hold uncollectable debt)
- Even though the debt is not collectable, it still has a debt class. Why? Because the system shows a customer's debt on many inquiries by debt class and it's important to show write-off debt on these queries.

Copyright © 2026, Oracle and/or its affiliates.

- The combination of Payment Segment Type and Do Not Overpay are important. Refer to The Ramifications of Write Offs in the General Ledger for a complete explanation.

> **Note**
>
> The adjustment type used to set the offending service agreement's current balance equal to its payoff balance is defined on each write-offable SA type. The adjustment type used to transfer the delinquent debt to the write-off service agreement is defined on the write-off SA type.
>
> **An Alternative.** If you have a limited number of liability accounts, you may decide to have a separate write-off service agreement for each liability account. Doing this would proliferate the number of service agreements created at write-off time. However, it would simplify the remittance of payment to the taxing authority if the reversed liability is ever paid.

Connection Charge Segmentation

If you levy connection charges, you have two options:

- You can create a SA type that exists purely to handle connect charge debt. After doing this, you'd create a start option for this SA type that causes an adjustment to be levied as part of the start service process. This adjustment would contain your standard connection charge. This approach would be used by a utility that had multiple services (e.g., a combined electric, gas, water utility) that only levies a single connection charge regardless of the number of services started. If you use this approach, make sure to indicate the SA type is non billable.
- You can levy a start adjustment on one of your existing SA types (e.g., CA/E-RES). The easiest way to do this is with a start option. On the start option you'd indicate an adjustment to be levied as part of the start service process. This adjustment would contain your standard connection charge. This approach would be used by a utility that had a single service offering (e.g., an electric-only utility). Refer to Setting Up Start Options for more information.

In the table below, we show what would be necessary if you want to have a separate service agreement for the connection charge.

CIS Division/ SA Type	Service Type	Distrib. Code	Bill Seg Type	Debt Class
CA/CONNECT	Other	A/R-UTIL	Not billed	REGU

Notice the following about the new connection charge SA type:

- It has a normal receivable distribution code.
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (its charged via an adjustment).
- The debt class is interesting - REGU (for regulated). We are intentionally linking the connection charge debt to the same debt class as the regulated debt from which it originates. This way, the C&C process will consider the connection charge debt the same as regulated debt and therefore perform the regulated collection (which results in the severance of all regulated service agreements).

Copyright © 2026, Oracle and/or its affiliates.

Charitable Contribution Segmentation

If your organization accepts charitable contributions made by your customers, you must create a SA type to hold these contributions.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/CHARITY	Other	A/P-CHAR	CHAR	RECUR	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Contrib. Amount

Notice the following about the new charitable contribution SA type:

- It has an interesting distribution code. This is because when a payment is distributed to these types of service agreements, the system must credit a payable account (i.e., charitable contribution payable) rather than a receivable account. Note well, we have assumed a receivable is not incurred when the bill segment for the charitable contribution is created.
- It uses an interesting bill segment type - RECUR. This bill segment type was set up to create recurring charges that don't automatically stop at some point in time.
- The debt class is interesting - CHAR (for charity). This is done so that past due charitable contribution debt is treated separately from other types of debt.
- The recurring charge control information is set up as defined.

NOT_SUPPORTED

Refer to the Description of Page under SA Type - Billing **for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges **for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

Warning

It's important that you assign the charitable contribution SA type with a payment segment type that only affect current balance (as opposed to affecting current AND payoff balance). This is because there is no receivable recognized when the contribution is billed and therefore there is no payoff balance to relieve when it's paid.

Copyright © 2026, Oracle and/or its affiliates.

Payment Arrangement Segmentation

If your organization allows customers to payoff outstanding debt using payment arrangements (e.g., current bill plus $X), you will need a new SA type for every debt class that can have a payment arrangement. If we assume you can have payment arrangements for both regulated and unregulated debt, then you'll need at least two more SA types (you may have more SA types if you need to segregate the payment arrangement receivable amount by utility type (or some other type)).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/PA-REGU	Other	A/R-ARRG	REGU	RECUR-AS	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Arrange Amount
CA/PA-UNRE	Other	A/R-ARRG	UNRE	RECUR-AS	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Arrange Amount

Notice the following about the new payment arrangement SA types:

- They have an interesting distribution code. This is because when funds are transferred to these types of service agreements, the system must debit a receivable (i.e., payment arrangement receivable).
- They use an interesting bill segment type - RECUR-AS. This bill segment type was set up to create recurring charges that stop when the customer no longer has a payoff balance.
- Each new SA type references the debt class whose debt it will pay off. We are intentionally linking the payment arrangement debt to the same debt class as the regulated debt from which it originates. This way, the C&C process will consider the arrangement debt as the same as regulated debt and therefore perform the regulated collection (which results in the severance of all regulated service agreements).
- The recurring charge control information is set up as defined.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to the Description of Page under SA Type - Billing **for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges **for how** you can use Start Options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

Merchandise Segmentation - Installment Billing

If your organization allows customers to purchase merchandise using an installment plan, you must create a SA type for this.

Note

No installments. If the customer must pay for the merchandise in one lump amount, you'd create an SA type similar to the Connection Charge Segmentation example.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/MERCH-I	Merch	A/R-MRCH	UNRE	RECUR-AS	Amount to bill is Not Allowed Amount is Required Frequency is Monthly Recurring Amount Label is Install Amount

Notice the following about the new merchandise SA type:

- It has a normal receivable distribution code.
- It uses an interesting bill segment type - RECUR-AS. This bill segment type was set up to create recurring charges that stop when the customer no longer has a payoff balance.
- The recurring charge control information is set up as defined.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to the Description of Page under SA Type - Billing **for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges **for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started. Refer to Start Option Considerations For SA Types That Use Initial Adjustments **for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

Deposit Segmentation - Installment Billing

If your organization allows customers to pay deposits using an installment plan, you must create an SA type for this.

Note

No installments. If the customer must pay for the deposit in one lump amount, you'd create an SA type similar to the Connection Charge Segmentation example. Just make sure the adjustment that's levied to charge for the deposit amount doesn't affect payoff balance (when you bill a deposit, the customer doesn't really owe anything because it's not a true receivable from an accountant's perspective).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type	Recurring Charge Control Info
CA/DEP-I	Other	A/P-DEPO	DEP	RECURATB	Amount to bill is Required Amount is Required Frequency is Monthly Recurring Amount Label is Install Amount

Notice the following about the new deposit SA type:

- It has an interesting distribution code. This is because when a payment is distributed to these types of service agreements, the system must credit a payable account (i.e., deposit payable) rather than a receivable account. Note well, we have assumed a receivable is not incurred when the bill segment for the deposit is created.
- It uses an interesting bill segment type - RECURATB. This bill segment type was set up to create recurring charges that stop when the system has billed the Total Amount to Bill.
- The debt class is interesting - DEP (for deposit). This is done so that past due deposit "debt" is treated separately from other types of debt.

Copyright © 2026, Oracle and/or its affiliates.

- The recurring charge control information is set up as defined. Note well, the Amount to bill is Required.

> **NOT_SUPPORTED**
>
> Refer to the Description of Page under SA Type - Billing **for the definition** of the recurring charge attributes. Refer to Start Option Considerations For SA Types That Use Recurring Charges **for how** you can use start options to automatically populate a service agreement's recurring charge fields with appropriate values when service is started.

> **Note**
>
> **Bill messages on receipt of deposit in full.** The base package includes a special FT Freeze algorithm that can be specified on deposit SA Types. It recognizes when a deposit has been paid in full, and creates a bill messages to inform the customer. Refer to algorithm DEP PIF MSG in Algorithm Types for more information.

Billable Charge Segmentation

You create a billable charge whenever a customer should be charged for a service that occurs outside the normal course of business. For example, you would create a billable charge to charge a contractor for the repair of a ruptured gas line. You can also use billable charges to "pass through" other bill ready charges generated outside the system, by another application, or by a 3rd party supplier.

A billable charge must reference a service agreement. This service agreement behaves just like any other service agreement:

- **Bill segments are created for the service agreement.** Whenever billing is performed for an account with billable charge service agreements, the system creates a bill segment for each service agreement with unbilled charges. If multiple unbilled charges exist for a given service agreement, only one bill segment will be created and it will contain details about all of the billable charges.
- **Payments are distributed to the service agreement.** Payments made by an account are distributed to its billable charge service agreements just like any other service agreement.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge service agreements for overdue debt and responds accordingly when overdue debt is detected.

Therefore, you must set up at least one SA type to hold your billable charge debt. You may have multiple charges based on billing frequencies, A/R booking, debt monitoring, etc. It's really up to you.

The easiest way to determine how many billable charge SA types you'll need is to define every conceivable billable charge (which you should have done when you designed your billable charge templates). Then ask yourself if they have the same billing and payment behavior, if so, you'll have one SA type. If not, you'll need one SA type for each combination.

Copyright © 2026, Oracle and/or its affiliates.

We will assume your billable charges are all used to levy unusual one-off charges that can be collected in the same way, therefore we'll need one SA type.

CIS Division/ SA Type	Service Type	Distrib Code	Debt Class	Bill Seg Type	Billable Charge Templates	Rate
CA/ ONETIME	Other	A/R-UTIL	UNRE	BILLCHRG	TREETRIM DAMAGE	None
CA/ PASSTHRU	Electricity	A/R-OTHER	UNRE	BILLCHRG	None	None
CA/ADDON	Electricity	A/R-OTHER	EXTERNAL	BILLCHRG	None	TAXES

Notice the following about the new one time SA type:

- It has a normal receivable distribution code.
- Its debt class is unregulated.
- It uses an interesting bill segment type - BILLCHRG. This bill segment type was set up to create bill segments using billable charges.
- It references the valid billable charge templates that can be used on this SA type.

> **Note**
>
> **One Time Charge.** The ONE TIME example shown above implies this SA type exists to hold one-time charges. Because of this, you should turn on the One Time Charge switch on the SA type so that service agreement's of this type are automatically closed when final payment is received. You don't have to do this because a customer could have a single billable charge service agreement that is perpetually active for pass through charges (i.e., it doesn't have a stop date). If you do this, the system will create a bill segment for this service agreement whenever it finds an unbilled billable charge linked to the service agreement.

Notice the following about the pass through SA Type:

- It doesn't use the normal distribution code or debt class. This is done so that the debt and receivable can be tracked separately. If these charges were being pass through from another system, you might want to track these financial values separately.
- It still uses the normal bill segment type - BILLCHRG. From a billing perspective, there is no difference between this and the one time SA Type.
- Templates are not relevant - these charges on not created on-line using templates, but are loaded via the Billable Charge Upload Staging.

Notice the following about the add on charges SA Type:

- This is an example of bill-ready charges (similar to pass through) to which the system adds on other charges, for example, taxes.
- It still uses the normal bill segment type - BILLCHRG. From a billing perspective, there is no difference between this and the one time SA Type.
- It also uses a Rate. In this case, the bill creation algorithm (specified on the bill segment type) will take any billable charge lines and attach them to a bill. In addition, these billable charges will include billable charge service quantities (SQs). These service quantities will

Copyright © 2026, Oracle and/or its affiliates.

also be swept onto the bill segment, and the Rate (TAXES in this example) will be applied. In order for taxes to be calculated, the billable charge SQs must include the total taxable amount - the system is not able to apply the rate on top of the other billable charges. But, it can apply the tax rate to the SQs that are supplied.

- You can also use this technique to bill other rate-ready service quantities, like kWh, CCF, *etc.* This is a way to process rate-ready data for which you have a contract, but you do not know the meter (and therefore, cannot collect real meter reads).
- If the rate has pre-processing calculation groups, these will be applied as well.

NOT_SUPPORTED

For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Over/Under Cash Drawer Segmentation

In order to balance a tender control that is out-of-balance, your organization must set up an account with a service agreement whose SA type references the over/under expense account. You will probably only have one service agreement that references this SA type, but you still must have it if you remit funds via a cash drawer.

NOT_SUPPORTED

For more information about over/under processing, refer to How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under).

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type
CA/OVR UNDR	Other	EXP-OV/UND	N/A	Not billed

Notice the following about the new SA type:

- It has an interesting distribution code. This is because when a payment segment is applied to this type of service agreement, the system must debit an expense account for under amounts (and credit it for over amounts).
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (it only has over/under payment segments linked to it).
- It uses the N/A debt class because the credit and collections process should never consider debt associated with service agreements of this type (because it's not really debt).

Payment Upload Error Segmentation

If the payment upload process detects an invalid account on a payment upload record, it will create a payment for the suspense service agreement defined on the upload process' tender

Copyright © 2026, Oracle and/or its affiliates.

source (see Setting Up Tender Sources). You should create a special SA type for this service agreement.

> **NOT_SUPPORTED**
>
> For more information about the payment upload process, refer to Phase 3 - Create Payment Events, Tenders, Payments, and Payment Segments.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	Bill Seg Type
CA/SUSPENSE	Other	EXP-MISC	N/A	Not billed

Notice the following about the new SA type:

- It has an interesting distribution code. This code should probably be a suspense account. All payment segments that are created for this service agreement will eventually be transferred to a "real" service agreement and therefore this GL account's balance should be zero when no payments are in suspense.
- It doesn't need a bill segment type because the system never creates bill segments for such service agreements (it only has invalid account payment segments linked to it).
- It uses the N/A debt class because the credit and collections process should never consider debt associated with service agreements of this type (because it's not really debt).

CIAC Segmentation

If your company bills and refunds Contribution In Aid of Construction (CIAC) contracts, you must create one or more SA types. CIAC contracts are typically used to levy charges associated with line extensions. These types of service agreements are different from other service agreements because the initial amount charged is refunded to the original payee when new properties (or extensions) are added to the extension (or when the new properties are subsequently billed for service).

> **Note**
>
> **Billable charges are used for the original CIAC service agreement.** CIAC SA types are always Billable Charge SA types because an operator must specify the exact amount to charge the contractor using a billable charge. Refer to Billable Charge Segmentation for more information.

The following points describe how CIAC processing is implemented in the system:

- When a new line extension is build, you will create a new service agreement that references your CIAC SA type. This service agreement should be linked to a service point associated with the extension. The easiest way to create this service agreement and link it to the service point is by using the Start Account transaction. It's important to remember to define your CIAC SA type as a valid SA type for the SP type used to represent the line extension.

Copyright © 2026, Oracle and/or its affiliates.

- When a new premise / service point is built that "hangs" off the original extension, the new service point should be linked to the original service point. This can be done by referencing the original service point as the "parent" service point of the new service point using a foreign key value characteristic.
- The system periodically monitors the original CIAC service agreement. The process that performs this monitoring is referenced on the original CIAC service agreement's SA type. The exact processing that takes place during this monitoring is up to your organization's specific business requirement. For example, if you refund 10% of every bill produced for the "downstream" service points to the original contractor, you would have logic in your CIAC monitoring process that looks for recent bill segments produced for the "downstream" service agreements and then creates an adjustment for the master service agreement.

> **Note**
>
> **CIAC refunds are idiosyncratic.** Because CIAC refund processing is idiosyncratic, we do not supply any CIAC monitoring processes in the base package. This is because the likelihood that they could be used is extremely low because of your organization's unique requirements.

Therefore, if your organization performs CIAC processing, you should create a special SA type.

CIS Division/ SA Type	Service Type	Distrib. Code	Debt Class	CIAC Process	Bill Seg Type
CA/CIAC	Other	A/R - CIAC	UNRE	Refund based on percent of future bills	Billable Charge

Loan Segmentation

If you loan money to customers that is recouped using an amortization schedule, you need to set up an SA type for the loan service agreement.

CIS Division/SA Type	Service Type	Distrib. Code	Non Billed Budget Monitoring	Debt Class	Renewal	Bill Seg Type
CA/LOAN	Other	A/R - LOAN	A/R - STLN	UNRE	LOAN	

> **NOT_SUPPORTED**
>
> Refer to Setting Up The System To Enable Loans for more information.

Copyright © 2026, Oracle and/or its affiliates.

Non-Billed Budget Segmentation

If you allow your customers to pay set amounts at specified intervals (e.g. every two weeks), you need to set up SA types for Non-Billed Budget service agreements. Non-billed budgets are typically used when your company bills on an infrequent basis and you want to provide your customers with a mechanism to make smaller payments more frequently. You may implement two types of Non-Billed Budgets monitored and unmonitored, each type requiring a different SA type. You may also implement different renewal options for Non-Billed Budgets.

CIS Division/SA Type	Service Type	Distrib. Code	Non Billed Budget Monitoring	Debt Class	Renewal	Bill Seg Type
CA/NBB-MON	Other	A/P - OVPY	Monitored	NBB	Optional	Not billed
CA/NBB-UNMON	Other	A/P - OVPY	Unmonitored	N/A	Optional	Not billed

NOT_SUPPORTED

For more information about monitored and unmonitored Non-Billed Budgets, refer to Defining Non-Billed Budget Options.

Designing SA Type For Other Segmentations

The earlier parts of this discussion described the most common factors that cause the creation of SA Types. However, many obscure factors could cause the introduction of more SA Types. In this section, we explain these more obscure factors.

Warning

We strongly recommend not being too pedantic when considering the factors described in this section. If you can only think of a few strange situations that would necessitate a SA type, think carefully before you introduce it. It's better to be a little less than perfect than end up with large number of obscure SA types.

- Cash Distribution Segmentation
- Adjustment Profile Segmentation
- Late Payment Charge Segmentation
- Debt Classification Segmentation
- Allow Estimates Segmentation
- Severance Criteria Segmentation
- Deposit Class Segmentation

Copyright © 2026, Oracle and/or its affiliates.

- Sub SA Types
- Financial Settlement SA Types
- Usage Request SA Types
- Initial Consumption Period Considerations
- Processing Sequence Considerations

Cash Distribution Segmentation

Every SA Type has a payment segment type. The payment segment type defines the cash account to which the SA type's payments should be booked. If different service agreements have different cash accounts, you will need to split the SA types accordingly.

Adjustment Profile Segmentation

Every SA Type has one or more adjustment profiles. These profiles define the valid adjustment types that can be booked to the SA type's service agreements. If different service agreements within an SA type have different mixtures of valid adjustment types, you must split the SA types accordingly.

Late Payment Charge Segmentation

An option exists on SA Type that causes the system to generate a late payment charge if payment is not received on time. If you don't levy late payment charges on all service agreements, you will need to determine when you do and design your SA types accordingly.

In addition, if you levy late payment charges, the percentage levied and the algorithm that defines the amount of the outstanding balance subject to the charge is defined on the SA type.

Debt Classification Segmentation

Every SA Type has a debt class. The debt class is used to categorize a service agreement's debt for the purpose of credit and collections (C&C) analysis. If a given SA Type has different categories of debt from C&C's perspective, you must split the SA Type.

NOT_SUPPORTED

For more information about debt class, refer to Designing Your Collection Procedures.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Write Off Debt Class vs. Normal Debt Class.** An SA type references both write off debt class and normal debt class. An SA type's write-off debt class controls the write-off rules imposed on service agreements of a given type. An SA type's normal debt class controls the collection rules imposed on service agreements of a given type. Refer to Different Collection Criteria For Different Customers And Different Debt for more information about collection rules. Refer to Different Write-Off Criteria For Different Customers And Different Debt for more information about write-off rules.

Allow Estimates Segmentation

Every SA Type has a switch that controls whether the system should have Oracle Utilities Meter Data Management estimate consumption if meter reads are missing at billing time. If a given SA Type has different situations when the system should and should not estimate, you will have to split the SA Type.

> **Note**
>
> **Override.** You can override the value of the SA Type's estimation switch on an individual service agreement. This means that if only a few service agreements don't abide by the SA Type's estimation switch, you can change the switch value of these service agreements.

Severance Criteria Segmentation

Every SA Type has severance criteria. The severance criteria define the severance process used to sever service if the customer doesn't pay. You can have multiple severance processes if different conditions warrant a different process. For example, you may have a different severance process if the customer has life support.

If you have a SA Type that requires different severance conditions other than those currently supported, you can make a programmatic change to introduce the additional conditions OR you can split the SA Type.

> **NOT_SUPPORTED**
>
> For more information about severance, refer to Designing Your Severance Procedures.

Deposit Class Segmentation

Copyright © 2026, Oracle and/or its affiliates.

Every SA Type that exists to hold a cash deposit will reference a deposit class. The deposit class defines the business rules that control various functions including interest calculation and refund criteria. You will need multiple deposit SA Types if any of the deposit class' rules / conditions differ for different types of deposits. For example, if residential customers use a different recommended deposit algorithm as compared to commercial customers, you'd need one SA type for residential deposits and another for commercial deposits (where the residential deposit SA type will reference the residential deposit class and the commercial deposit SA type will reference the commercial deposit class).

You will need additional deposit SA types if your customers can have multiple deposits where each deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one SA type for regulated deposits and another for unregulated deposits.

NOT_SUPPORTED

For more information about deposit class, refer to Designing and Defining Deposit Classes.

Sub SA Types

If you operate in a deregulated environment AND if you provide billing services for other service providers, you will need to create SA types to handle the billing of the service providers' charges.

NOT_SUPPORTED

Refer to Designing Your SA Types And Start Options For Sub SAs for more information.

Financial Settlement SA Types

If you operate in a deregulated environment, you may have to create financial settlement service agreements for the service providers. As explained in Service Providers Have Service Agreements Too, We Bill For Them and They Bill For Us service providers require a service agreement to hold adjustments used to increase how much you owe the service provider (or how much they owe you).

NOT_SUPPORTED

Refer to Designing SA Types For Service Provider Financial Settlements for more information.

Copyright © 2026, Oracle and/or its affiliates.

Usage Request SA Types

Oracle Utilities Meter Data Management stores meter reading information and is responsible for calculating bill determinants during billing. The SA types for these customers will require special setup.

NOT_SUPPORTED

For more information about usage requests, refer to The Big Picture Of Usage Requests.

Initial Consumption Period Considerations

Bill segments produced for a service agreement have two time periods:

- The bill segment period. The bill segment period defines the entire period of time covered by a bill segment's charges.
- The consumption period. The consumption period defines the period of time used to calculate the number of days for daily charges.

The consumption period almost always starts one day after the bill segment period. The consumption period always ends on the bill segment's end date. For example, a bill segment period that spans 5-Jan-2002 through 6-Feb-2002 will almost always have a corresponding consumption period of 6-Jan-2002 through 6-Feb-2002. The reason that the start dates don't match is because a bill segment's start date equals the end date of the prior bill segment (i.e., the start date was already counted in the previous bill segment's consumption period and we don't want to count it twice).

The only time when the previous paragraph isn't true is the first bill segment that's produced for a new service agreement. This is because different utilities count the first day of a new service agreement differently than others. Because of this, a flag exists on SA Type called **Initial Start Date Option**. This flag controls whether the service agreement's start date is included in the consumption period in a service agreement's first bill segment.

The following table describes the ramifications of the options you can set for this flag.

Flag Value	Consumption Period Calculation	Use This Option When
Add 1 Day Always	The consumption period's start date is calculated by adding 1 day to the service agreement's start date. (The SA start date is never included in the consumption period for the first bill segment.)	You want the initial bill and all subsequent bills to have a consistently calculated consumption period (i.e., the consumption period for the first and all other bills is always one day less than the bill segment's period).

Copyright © 2026, Oracle and/or its affiliates.

Add 1 Day for Back-to-back	A back-to-back occurs when any service point for this service agreement was previously linked to a different service agreement that was stopped on the same date that the new service was started (i.e., there is no gap in the service). If a back-to-back is detected, the consumption period start date is calculated by adding 1 day to the SA start date. If no back-to-back is detected, the start date of the consumption period is the SA start date.	You want to flexibly handle consumption period calculation. If you start customers on the same date as the stop date of the previous customer, billing does not include the SA start date in the consumption period. However, if you start a new customer one day (or more) after the stop date of the previous customer, billing includes the first day of the service agreement in the consumption period.
Include First Day	The start date of the consumption period is the service agreement start date. (The SA start date is always included in the consumption period.)	You want to always include the first day AND you will never encounter a back-to-back situation.

The example below shows how the consumption period would be calculated with the various options for a customer who starts service on January 1.

	First Bill **Meter Read: Jan 31** **Bill Segment Period: Jan 1 to Jan 31**	Second Bill **Meter Read: Feb 28** **Bill Segment Period: Jan 31 to Feb 28**	Third Bill **Meter Read: Mar 31** **Bill Segment Period: Feb 28 to Mar 31**
Consumption period using Add 1 Day Always	Jan 2 to Jan 31 (**30 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Add 1 Day for Back-to-back when back-to-back is detected	Jan 2 to Jan 31 (**30 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Add 1 Day for Back-to-back when back-to-back is NOT detected	Jan 1 to Jan 31 (**31 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)
Consumption period using Include First Day	Jan 1 to Jan 31 (**31 days**)	Feb 1 to Feb 28 (28 days)	Mar 1 to March 31 (31 days)

There may be SA types for which the value of this flag does not affect the consumption period calculation and still other SA types where this flag is never used. For example,

- For billable charge service agreements, the consumption period is equal to the start and end dates on the billable charge and therefore this flag is not applicable.
- A sub SA either inherits the consumption period from the master SA or it uses billable charges. As a result, billing does not use this flag.

Copyright © 2026, Oracle and/or its affiliates.

- For some service agreements, the charges on the rate are not affected by the consumption period. For example, if you have a customer with a simple meter and a simple usage-based charge, billing calculates the amount of consumption between the start reading and end reading and applies the rate (i.e., the number of days in the consumption period doesn't impact the charges in some rates).
- Some service agreements are not billed, for example, overpayment service agreements. For this type of service agreement and other types of service agreements that are not billed, this flag is not applicable. However, the system does not prevent a value from being entered in these cases to allow for an implementation process to use the flag if needed.

Setting this flag to an appropriate value is significant for certain types of service agreements.

- For services whose rate includes daily charges, the configuration of this flag may impact the first bill segment for the service agreement. For example, if the first bill period is October 1st through October 31st, do you consider consumption period to be 31 days or 30 days? How you want to bill the customer on the first bill determines how you set this flag.
- For service agreements that request usage (or bill determinants) based on interval data, the interval usage recordings typically begin on the first day of service. If you add 1 to the start date, you miss billing for intervals on the first day. In this case, you should set the value to Add 1 Day for Back-to-back.

Processing Sequence Considerations

You may have customers with a complex collection of contracts such that the calculation for one bill segment relies on information calculated by another bill segment for the same account. For example, perhaps you need to process your bill segments as follows:

- Step 1: Calculate bill segments for all the account's "single site" service agreements (i.e., each service agreement related to a single premise)
- Step 2: Calculate the bill segment for additional charges for the "head office" service agreement where the charges are based on the aggregated consumption of all premises from all bill segments (calculated in the first step).

To create the "head office" bill segment for the account, you must control the order in which the system creates the bill segments for each service agreement.

The SA type allows you to indicate a billing processing sequence that controls the order in which the service agreements are processed. The processing sequence is optional and service agreements are processed in ascending order of their SA type's billing processing sequence. As a result, SA Types with a processing sequence of zero are processed first, then 1, then 2, etc.

> **Note**
>
> If you have sub SAs linked to a master SA, the billing processing sequence is used first to order the creation of bill segments for the master SAs. If you populate a processing sequence on an SA type for a sub SA, it is used to control the order in which the sub SAs for a given master SA are processed relative to each other. Refer to Designing Your SA Types And Start Options For Sub SAs for more information.

Copyright © 2026, Oracle and/or its affiliates.

The billing processing sequence also controls the order of service agreements in the following other processes:

- Execution of pre bill completion algorithms. The system processes each service agreement in the billing processing sequence order. Within each service agreement, the pre bill completion algorithms on its SA type are processed in the order of the algorithm's sequence.
- Execution of bill completion algorithms. The system processes each service agreement in the billing processing sequence order. Within each service agreement, the completion algorithms on its SA type are processed in the order of the algorithm's sequence.

Designing Prepayment Billing Options

This section provides guidelines describing how to design service agreements that handle smart meter prepaid billing.

When a customer is subject to prepayment, normal cycle-based billing is not used to calculate the bill segments for their service. Rather, a prepay billing (PPB) process creates bill segments frequently (for example, daily) and periodically requests funds to maintain an adequate payoff balance. Bills will only be used to periodically inform the customer of the various financial transactions that have occurred since the last bill was produced.

Implementing prepayment billing requires the configuration of the following objects:

- SA Type
- Adjustment Types
- Bill Segment Type
- Service Task Type

These configuration tasks are described below.

> **Note**
>
> For additional information, see the detailed descriptions in the application for the C1-PrepayBillerTaskType and C1-PrepayBillerTask business objects.

SA Type for Prepay Billing

To configure a prepay billing SA Type, create a new SA type with the following attributes:

- Set the Special Role flag to **Bill Determinants Required.** The Prepay Option should be set to **Prepaid Smart Meter.**
- Specify the SA Creation system event with the algorithm CI_CREATEPPB to create the service agreement's prepay biller.
- Specify the Payment Freeze system event with the algorithm STPZ-RMVC to set current amount to zero on a payment.
- If required, specify the Bill Completion system event with an algorithm of the type C1-CREPPBBM to add a bill message that informs an end customer of their prepaid charges for the period. You may need to create an algorithm of this type if it does not exist in your system.
- **Eligible for Budget** should be set to false.

Copyright © 2026, Oracle and/or its affiliates.

- Reference a new **Prepay Bill Segment Type**, which is described below.

The following adjustment types must also be defined on the prepay SA type:

- **Adjustment Type (Xfer)** specifies the type of adjustment used to transfer funds from the delinquent service agreements to the prepaid service agreement
- **Adjustment Type (Current=0)** specifies the type of adjustment used to set the prepaid current balance to zero after debt has been transferred
- **NSF Adjustment Type** specifies the type of adjustment used to levy an NSF charge. The NSF Adjustment Type is described in greater detail below.

The new SA types must be added to the data value mapping between CCB and MDM. This is to enable the synchronization of SAs of this type to MDM. The new CCB SA types can be mapped to existing US types in MDM. For additional information about this configuration, please see the integration documentation.

Adjustment Types for Prepay Billing

- **Funds Request Adjustment Type:** This adjustment type captures the type of adjustment used to request more prepaid funds. This adjustment type is specified on the prepaid biller task type. Specify the following algorithms on this adjustment type:
  - **Adj. Financial Algorithm** should reference the Adjustment FT creation algorithm ADJT-CA (Payoff=0/Current=Adj Amount (No GL)).
  - **Adjustment Freeze Algorithm** should reference an algorithm of the type C1-PPBADJFRZ (Set Prepaid Funds Request FT Details). You may need to create an algorithm of this type if it does not exist in your system.
- **NSF Adjustment Type:** For prepaid customers on autopay, when a user cancels an autopay tender the cancel reason must indicate that an NSF charge should be levied. The system invokes the NSF Charge algorithm specified on the tender's account's customer class which creates the NSF adjustment on the appropriate SA. The charge is levied using the NSF Adjustment Type specified on the SA's SA type. This adjustment type must reference the new adjustment freeze algorithm that will disable automatic payments on the PPB. This algorithm must be based on the algorithm type C1-DISPPBAP (Disable PPB's Automatic Payment). You may need to create an algorithm of this type if it does not exist in your system.

Bill Segment Type Required for Prepay Billing

Configure a new bill segment type for prepaid service agreements. This bill segment type should have the following attributes:

- **Create Algorithm** should be set to C1-BSBS-UR to create bill segment using a usage request.
- **Financial Algorithm** should be set to a new algorithm with algorithm type C1-NEM-GL (Payoff Amt = Bill Amt / Current Amt = 0 (GL affected)).You may need to create an algorithm of this type if it does not exist in your system.
- **Get Consumption Algorithm** should be set to C1-BSGC-USG to get bill segment consumption using a usage request.

Service Task Type for Prepay Billing

Configure a new service task type for prepay billing. This task type should reference the Prepay Biller (PPB)Task Type Business Object (C1-PrepayBillerTaskType). The task type must be specified on the new prepay billing SA creation algorithm. The task type captures the adjustment type used when an adjustment is created to request additional funds from the

Copyright © 2026, Oracle and/or its affiliates.

customer. Ensure that this adjustment type references the adjustment freeze algorithm (C1-PPBADJFRZ) that populates the financial transaction's arrears date.

SA Types And The Financial Design

In this section, we provide an example of how our SA Types map to Bill Segment Types, Payment Segment Types, and Adjustment Profiles. This example is meant to help solidify the power of the financial model, it is not necessarily indicative of how your specific implementation will look.

> **Warning**
>
> If you are not comfortable with the topics described in Defining Financial Transaction Options, the following table will not make sense.

Division/SA Type	Distribution Code	Bill Segment Type	Payment Segment Type	Adjustment Profiles
CA/G-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES, BUDGET
CA/G-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/G-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/W-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES, BUDGET
CA/E-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-RES	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-COM	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/WW-IND	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/CABLE	A/R-UTIL	BD-RATED	NORMAL	BALXFER, MISCFEES
CA/E-COY	E-COMP	COMPUSAG		
CA/G-COY	E-COMP	COMPUSAG		

Copyright © 2026, Oracle and/or its affiliates.

CA/W-COY	E-COMP	COMPUSAG		
CA/WO-STD			NORMAL	BALXFER
CA/WO-LIA			NORMAL	BALXFER
CA/CONNECT			NORMAL	BALXFER, MISCFEES
CA/CHARITY	A/P-CHAR	RECUR	CHARITY	CHARITY
CA/PA-REGU	A/R-ARRG	RECUR-AS	NORMAL	BALXFER, MISCFEES, DPA
CA/PA-UNRE	A/R-ARRG	RECUR-AS	NORMAL	BALXFER, MISCFEES, DPA
CA/MERCH-I	A/R-MRCH	RECUR-AS	NORMAL	BALXFER, MISCFEES, MERCH
CA/DEP-I	A/P-DEPO	RECUR-AS	NORMAL	BALXFER, MISCFEES, DEPOSIT
CA/ONETIME	A/R-UTIL	BILLCHRG	NORMAL	BALXFER, MISCFEES
CA/OVR UNDR	EXP-OV/UND		NORMAL	
CA/OVERPAY	A/P-OVER		NORMAL	BALXFER
CA/SUSPENSE	A/R-SUSP		NORMAL	
CA/NBB	A/P-OVPY		NORMAL	BALXFER, MISCFEES, NBB

If you operate in a deregulated environment, you will also have additional SA types as described under Designing Your SA Types And Start Options For Sub SAs and Designing SA Types For Service Provider Financial Settlements you may have additional SA types.

Setting Up SA Types

In the previous section, Designing SA Types, we presented a case study that illustrated a mythical organization's SA types. In this section, we explain how to use the windows on the SA Type window group to maintain your SA Types.

- SA Type - Main Information
- SA Type - Detail
- SA Type - Billing
- SA Type - Rate
- SA Type - SP Type
- SA Type - Adjustment Profiles
- SA Type - C&C
- SA Type - Billable Charge Template
- SA Type - Characteristics
- SA Type - Algorithms

Copyright © 2026, Oracle and/or its affiliates.

- SA Type - Billable Charge Overrides
- SA Type - Contract Option Type
- SA Type - NBB Recommendation Rule
- Setting Up Start Options
- Start Option Merge

SA Type - Main Information

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Add** to define core information about your SA Types.

> **Warning**
>
> Every SA Type is owned by a CIS Division. This Division controls many values that can be referenced on the SA Type. If you don't understand Divisions and their place in the application, do NOT attempt to set up your SA Types. Rather, refer to Setting Up CIS Divisions before proceeding.

Description of Page

Enter a unique combination of **CIS Division** and **SA Type** for every service agreement type.

Enter a **Description** for the SA type.

Service Type defines the type of service associated with the SA type. If the SA type has rates, only rates belonging to this service type may be linked to the SA type.

Select the **Distribution Code** and **GL Division** that defines the receivable account for receivable-oriented service agreements. For non-receivable oriented service agreements, this distribution code is typically as follows:

- Charitable contributions. The distribution code is a charity payable account.
- Deposits. The distribution code is a deposit payable account.
- Non-Billed Budgets. The distribution code is an overpayment payable account.
- Company usage. The distribution code is a company usage expense account.
- Write off. The distribution code is a write-off expense account.
- Payment arrangements. The distribution code is a payment arrangement receivable account.

> **NOT_SUPPORTED**
>
> For more information about GL accounts, refer to The Source Of GL Accounts On Financial Transactions.

Select the **Revenue Class** associated with the SA Type (and its service agreements). The revenue class may affect the revenue account(s) generated by the service agreement's rate.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Designing Calculation Groups and Rules for more information about revenue class.

Turn on **Start Options Required** if you want to force a customer service rep to choose a start option when they start service for this SA Type (on the Start Account window). If this switch is off and a rate is required for the SA Type, the system defaults the SA type's default rate on new service agreements. The default rate is defined on the SA Type - Rate page.

NOT_SUPPORTED

Refer to Setting Up Start Options for more information about the pros and cons of requiring start options.

Select the **Pay Segment Type** that defines how payment segments linked to service agreements of this type affect:

- The service agreement's payoff and current balances

NOT_SUPPORTED

For more information about payment segment types, refer to Setting Up Payment Segment Types.

When a tender is canceled, a cancellation reason must be supplied. If the cancellation reason indicates a NSF (non sufficient funds) charge should be levied, the system invokes the Levy an NSF Charge algorithm specified on the tender's account's customer class. Because adjustments must be linked to a service agreement, the algorithm must determine the appropriate service agreement to use to levy the adjustment based on business rules. The charge is levied using the **NSF Adjustment Type** of the appropriate service agreement's SA type.

Warning

You must specify adjustment type profiles on the SA type (on the Adjustment Type window) before adjustment types will appear in the above drop downs.

NOT_SUPPORTED

For more information about adjustment types, refer to Setting Up Adjustment Types. For more information about cancellation reasons, refer to Setting Up Payment Cancellation Reasons.

Copyright © 2026, Oracle and/or its affiliates.

Select the **Payment Priority**. This field is available for use by the algorithms that distribute partial payments amongst an account's service agreements. Higher priority service agreements will have their debt relieved before lower priorities. Refer to Distribution Based on Payment Priority and Delinquent Payment Distribution Algorithm for information about payment distribution algorithms that use this field.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is PAY_PRIORITY_FLG.

> **NOT_SUPPORTED**
>
> For more information about distribution priority, refer to Distributing A Payment Amongst An Account's Service Agreements.

Select the **Delinquent Payment Priority**. This field is available for use by the algorithms that distribute partial payments amongst an account's service agreements. Higher priority service agreements will have their debt relieved before lower priorities. Refer to Delinquent Payment Distribution Algorithm for information about a payment distribution algorithm that uses this field.

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is DEL_PRIORITY_FLG.

Turn on **Do Not Overpay** if the system is not allowed to distribute an overpayment to this type of service agreement (i.e., the service agreement is not allowed to have a system-created credit balance). This field is available for use by algorithms that distribute overpayments. Refer to Overpayments Held On Highest Priority Service Agreement for information about an overpayment algorithm that uses this field.

Turn on **Late Payment Charge** if the system should generate a late payment charge for this type of service agreement if payment is not received on time. If this is turned on, you must define the **LPC Calc. Algorithm** used to calculate the late payment charge amount. Refer to Defining Late Payment Charge Options for more information about late payment charges. Examples of algorithm types used for calculating late payment charges are BILPC-SPRC and BILPC-TOTAL.

Define the **Adj**ustment **Type (Synch Curr)** that will be used to synchronize (make equal) the current amount with the payoff amount on a service agreement of this type. This type of processing happens as follows:

- Most write-off algorithms that perform financial efforts (e.g., writing off debt), will issue an adjustment of this type if the service agreement's current and payoff balances are not equal.
- If a user stops a customer on a budget plan, the system issues adjustments of this type to synchronize the customer's current and payoff balances.

Copyright © 2026, Oracle and/or its affiliates.

- If a user stops a service agreement covered by a **Non-Billed Budget** , the system issues adjustments of this type to synchronize the customer's current and payoff balances.
- If a cancellation of a bill segment occurs after a customer has stopped participating in a budget plan, an adjustment of this type is issued to synchronize the imbalance created when the bill segment's financial transaction is canceled.

Turn on **CIAC SA Type** and specify an appropriate **CIAC Refund Process** if service agreements of this type are used to bill for Contribution In Aid of Construction (CIAC) charges. Refer to CIAC Segmentation for more information.

SA Type - Detail

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Detail** page to define additional details about a given SA type.

Description of Page

Turn on **Display As Alert** if Control Central should display an alert if an account has a service agreement of this type that isn't Closed or Canceled. If this switch is on, also enter the **Alert Information** to appear on Control Central. We recommend only using this feature on unusual SA types (e.g., payment arrangements, write-offs) so that a CSR is not presented with an alert for every SA type.

If this SA Type is used for any of the **Special Roles**, defined in the dropdown, indicate which one. Valid values are: Billable Charge, Cash Deposit, Loan, Non-Billed Budget, Payment Arrangement, Write Off, Bill Determinants Required. This information is used on windows with functionality that can only be used by service agreements used for specific roles. For example, the Billable Charge window group can only reference Billable Charge service agreements.

If Special Role is Cash Deposit, you must define the **Deposit Class** of the deposit. You should also define a **Deposit Class** on every SA type to which a given deposit can be distributed.

> **NOT_SUPPORTED**
>
> Refer to What Do Deposit Classes Do? for more information.

If the Special Role is Loan, you must also define the following fields:

- Use the **Interest Bill Factor** to define the bill factor code for the loan interest rate.
- Use **Override Interest** to indicate whether the interest rate defined on the interest bill factor may be overridden at the SA level. If you select Allowed, the interest rate may be overridden by a contract value on a start option or the SA.
- Use the **Loan A/R Distribution Code** to define the distribution code to be used when posting the short-term receivable amount to the general ledger (the normal distribution code is used for the long-term receivable). If the normal distribution code is the same as the **Loan A/R Distribution Code**, the SA type does not differentiate between long- and short-term receivables. If the two distribution codes are different, the SA type differentiates between long- and short-term receivables.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Defining Loan Options for more information about **Interest Bill Factor**, **Override Interest** and **Loan A/R Distribution Code**.

If the Special Role is Non-Billed Budget , you must also define the following:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to the transfer accumulated credit from the Non-Billed Budget SA to the SAs covered by the Non-Billed Budget when the account is billed or the Non-Billed Budget SA is stopped.
- Use the **Non-Billed Budget Monitoring** to specify whether the Non-Billed Budget is monitored by the account debt monitor.

If the SA type is defined as Eligible for Non-billed Budget, you must also define the following:

- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the service agreement's current balance to zero when a service agreement of this type is linked to an active, monitored Non-Billed Budget.

If the Special Role is Payment Arrangement and you have not enabled Use Alternate PA Creation Logic, you must also define two adjustment types:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the delinquent service agreements to the payment arrangement service agreement.
- Set **Use Alternate PA Creation Logic** if your payment arrangement SA Types share a debt class with other SA Types. When this is set, the current balance on the transfer adjustment on the payment arrangement SA is set to zero instead of using the below adjustment type to set the current balance to zero. When payment arrangements are in their own debt class, the additional credit adjustment offsets the transfer adjustment. This results in an offsetting credit and debit in the payment arrangement debt class and a single credit on the non-payment arrangement debt class. However, when a single debt class is used, the two credits inaccurately impact the way debt age (arrears) is calculated and presented. Using the alternate PA creation logic ensures that debt age is calculated and displayed accurately when a single debt class is used.
- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the payment arrangement's current balance to zero after funds have been transferred.

If the Special Role is Write Off, you must also define the following adjustment types:

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the uncollectable service agreements to the write off service agreement.

Note

It may be important to your organization to track the original source of debt that is being written off. The system is delivered with three characteristic types: C1-WODST, C1-WOCHT, and C1-WOCHV that can record the distribution code, characteristic type and characteristic value from the FT GL(s) being written off onto the write off adjustment. To enable this logic, specify these characteristic types as allowable characteristic types on your write off transfer adjustment type.

If the Prepay Option is **Prepaid Smart Meter**, you must also define two adjustment types:

Copyright © 2026, Oracle and/or its affiliates.

- Use **Adjustment Type (Xfer)** to specify the type of adjustment used to transfer funds from the delinquent service agreements to the prepaid service agreement.
- Use **Adjustment Type (Current=0)** to specify the type of adjustment used to set the prepaid current balance to zero after debt has been transferred.

> **Warning**
>
> You must specify adjustment type profiles on the SA type (on the Adjustment Type window) before adjustment types will appear in the above drop downs.

The **Stop Option** can be used to automatically stop a service agreement when all other service agreements of an account are stopped. Valid values are: Automatically Stop SA. When all service point related service agreements of an account are stopped then all additional service agreements where the **Stop Option** is set to Automatically Stop SA will also be stopped. This is useful, for example, with charitable contribution service agreements. The charity service agreement should be stopped when all utility related services are stopped.

> **NOT_SUPPORTED**
>
> Refer to The Lifecycle of a Service Agreement for more information on the **Stop Option.**

Turn on **One Time Charge** if this SA type is used for one-time invoices. When a one-time invoice service agreement is created, the system sets the stop date of the SA to be equal to the start date.

Turn on **Sub SA** if this SA type is used to define the business rules for sub service agreements.

> **NOT_SUPPORTED**
>
> Refer to Sub Service Agreements for more information about sub service agreements.

Renewal of SAs of this type may be Optional, Not Allowed or Required depending on your business processes. If renewal is not allowed, the SA expires on the expiration date. Renewal treatment is an important consideration for SAs that require an expiration date, such as non-billed budget SAs.

If renewal is required or optional, specify the **Days Before Expiration for Renewal**. Note that currently this is only used by Non-Billed Budgets to calculate the renewal date based on the expiration date.

If the Special Role is Non-billed Budget, **Non-Billed Budget Monitoring** must indicate whether the Non-Billed Budget is monitored by the account debt monitor.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Credit and Collections and Non-Billed Budgets for more information about monitoring Non-Billed Budgets.

Set **Eligible for Payment Arrangement** to Eligible if this SA type can be included in a payment arrangement.

Note

This is automatically set to Ineligible on SA types with a Special Role of Payment Arrangement.

Where Used

The alert information is used by Control Central to alert a CSR when unusual service agreements exist for an account. Refer to Control Central - Search for more information.

Only SA types designated as being Billable Charge may have billable charges linked to them. Refer to Maintaining Billable Charges for more information.

Only SA types designated as being Cash Deposit are processed by the various deposit-related background processes (e.g., interest calculation, automatic refund, etc.). Refer to The Big Picture Of Deposits for more information.

Only SA types designated as Loan are used to define the loan terms for a loan SA. Refer to **Loans** for more information.

Only SA types designated as Non-billed Budget may be used to set up Non-Billed Budgets. Refer to **Non-Billed Budgets** for more information.

Only SA types designated as being Payment Arrangement may be used on the payment arrangement window group. Refer to Setting Up Payment Arrangements for more information.

Only SA types designated as being Write Off may be specified as the write off SA type on distribution codes. Refer to Setting Up Distribution Codes for more information.

Only service agreements whose SA type is designated as being Write Off appear on the Write Off SAs query. Refer to Write Off - Write Off SAs for more information.

SA Type - Billing

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Billing** page to define how the system manages bill segments for service agreements of a given SA type.

Description of Page

Turn on **Eligible for Billing** if the system should create bill segments for service agreements of this type. This will typically be turned on for all service agreements except for those used to hold write-off amounts or to levy one-off adjustments.

Copyright © 2026, Oracle and/or its affiliates.

Define the minimum number of days a bill segment (other than the final segment) must span using **Minimum Days for Billing**. This is useful to prevent initial bill segments that span only a few days.

> **NOT_SUPPORTED**
>
> For more information about minimum days, refer to Preventing Short Bill Segments.

Select the **Bill Segment Type** that controls both how bill segments for this SA Type will be created and how the related financial transaction affects the general ledger and the customer's debt.

> **NOT_SUPPORTED**
>
> For more information about bill segment types, refer to Setting Up Bill Segment Types.

Use **Default Description on Bill** to define the verbiage that should print on the customer's bill.

> **Note**
>
> **Rates overwrite this description.** The Default Description on Bill is not applicable for service agreements whose charges are calculated using a rate. Why? Because the description that appears on the bill segment is defined on the rate schedule's rate version calculation group.
>
> **Billable charges overwrite this description.** The Default Description on Bill is not applicable for service agreements whose charges are calculated using a billable charge. Why? Because the description that appears on the bill segment is defined on the billable charge.

Use the **Billing Processing Sequence** if you need to control the order in which service agreements linked to this SA type are processed by billing.

> **NOT_SUPPORTED**
>
> Refer to Processing Sequence Considerations for more information.

Use **Bill Print Priority** to define the order in which the SA type's bill segments should appear on bills (relative to the other SA types that appear on a bill).

> **Note**
>
> The values for this field are customizable using the Lookup table. This field name is BILL_PRT_PRIO_FLG.

Copyright © 2026, Oracle and/or its affiliates.

Use **Max Bill Threshold** if you want the system to generate a bill error when a bill segment is produced in batch that exceeds a given value. These bill errors will appear on the standard billing queries and To Do lists. If, after reviewing the high value bill segment, an operator truly intends to send the bill out, they should regenerate the bill. Refer to How To Correct A Bill Segment That's In Error for more information.

> **Warning**
>
> The value entered in this field will DEFAULT onto service agreements of this type when they are first created. An operator may change the default value on a service agreement in case a specific customer has unusually high bills that continually error out. It's important to be aware that if you change the value of High Bill Amount on an SA type and there already exist service agreements of this type, the existing service agreements will contain the original value (the new value on the SA type will not be propagated on the existing service agreements).

Use **Graph Unit Of Measure** to define the unit of measure of the graphed consumption on the bill (if any).

Turn on **Allow Estimates** if the system should indicate to Oracle Utilities Meter Data Management whether to generate estimated consumption if meter reading(s) cannot be found at billing time. This value is defaulted onto service agreements and can be overridden on an individual service agreement.

Turn on **Characteristic Premise Required** if a characteristic premise must be linked to the service agreement when the service agreement is activated. The characteristic premise is used to define the taxing authorities associated with the service agreement's bill segments. It is also used to identify where the service agreement's service is located on various windows.

> **NOT_SUPPORTED**
>
> For more information about how characteristic premise is used, refer to **An Illustration Of A Bill Factor And Its Characteristics**.

Use the **Initial Start Date Option** to control how billing should calculate the consumption period for the very first bill for service agreements of this type. This field is not applicable for sub SA types or SA types with a special role of Billable Charge. Valid values are Include First Day, Add 1 Day Always and Add 1 Day for Back-to-back. Refer to Initial Consumption Period Considerations for more information.

Non-metered service agreements may have the end date of their bill segments defined on a user-maintained bill period schedule. This option is used when bill segments must fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter). If this SA type should be billed like this, select Use Bill Period in the **Use Calendar Billing** field. When this option is used, you must define the **Bill Period** whose schedule defines the bill segment end dates.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about bill period schedules, refer to Designing Bill Periods. For more information about other bill end date methods, refer to Ways To Control The End Date Of A Bill Segment.

Instead of the Use Bill Period method, non-metered service agreements may have their bill segment end date based on the first day of service. For example, if service started on the 16th of some month, the ongoing bill segments will start on roughly the 16th of each month. This option is frequently used to bill for garbage or cable service. If this SA type should be billed like this, select Anniversary Future Billing or Anniversary Past Billing in the **Use Calendar Billing** field. When either option is used, you must define the **Anniversary Bill Frequency**. This frequency defines the amount of time between bill segments.

NOT_SUPPORTED

For more information about anniversary billing, refer to Using The Anniversary Method. For more information about other bill end date methods, refer to Ways To Control The End Date Of A Bill Segment.

Total Bill Amount indicates whether service agreements of this type can use the total amount to bill field on the service agreement page. Valid values are Not Allowed and Required. Only SA types used to bill for deposits or loans should have this field set to Required.

If Required is selected, you must enter the **Total Amount To Bill Label**. The **Total Amount To Bill Label** defines the label that prefixes the total bill amount on the service agreement page for service agreements of this SA type.

NOT_SUPPORTED

For more information about total amount to bill and deposit service agreements, refer to Total Amount To Bill. For more information about total amount to bill and loan service agreements, refer to **Setting Up The System To Enable Loans** .

Recurring Charge indicates whether service agreements of this type can use the recurring charge field on the service agreement window. Valid values are Not Allowed, Optional and Required. If either Optional or Required are used, you must enter:

- **Recurring Chg Amt Label**. This defines the label that prefixes the recurring charge amount on the service agreement window for service agreements of this SA type.
- **Recurring Charge Frequency**. This defines the following:
  - Specifies the frequency at which the Recurring Charge Amount specified on service agreements of the SA Type is to be billed.
  - Serves as the basis for proration of the Recurring Charge Amount.
  - Specifies the frequency at which service agreements of the SA Type without a rate and/or meters will be billed.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information about how to use the recurring charge information, refer to Charitable Contribution Segmentation, Merchandise Segmentation - Installment Billing, Deposit Segmentation - Installment Billing, Payment Arrangement Segmentation, Budget Billing Segmentation, The Terms Of A Loan Are Stored On A Service Agreement.

Turn on **Eligible for Budget** if service agreements of this type can participate in budget billing. If this switch is turned on, then you must define the **Adj**ustment **Type (Synch Current)** that will be used to synchronize (make equal) the current amount with the payoff amount on a service agreement of this type when a budget is cancelled. (The Adjustment Type (Synch Current) field is on the main page.)

NOT_SUPPORTED

Refer to Budget Billing for more information about budgets in general. Refer to Budget Billing Segmentation and Designing and Defining Budget Plans for more information.

Set the **Eligible for Non-Billed Budget** flag to Eligible for Non-Billed Budget if you want SAs of this type to be eligible to be covered by a Non-Billed Budget. If this flag is set to Eligible for Non-Billed Budget, you must also define the **Adjustment Type (Current = 0)** field (on SA Type - Detail).

NOT_SUPPORTED

Refer to Current Amount For SAs Covered By A Non-Billed Budget and SA Types for SAs Covered by Non-Billed Budgets for more information.

- Require Total Amount Switch versus Bill Segment Algorithm
- Allow Recurring Charge Switch versus Bill Segment Algorithm

Require Total Amount Switch versus Bill Segment Algorithm

The following table shows valid combinations of the SA type's required total amount switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not supported in the system. Otherwise, we list typical types of service agreements that will use a combination.

Bill Segment Create Algorithm	SA Type Require Total Amount Switch: Not allowed	SA Type Require Total Amount Switch: Required

Copyright © 2026, Oracle and/or its affiliates.

Apply Rate To Usage Request	Metered services. Lamp services. Misc item services. Company usage. Misc recurring charges whose value is specified in a rate or is taxable.	N/A
Recurring Charge With Auto Stop	Payment arrangements. Merchandise installment plans. Zero-interest loans.	N/A
Recurring Charge For Amount To Bill	N/A	Deposit installment plans.
Recurring Charge	Charitable contributions.	N/A
Billable Charge	One time invoices. Pass through charges	N/A
Loan	N/A	Loans.

Allow Recurring Charge Switch versus Bill Segment Algorithm

The following table shows valid combinations of the SA type's allow recurring charge switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not supported in the system. Otherwise, we list typical types of service agreements that will use a combination.

Bill Segment Create Algorithm	**SA Type Recurring Charge Switch:** Not allowed	**SA Type Recurring Charge Switch:** Optional	**SA Type Recurring Charge Switch:** Required
Apply Rate To Usage Request	Metered services - no budget Lamp services Misc item services. Company usage. Misc recurring charges whose value is specified in a rate or is taxable.	Metered services - budget optional	Metered services - budget required
Recurring Charge With Auto Stop	N/A	Payment arrangements. Merchandise installment plans. Zero-interest loans. SEE NOTE!	Payment arrangements. Merchandise installment plans. Zero-interest loans. SEE NOTE!
Recurring Charge For Amount To Bill	N/A	Deposit installment plans. SEE NOTE!	Deposit installment plans. SEE NOTE!
Recurring Charge	Charitable contributions	N/A	N/A

Copyright © 2026, Oracle and/or its affiliates.

Billable Charge	One time invoices. Pass through charges	N/A	N/A
Loan	N/A	N/A	Loans.

> **Note**
>
> Most recurring charge SA types require a recurring charge amount on their service agreements. However, the above matrix indicates you can have recurring charge SA types where this value is optional. Why? A special algorithm exists in billing that says if the recurring charge amount is 0 (zero) the system will bill the remaining payoff balance or total amount to bill. This algorithm exists so that you can easily bill the amount in one lump sum (i.e., don't bill it in installments).

Where Used

The billing information is used when the system creates a bill segment for service agreements of this type.

SA Type - Rate

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Rate** page to define the rates that may be referenced on service agreements of a given type.

Description of Page

Turn on **Rate Required** if the bill segment creation algorithm for the SA type expects a rate schedule to be referenced on service agreements of this type.

> **NOT_SUPPORTED**
>
> For more information, refer to **Rates** .

When a rate changes within a bill segment's consumption period, the period is divided into separate usage periods – one for each active rate schedule during that period. The usage periods and their related consumption (or billable quantities) are then prepared for billing, and the rate schedule calculations are applied to their associated periods.

This feature is possible if the usage subscription type is configured with the base Dynamic Usage Group Determination algorithm, which determines the applicable usage group for each effective rate schedule. This algorithm requires that the CCB rate schedule has usage group extendable look-ups configured for the rate schedule and device configuration type combinations.

A typical use case is when there is a smart meter rollout in the middle of billing period and there are rate schedule changes that coincide with the meter exchange date. The usage group that is tied to the old meter will have rules that calculate scalar consumption, and the usage group that is tied to the new meter will have rules that calculate interval consumption. The end

Copyright © 2026, Oracle and/or its affiliates.

result is that the customer is billed with scalar consumption for the period where old meter was installed, and billed with interval consumption for the period where new meter is installed.

If the Contract Management module is not turned off, which includes umbrella agreement management, indicate the **Rate Source**. Check SA Only indicates that the rate schedule currently in effect for the service agreement should always be used. Check TOS First, then SA indicates that if the service agreement is linked to a terms of service record, the service agreement's rate could be overridden by one linked to the terms of service record.

The information in the **Rate Schedules** collection defines the rates that may be referenced on service agreements of this type. The following fields are required for each SA Type:

Rate Schedule Specify the rate schedule; its description is displayed adjacent.

Use As Default Turn on this switch for the rate to be defaulted on new service agreements.

- Rate Required versus Bill Segment Algorithm

Rate Required versus Bill Segment Algorithm

The following table shows appropriate combinations of the SA type's rate required switch and the bill segment creation algorithm defined on the SA type's bill segment type. If N/A appears in a cell, the combination is not applicable. Otherwise, we list typical types of service agreements that will use a combination. Be aware that no cross validation exists between the rate required switch and the bill segment creation algorithm when setting up the SA type.

Bill Segment Create Algorithm	SA Type Rate Required Switch: Not allowed	SA Type Rate Required Switch: Rate required on SA
Apply Rate To Usage Request	N/A	Metered services. Lamp services. Misc item services. Company usage.
Recurring Charge With Auto Stop	Payment arrangements. Merchandise installment plans. Zero-interest loans.	N/A
Recurring Charge For Amount To Bill	Deposit installment plans.	N/A
Recurring Charge	Charitable contributions.	N/A
Billable Charge	One time invoices.	Billable charges that require a rate to add-on extra charges (like taxes) or billable charges where the consumption is interfaced and the system is responsible for calculating the charges.
Loan	Loans.	N/A

Where Used

This information is used to default and validate the rate specified on a service agreement. Refer to Service Agreement - Rate Info for more information.

Copyright © 2026, Oracle and/or its affiliates.

SA Type - SP Type

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **SP Type** page to define the service point types that may be referenced on service agreements of a given type.

Description of Page

Turn on **Service Points Required** if at least one service point should be linked to service agreements of this type in order to properly bill the service agreements.

The information in the **SP Types** collection defines the service point (SP) types that may be referenced on this SA type's service agreements. The following fields are required for each SA Type:

SP Type Specify the SP type; its description is displayed adjacent.

> **NOT_SUPPORTED**
>
> For more information about SP types, refer to Designing SP Types.

Where Used

This information is used to validate the types of service points linked to a service agreement. Refer to Service Ag reement - SA / SP for more information.

SA Type - Adjustment Profiles

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Adj Profile** page to define the adjustment profiles that define adjustment types that may be referenced on service agreements of a given type.

Description of Page

Define the **Adjustment Type Profiles** that, in turn, define adjustment types that may be referenced on service agreements of a given type.

> **NOT_SUPPORTED**
>
> For more information about adjustment type profiles, refer to Setting Up Adjustment Type Profiles.

Where Used

This information is used to validate the adjustments linked to the service agreement. Refer to Adjustment - Main Information for more information.

Copyright © 2026, Oracle and/or its affiliates.

SA Type - C&C

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **C&C** page to maintain attributes that affect how the system severs the service agreement when collection attempts fail.

> **NOT_SUPPORTED**
>
> Refer to Designing Your Severance Procedures for more information.

Description of Page

Select the **Debt Class** associated with the SA Type. Any debt on a service agreement of this SA Type will be categorized under this debt class.

Select the **Write Off Debt Class Code** associated with the SA Type. Any debt on a service agreement of this SA Type will be categorized under this debt class during write-off processing.

> **Note**
>
> **Write Off Debt Class vs. Regular Debt Class.** It's important to be aware that a SA type references both a regular debt class and a write-off debt class. The regular debt class controls the collection criteria applied against an account's service agreements. The regular debt class is also used to segregate an account's outstanding balance on several queries in the system. The write-off debt class controls the write-off criteria applied against an account's stopped service agreements. The reason the system supports two different debt classes is because you may categorize your service agreements differently when you try to collect overdue debt versus when you write-off debt. Refer to The Big Picture Of Write Off Processing for more information.

The information in the **Severance Criteria** collection defines the SA Type's severance criteria. Severance criteria define the severance process to be executed for service agreements of a given SA type. The severance process may differ depending on some attribute of the customer or premise. For example, you may have a different severance process if the customer has life support equipment.

> **NOT_SUPPORTED**
>
> The following information is not intuitively obvious. Refer to Designing Your Severance Procedures for more information.

The following fields are required for each instance:

Priority The priority controls the order in which the system determines if the severance process should be applied (the first severance process whose algorithm applies is used). Higher priorities are checked before lower priorities.

Copyright © 2026, Oracle and/or its affiliates.

Severance Criteria Algorithm Select the algorithm to be used to check if the severance process should be initiated for service agreements of this type. If a condition is met, a severance process is created using the associated severance process template.

If you haven't done so already, you must set up this algorithm in the system. To do this:

- Create a new algorithm (refer to Setting Up Algorithms).
- On this algorithm, reference an Algorithm Type that determines if the severance of a service agreement should be processed using the associated **Severance Process Template**.

> **Important**
>
> You must have at least one entry in this collection otherwise the system will not start a severance process when a service agreement of this type needs to be stopped due to non payment. This entry should have the lowest priority code and should reference a **Severance Criteria Algorithm** that references the SV CRIT DFLT algorithm type.

Severance Process Template Specify the severance process template to use to sever the service agreement; its description is displayed adjacent.

Where Used

The debt class has multiple uses:

- The system summarizes an account's debt by debt class on Account - Main Information and Account - Financial Balances.
- Debt class is one component that controls how the system analyzes an account's overdue debt (the others are the account's collection class and currency). Refer to Different Collection Criteria For Different Customers And Different Debt for more information.
- Write off debt class is one component that control how the system writes off an account's stopped service agreements. Refer to Different Write-Off Criteria For Different Customers And Different Debt for more information.

The severance criteria are used when a collection event is activated that indicates that service should be severed.

SA Type - Billable Charge Template

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **BC Template** page to define the billable charge templates that can be used on service agreements of a given type.

> **Note**
>
> **Only billable charges have billable charge templates.** Only service agreements that are defined as Billable Charges (in the Special Role on the Details window) may use the grid on this window.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

The information in the **Billable Charge Template** collection defines the SA Type's permissible billable charge templates. A billable charge template contains the default bill lines, amounts and distribution codes used to levy a one-off charge. The following fields are required for each template:

Billable Charge Template Specify the billable charge template. Its description is displayed adjacent.

Use As Default Turn on this switch for the template to be defaulted on new billable charges linked to service agreements of this type (if any).

> **NOT_SUPPORTED**
>
> For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Where Used

This information is used to limit the billable charge templates that can be used for a given SA type.

SA Type - Characteristics

To define characteristics common to all service agreements of a given type, open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

Use the characteristics collection to define a **Characteristic Type**, **Sequence**, and **Characteristic Value** common to all service agreements of this type.

> **Note**
>
> You can only choose characteristic types defined as permissible on a SA Type record. Refer to Setting Up Characteristic Types & Their Values for more information.

SA Type - Algorithms

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Algorithm** page to define the algorithms that should be executed for service agreements of a given type.

Description of Page

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (descriptions of all possible events are provided below).

Copyright © 2026, Oracle and/or its affiliates.

- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

> **Warning**
>
> These algorithms are typically significant processes. The absence of an algorithm may prevent the system from operating correctly.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description

Copyright © 2026, Oracle and/or its affiliates.

Bill Completion	Optional	These algorithms are executed whenever a bill is completed for an account that contains a non-canceled service agreement of this type. The following situations necessitate the definition of a completion algorithm on an SA type: - As explained under Technical Implementation of A/R Transfer and Technical Implementation of Routing Billable Charges To Service Providers, when a bill is completed, the system needs to set up the data necessary to interface any "master" SA's charges to the service provider and to transfer the receivable balance from the customer to the service provider. The system will only do this if you specify an appropriate algorithm on the master SA types. - As explained under Billing For SAs Covered By The Non-Billed Budget, when a bill is completed for accounts that have a Non-Billed Budget SA, the system needs to distribute the Non-Billed Budget's credit balance to the covered SAs. The system will only do this if you specify an appropriate algorithm on the NBB SA types. - As explained under Overpayment Segmentation, when a bill is completed, the system may apply an excess credit from a prior overpayment to an account's service agreements. **Note**. Algorithms of this type are called for all non- Canceled service agreements, regardless of whether or not they are billed. If your algorithms should only be processed under certain conditions (for example, only process this algorithm for Active service agreements), then it is the responsibility of the algorithm to check the conditions before continuing.

Copyright © 2026, Oracle and/or its affiliates.

Bill Eligibility	Optional	These algorithms are executed during billing and allow logic to be introduced to check additional criteria to determine if an SA is eligible for billing.
Break NBB SA	Optional	These algorithms are executed when a Non-Billed Budget is manually stopped via the **Non-Billed Budget maintenance page** . Note that the Payment Arrangement algorithm and the Break Pay Arrangement algorithm are mutually exclusive.
Break Pay Arrangement	Optional	These algorithms are by executed by severance events when the event is created for payment arrangement SAs. This algorithm should be specified on SA types with a special role of Payment Arrangement to perform special actions that take place when a customer breaks a payment arrangement. Refer to Monitoring Payment Arrangements for more information about breaking payment arrangements.
Budget Eligibility	Optional	These algorithms are executed when determining if a service agreement is eligible for budget. Algorithms of this type are only applicable on SA types that are marked as eligible for budget and may be used to override that setting and indicate that the service agreement is not eligible. For example, maybe service agreements in a certain rate are not eligible. Or perhaps service agreements with a given characteristic value are not eligible.
Cut Process Rule	Optional	These algorithms are executed to create a cut process for service agreements of this type. Refer to The Big Picture Of Cut Processes for more information.

Copyright © 2026, Oracle and/or its affiliates.

FT Freeze	Optional	These algorithms are executed whenever a financial transaction is frozen that is linked to a service agreement of this type. The following situations necessitate the definition of an FT freeze algorithm: - As explained under Technical Implementation of Routing Consumption To Service Providers, when a master SA's bill segment is frozen, the system must check if there are any service providers who need the bill segment's consumption. If so, it sets up the data necessary to interface the master SA's consumption (snapshot on the bill segment) to the service provider(s). The system will only do this if you specify an appropriate FT Freeze Algorithm on the master SA types. - As explained under Technical Implementation of Paying The Service Provider, when a financial transaction (FT) is frozen that is associated with a sub SA, the system must check if this FT should trigger the "payment" of a service provider. If so, it has to create an adjustment to increase how much we owe the service provider. The system will only do this if you specify an appropriate FT Freeze Algorithm on the sub SA types.

Copyright © 2026, Oracle and/or its affiliates.

High Bill Amount	Optional	This algorithm type checks for high bill amounts during batch billing. A bill threshold amount can be defined on a service agreement. By default the system compares this threshold amount to the bill segment amount during batch billing. If the bill segment amount exceeds the given threshold value, a bill error is generated. No proration takes place on the threshold amount if the bill segment's consumption period falls outside of the rate frequency's normal period. This algorithm may be used to override the default system behavior. It will prorate the SA's bill threshold amount before comparing it to the batch bill segment amount if the consumption period falls outside of the rate frequency's normal period.
Landlord Reversion	Optional	These algorithms are used to create, update or cancel a service agreement for a landlord when service is started, stopped or updated for a premise that references a landlord agreement. Algorithms of this type are called: - When starting a new SA (start initiation) - When stopping an SA (stop initiation) - When canceling a pending start SA - When changing the stop / start dates of back-to-back service agreements
Loan Interest Charge	Optional	These algorithms are executed whenever the interest charge needs to be calculated for a loan, such as when the loan amortization schedule is created and when a bill segment is created for a loan SA. This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information.

Copyright © 2026, Oracle and/or its affiliates.

Loan Periods and Amount	Optional	These algorithms are executed whenever a user clicks the Calculate button on **Loan - Main** or on the Start SA Confirmation dialog for a loan SA. It calculates either the number of periodic payments or the payment amount (depending on whether the user specifies the number of payments or the payment amount as input). This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information.
Loan Schedule	Optional	These algorithms are executed whenever the system needs to create a **loan amortization schedule** for a loan, such as when you renegotiate the terms of a loan on **Loan - Main** . This algorithm should be specified on SA types with a special role of Loan. Refer to Defining Loan Options for more information.
Override Proration Days	Optional	This algorithm is invoked from rate application, right after base has determined if proration applies. It is used to override base days, normal days and “is within proration window” indicator, which were determined by base logic. Only one algorithm may be plugged-in to this system event. This is available only in the calculation rule based rating engine.

Copyright © 2026, Oracle and/or its affiliates.

Payment Arrangement	Optional	These algorithms are by executed to handle the creation, breaking and canceling of payment arrangement SAs. This algorithm should be specified on SA types with a special role of Payment Arrangement to perform special actions that take place during the lifecycle of a payment arrangement. Refer to Monitoring Payment Arrangements for more information about payment arrangements. Note that the Payment Arrangement algorithm and the Break Pay Arrangement algorithm are mutually exclusive.
Payment Freeze	Optional	These algorithms are executed whenever a payment is frozen. The following situations necessitate the definition of such an algorithm: - For a loan SA, such an algorithm is required to create a frozen adjustment that transfers any credit balance resulting from an overpayment to the loan's principal balance. Refer to Defining Loan Options for more information.
Pre-Bill Completion	Optional	These algorithms are executed immediately prior to bill completion when a bill contains a bill segment for a service agreement whose SA type has such an algorithm. The following situations necessitate the definition of such an algorithm: - If you want to delete a bill segment that's in error on the last night of a bill cycle when there are other bill segments that aren't in error, use such an algorithm.

Copyright © 2026, Oracle and/or its affiliates.

Process NBB Scheduled Payment	Optional	These algorithms are executed by the NBB Scheduled Payment Processing background process whenever a scheduled payment is due. If the Non-Billed Budget SA is unmonitored, this algorithm is not called. This algorithm should be specified on Non-Billed Budget SA types to create the necessary adjustments for the Non-Billed Budget SA.
Proposal SA Acceptance	Optional	These algorithms are executed when a proposal service agreement is accepted. Refer to Enabling The Creation Of A Real Service Agreement for more information.
Proposal SA Bill Segment Generation	Optional	These algorithms are executed to generate simulated bills segments for a **proposal service agreement** . Refer to **Enabling The Generation Of Simulated Bill Segments** for more information.
Proposal SA Creation	Optional	These algorithms are executed when a **proposal service agreement** is created. Refer to **Enabling The Automatic Generation Of Billing Scenarios** for more information.
SA Activation	Optional	These algorithms are executed when a service agreement status changes from Pending Start to Active. It performs any additional activities that are necessary to activate an SA. The following situations necessitate the definition of such an algorithm: If you want to create a customer contact to indicate that a non-billed budget has been activated, use such an algorithm.

Copyright © 2026, Oracle and/or its affiliates.

SA Cancel	Optional	These algorithms are executed when a service agreement status changes to Canceled. It performs any additional activities that are necessary to cancel an SA. An example of when you may use this algorithm is that perhaps your business rules dictate that the creation of a payment arrangement should create a credit rating history transaction. When a payment arrangement SA is canceled, the credit rating should be updated with an end date.
SA Creation	Optional	These algorithms are executed when a service agreement is created. The following situations necessitate the definition of such an algorithm on an SA type: - If you want to create a To Do entry whenever a new service agreement of a given type is added, specify such an algorithm. - If you want to automatically activate SAs of a given type (instead of waiting for the background SA activation process to run), specify such an algorithm. - If you want to create a Workflow Process when a service agreement of a given type is added, specify such an algorithm.
SA Information	Optional	We use the term "SA information" to describe the basic information that appears throughout the system to describe a service agreement. The data that appears in "SA information" is constructed using this algorithm. Plug an algorithm into this spot to override the "SA information" algorithm on installation options or the system default "SA information" if no such algorithm is defined on installation options.

Copyright © 2026, Oracle and/or its affiliates.

SA Renewal	Optional	These algorithms are executed by the Service Agreement Renewal background process whenever an SA is due for renewal or when the user clicks the **Renew** button (for **Non-Billed Budgets**). It performs any activities that are necessary to renew an SA and returns the new renewal and expiration dates for the SA.
SA Stop	Optional	These algorithms are executed whenever a service agreement's status changes from pending stop to stopped. The following situations necessitate the definition of a SA stop algorithm: - For Non-Billed Budgets to distribute any remaining credit balance from the Non-Billed Budget SA to the covered SAs, you must specify such an algorithm. - For service credit memberships that have a refundable membership fee, an SA Stop algorithm attempts to refund the fee if this is the last SA linked to the membership that is being stopped.

Copyright © 2026, Oracle and/or its affiliates.

SA Stop Initiation	Optional	These algorithms are executed whenever a service agreement's status becomes pending stop. The following situations necessitate the definition of a stop initiation algorithm on an SA type: - As explained under Finalizing Pending Stops, service agreements are normally transitioned from pending stop to stopped by a background process (or manually). For Non-Billed Budget SAs to transition to stopped automatically (without waiting for the background process), you must specify such an algorithm. **When does a SA become pending stop?** Service agreements typically become pending stop when a user initiates a request to stop service on Start Stop - Main. A severance process with an Expire SA severance event causes a service agreement to become pending stop (when the event is executed). Additionally, the Stop Expired Service Agreements background process starts the process to initiate the stop of an SA when the expiration date is on or before the process date.
Start Stop Field Work	Optional	These algorithms are executed to create the field activities necessary to start and stop service. Refer to Starting Service and Field Activities and to Stopping Service and Field Activities for a description of when algorithms of this type are called. The following situations necessitate the definition of a start stop fieldwork creation algorithm: - If a service agreement has field activities created to start and stop service at its service points, its SA type must have an appropriate start stop field work creation algorithm.

Copyright © 2026, Oracle and/or its affiliates.

SA Type - Billable Charge Overrides

The BCU2 - Create Billable Charge background process is responsible for creating billable charges for each billable charge upload staging record interfaced into the system. This process will override the values of the various switches referenced on a billable charge upload staging line if the respective service agreement's SA type has an override value for the billable charge upload staging line's billable charge line type.

> **Note**
>
> **This information is optional.** If you don't need to override the values of a **Billable Charge Line Type** you don't need to set up this information.

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **BC Upload Override** page to define override values for a given SA Type / Billable Charge Upload Staging Line Type.

Description of Page

Use the **Billable Charge Overrides** collection to define values to be overridden on billable charge lines uploaded from an external system (refer to the description above for the details). The following switches may be overridden for a given **SA Type** and **Billable Charge Line Type.**

- Use the **Show on Bill** switch to define the value to be defaulted into the Show on Bill indicator on billable charge upload lines that reference this line type.
- Use the **Appear in Summary** switch to define the value to be defaulted into the App in Summary indicator on billable charge upload lines that reference this line type.
- Use **Memo Only, No GL** switch to define the value to be defaulted into the Memo Only, No GL indicator on billable charge upload lines that reference this line type.
- Use **Distribution Code** to define the value to be defaulted into the Distribution Code on billable charge upload lines that reference this line type.

SA Type - Contract Option Type

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **Contract Option Type** page to define the contract option types, which are valid for service agreements of a given type.

> **Note**
>
> **This tab may not appear.** This tab is suppressed if the interval billing Contract Options module is turned off.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

If the SA Type's special role flag is set to Bill Determinants Required, you may define the **Contract Option Types** that are valid for contract options linked to a service agreement of this type.

Where Used

This information is used to validate the contract options linked to the service agreement. Refer to Service Agreement - Contract Options for more information.

SA Type - NBB Recommendation Rule

Open **Admin**, and then **Customer**, and then **SA Type**, and then **Search** and navigate to the **NBB Rec'n Rule** page to define the recommendation rules that are valid for Non-Billed Budget SAs of this type.

Description of Page

If the SA Type's special role is Non-Billed Budget, you may define the **Recommendation Rules** that are valid on non-billed budget SAs of this type. Check the **Use As Default** box to indicate the default recommendation rule for service agreements of this type.

> **NOT_SUPPORTED**
>
> For more information about Non-Billed Budgets, refer to **Defining Non-Billed Budget Options** .

Where Used

The Non-Billed Budget recommendation rules are used to recommend the payment amount and payment schedule for Non-Billed Budget service agreements. Refer to **Maintaining Non-Billed Budgets** for more information.

Setting Up Start Options

Start options save users time and prevent data entry errors because they default many values on a service agreement (e.g., the rate schedule, recurring charge amount, contract riders, contract terms, characteristics, terms and conditions, etc. can all be defaulted onto a service agreement from a start option).

A SA type may have zero or more start options.

- A SA type without start options is usually one that has a very limited number of options. For example, if a SA type has a single valid rate and no customer-specific contract values, you don't need to setup a start option (the SA's default rate can default based on the information defined when you setup the SA type).
- A SA type with multiple start options is one where many different permutations are possible. For example, a SA type that can have multiple rates and each rate can have multiple riders is a good candidate for start options (where each start option will default, for example, a specific rate and set of contract riders).

Copyright © 2026, Oracle and/or its affiliates.

When the Start/Stop transaction is used to start service AND the service being started uses a SA type with start options, the user is asked to select one of the start options. The service agreement that's created is populated with fields from the start option.

When the Order transaction is used to start service, the user selects start options, but only indirectly. It works like this:

- A user selects a "package" of services to start service for a customer.
- A "package" contains one or more start options.
- When a user selects a package, the system creates a service agreement for each start option on the package (and defaults the information on each service agreement from the respective start option).

A start option's default values may change over time (i.e., the information on a start option is effective-dated). The start service logic uses the version of the start option that is effective on the day service starts.

Start options can cause a great deal of information to be populated on a service agreement. There are several ways to change this default information:

- A user may override this information using the Service Agreement transaction.
- If the service agreement is in the pending start state, you can use the Start/Stop - Pending SAs page to change the service agreement's SA type and/or start option.
- If the service agreement is in active or pending stop states, a button appears on Service Agreement - Main called **Apply New Start Option**. When pressed, the user is allowed to define a start option and the date its terms become effective on the service agreement. Refer to Changing A Start Option for the details of this functionality.

The topics in this section describe how to setup start options.

> **Note**
>
> **The merge transaction can save setup time.** The Start Options Merge transaction can be used to construct a start option by copying pieces from other start options.

- Start Option Considerations - Rate-Oriented SA Types
- Start Option Considerations - SA Types That Use Recurring Charges
- Start Option Considerations - Initial Adjustment SA Types
- Start Option - Main
- Start Option - Rate Info
- Start Option - Characteristics & Qty
- Start Option - Contract Option

Start Option Considerations - Rate-Oriented SA Types

To understand the following discussion, you should be familiar with the following concepts:

- SA types that use rates have one or more valid rate schedules. Only these rate schedules may be defined on service agreements of a given type.

Copyright © 2026, Oracle and/or its affiliates.

- One of a SA type's rate schedules may be designated as the "default" rate. The system assigns the "default" rate to new service agreements when a CSR doesn't choose a start option.
- Start options may be used by customer service reps when a service is started for a customer. A start option causes the customer's new service agreement to be populated with a specific rate and contract terms (e.g., contract riders, contract values). The use of start options is not allowed if the service's SA type does not require a start option. The use of start options is required if the service's SA type requires a start option.

Whether or not a SA Type has start options is dependent on the following factors:

- If a SA Type has only one valid rate and the rate doesn't use customer-specific contract terms (e.g., contract riders, contract values), the SA Type does not need any start options. Why? Because the system default's the SA type's default rate on new service agreement when no start option is used at start time.
- If a SA Type has only one valid rate, but under unusual circumstances, it uses customer-specific contract terms, you'll want an option for every situation (both the standard one and the unusual ones).
- If a SA Type has a single rate with a variety of contract terms (which, by definition, are different for each customer), then you'd want a start option for each permissible combination of contract terms. You'd also want to turn on the SA type's Start Option Required switch to make your CSR's pick one of the start options when service is started (rather than let the system use the SA type's default rate).
- If a SA type has multiple valid rates and a variety of contract terms (a combination of the previous two points), you'd need a start option for each permissible combination. You'd also want to turn on the SA type's Start Option Required switch to make your CSR's pick one of the start options when service is started (rather than let the system use the SA type's default rate).

Start Option Considerations - SA Types That Use Recurring Charges

To understand the following discussion, you should be familiar with the following concepts:

- Many SA types use the recurring charge algorithms to generate the bill segments. For example, if you let a customer pay for a $900 heat pump in 3 installments of $300, you'd have an SA type called CA/MERCH-I and indicate it uses the recurring charge algorithm.
- When a CSR creates an SA type that uses recurring charge algorithms, they can enter the recurring charge amounts on Start/Stop Service - Start Confirmation window OR they can specify a start option on this window and let the system populate the recurring charge amount. For example, if you let a customer pay for a $900 heat pump in 3 installments of $300, you can set up a start option called HP 3PAY with an installment amount of $300.
- Depending on the type of service being started, you might also need to generate an adjustment when service is started in order to initialize the total debt. For example, when a customer buys a heat pump we'll need an adjustment issued to realize the entire $900 of revenue.

The following table provides examples of recurring charge SA types with several typical start options:

CIS Division/ SA Type	Start Option	Adjustment Type	Install Amount	Comments

Copyright © 2026, Oracle and/or its affiliates.

CA/CHARITY	DONATE $5		5	This causes a charitable contribution to be created with an installment amount of $5.
CA/CHARITY	DONATE $10		10	This causes a charitable contribution to be created with an installment amount of $10.
CA/PA-REGU	PAY $10 PM		10	This causes a payment arrangement to be created with an installment amount of $10.
CA/PA-REGU	PAY $20 PM		20	This causes a payment arrangement to be created with an installment amount of $20.
CA/PA-UNRE	PAY $10 PM		10	This causes a payment arrangement to be created with an installment amount of $10.
CA/PA-UNRE	PAY $20 PM		20	This causes a payment arrangement to be created with an installment amount of $20.
CA/MERCH-I	HP 3PAY	HEATPUMP	300	This causes a merchandise service agreement to be created with an installment amount of $300. It also causes an adjustment to be issued to realize the $900 of revenue.

Copyright © 2026, Oracle and/or its affiliates.

CA/MERCH-I	BBQ 3PAY	BBQ	250	This causes a merchandise service agreement to be created with an installment amount of $250. It also causes an adjustment to be issued to realize the $750 of revenue.
CA/DEP-I	PAY $40 PM		40	This causes a deposit service agreement to be created with an installment amount of $40. The CSR would be required to define the total deposit amount to be billed over the life of the service agreement on the secondary SA window on the Start Account window.

Start Option Considerations - Initial Adjustment SA Types

To understand the following discussion, you should be familiar with the following concepts:

- Some SA types depend on an adjustment to book their initial debt. For example, the CA/MERCH-I SA type requires an initial adjustment to book the payoff amount for the entire cost of the merchandise. Another example would be a loan service agreement (refer to Booking The Principal Amount Using An Adjustment for the details).
- When a CSR starts an SA type that requires an initial adjustment, they can create the adjustment immediately after starting service OR then can specify a start option when they start service and let the system generate the adjustment. You could let the system create the adjustment to book the $900 associated with the heat pump as shown in the previous section.

The following table provides examples of SA types that use adjustments with several typical start options:

CIS Division/ SA Type	Start Option	Adjustment Type	Install Amount	Comments

Copyright © 2026, Oracle and/or its affiliates.

CA/MERCH-I	HP 3PAY	HEATPUMP	300	This causes a merchandise service agreement to be created with an installment amount of $300. It also causes an adjustment to be issued to realize the $900 of revenue.
CA/MERCH-I	BBQ 3PAY	BBQ	250	This causes a merchandise service agreement to be created with an installment amount of $250. It also causes an adjustment to be issued to realize the $750 of revenue.
CA/CONNECT	CONNECT	CONNECT		This causes an adjustment to be issued to realize the connection charge.

Start Option - Main

Open **Admin**, and then **Customer**, and then **SA Type Start Option**, and then **Add** to define a SA type's start options.

Description of Page

Every start option is uniquely identified by the following fields:

CIS Division & SA Type Enter the Division and SA type to which the start option is linked.

Start Option Enter the unique identifier of the option. Pick something easy to recognize as this will be used by CSRs to pick an option when they start service.

Effective Date Enter the earliest effective date. It should be the same as the earliest effective date of the start option's rate (although it doesn't hurt for it to be earlier). The date defaults to the current date. (The status, below, should be Active.)

The remaining fields further describe a start option.

Enter a **Description** for the start option.

Indicate its **Status**. For new start options, the status should be Active. When it's no longer applicable, change it to Inactive.

Enter the primary **Rate Schedule** that should be defaulted onto service agreements created using this option. Refer to Start Option Considerations For SA Types That Use Rates for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> For more information about a service agreement's rates, refer to Service Agreement - Rate Info.

Enter the **Adjustment Type** that should be generated, if any, when service is started using this option. Refer to Start Option Considerations For SA Types That Use Initial Adjustments for more information.

Enter the **Recurring Charge Amount** that should be defaulted onto service agreements created using this option. This field is only visible when the SA type allows recurring charges. In addition, the prompt for this field is defined on the SA type table on the billing window (e.g., it could appear as Payment Amount, Budget Amount, or Installment Amount). Refer to Start Option Considerations For SA Types That Use Recurring Charges for more information.

Enter the **Currency Code** in which monetary amounts are denominated.

> **Note**
>
> The currency code defaults from the installation record.

Enter the **Total Amount to Bill** that should be defaulted onto service agreements created using this option. This is useful to initiate either a loan or a deposit. The prompt for this field is defined on the SA type table on the billing window (e.g., it could be Deposit Amount or Loan Amount).

Use **Number of Periods** to default the number of payment periods of service agreements created using this option. This field is only allowed for SA types with special role of Loan. Refer to The Terms Of A Loan Are Stored On A Service Agreement for more information.

If your SA Type has a special role of Billing Determinants Required, you may enter the **Cutoff Time** and **Start Day Option** that should be defaulted onto service agreements that are created using this option. For start day option, you may choose **Current Day** or **Previous Day**.

If your SA Type has a special role of Billable Charge, then you can setup the start option to automatically create a billable charge when a service agreement is created using this start option. For example, you might have a start option that automatically creates a "one-time invoice" service agreement along with a "tree trimming" billable charge. To use this feature you should turn on the **Create Billable Charge** switch and specify the **Billable Charge Template** that will be used to create the billable charge. These fields are only allowed for SA types with special role of Billable Charge.

> **NOT_SUPPORTED**
>
> Refer to Setting Up Billable Charge Templates for more information about templates. Refer to An Easier Way To Create One Time Charges for an example of how you can setup a campaign with packages that use this type of start option.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The duplicate action in the page actions toolbar enables you to copy another start option. Refer to Duplicate Button in the system wide standards document for more information.

Start Option - Rate Info

Open **Admin**, and then **Customer**, and then **SA Type Start Option**, and then **Search** and navigate to the **Rate Info** page to define the start option's default values for contract riders and contract values.

Description of Page

The information in the **Contract Riders** collection defines the contract riders to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Bill Factor The bill factor defines the type of rider. You may only reference bill factors designated as being applicable for contract riders.

Number of Days The number of days the rider should be in effect. This value is used by the system to set the stop date on the service agreement's contract rider. If the rider has no expiration, set this field to 0. Default note: this field will be set to 0 if left blank.

> **NOT_SUPPORTED**
>
> For more information about a service agreement's contract riders, refer to Service Agreement - Contract Riders.

The information in the **Contract Values** collection defines the contract values to be defaulted onto service agreements created using this start option. The following fields are required for each event:

Bill Factor The bill factor defines the type of value. You may only reference bill factors designated as allowing values in contract terms.

Number of Days The number of days the value should be in effect. This value is used by the system to set the stop date on the service agreement's contract value. If the value has no expiration, set this field to 0.

Value The amount of the contract value.

> **NOT_SUPPORTED**
>
> For more information about a service agreement's contract values, refer to Service Agreement - Contract Values.

Copyright © 2026, Oracle and/or its affiliates.

Start Option - Characteristics & Qty

Open **Admin**, and then **Customer**, and then **SA Type Start Option**, and then **Search** and navigate to the **Characteristics & Qty** page to define the start option's default values for characteristics and contract quantities.

Description of Page

The information in the **Characteristics** collection defines the characteristics to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Characteristic Type This defines the type of characteristic. Note: you may only define characteristics valid on service agreements.

Characteristic Value This defines the characteristic value that will be defaulted.

> **NOT_SUPPORTED**
>
> For more information about a service agreement's characteristics, refer to Service Agreement - Characteristics.

The information in the **Contract Quantity** collection defines the contract quantities to be defaulted onto service agreements created using this start option. The following fields are required for each instance:

Contract Quantity Type This defines the type of contract quantity.

Contract Quantity The amount of the contract quantity.

> **NOT_SUPPORTED**
>
> For more information about a service agreement's contract quantities, refer to Service Agreement - Contract Quantity.

Start Option - Contract Option

> **Note**
>
> **This tab may not appear.** This tab is suppressed if the interval billing Contract Options module is turned off.

Open **Admin**, and then **Customer**, and then **SA Type Start Option**, and then **Search** and navigate to the **Contract Option** page to define the start option's contract option default values.

Copyright © 2026, Oracle and/or its affiliates.

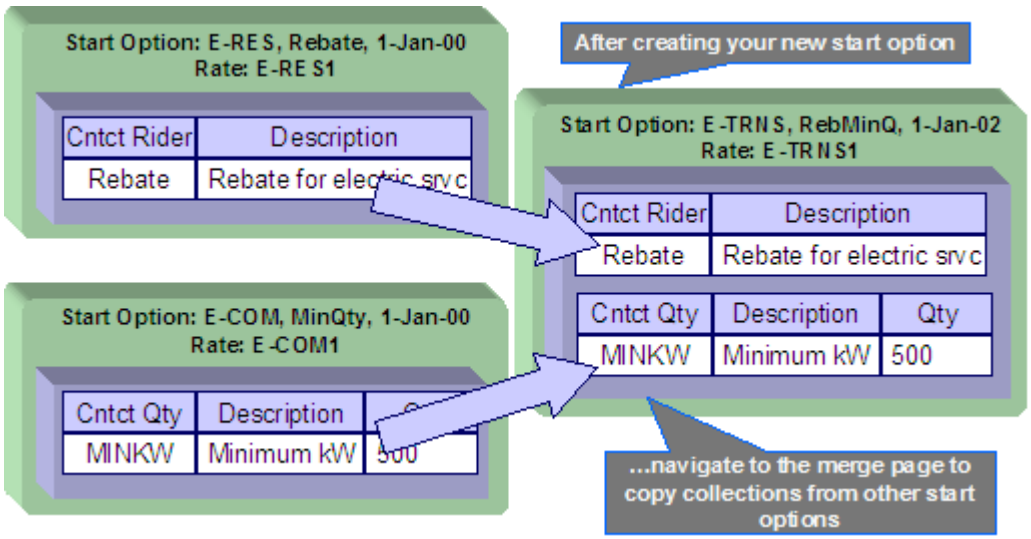
Description of Page

The collection of contract option types will be used to link contract options to a service agreement that will be created with this start option. If the contract option to create for the service agreement will be SA Specific, simply indicate the **Contract Option Type**. This will cause a new contract option to be created with this contract option type and the new contract option will be linked to the new SA at start time. If the contract option to create for the service agreement will be Shared, indicate the contract option type and the **Contract Option ID**. This will cause the specified contract option to be linked to the new SA at start time.

Start Option Merge

Use this page to modify an existing start option by copying information from other start options. This page may be used to copy records from the contract rider, contract value, contract quantity, characteristic, contract option and TOU contract value collections from one or more existing start options to another.

> **Note**
>
> The target start option must exist prior to using this page. If you are creating a new start option, you must first go to the Start Option page to add the new start option and then navigate to the merge page to copy collection information.



> **Note**
>
> **Duplicate versus Merge**. The **Start Option** page has Duplication capability. You would duplicate a start option if you want to a) create a new start option AND b) populate it with all the information from an existing start option. You would use the start option merge page if you want to build a start option using pieces of one or more start options.

- Start Option Merge - Main

Copyright © 2026, Oracle and/or its affiliates.

- Removing A Row From A Grid
- Adding A New Row To A Start Option
- Removing An Uncommitted Row From A Start Option
- Start Option Merge - Characteristics and Quantities
- Start Option Merge - Contract Option
- Start Option Merge - Terms and Conditions

Start Option Merge - Main

Open **Admin**, and then **Customer**, and then **SA Type Start Option Merge.**

Description of Page

Select the **Original Start Option**, which is the target for merging the start option collection information.

Select the **Merge From Start Option**, which is your template start option to copy the collections from.

> **Note**
>
> You may only copy information from one Merge From start option at a time. If you wish to copy information from more than one start option, select the first Merge From start option, copy the desired records, Save, then select the next Merge From start option.

The left portion of the page will display any existing records in the collections for the original start option. The right portion of the page will display the existing records in the collections for the Merge From start option.

You may use the **Copy All** button to copy all the records in all the collections from the Merge From start option to the Original start option. If you do not choose to copy all, you may copy records individually as described below.

The left portion of the **Contract Riders** collection initially displays existing contract riders linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Bill Factor** and **Number of Days** for each contract rider is displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The left portion of the **Contract Values** collection initially displays existing contract values linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Bill Factor**, **Number of Days** and **Value** for each contract value is displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The topics, which follow, describe how to perform common maintenance tasks:

Removing A Row From A Grid

Copyright © 2026, Oracle and/or its affiliates.

If you wish to remove a record linked to the Original start option, click the "-" button to the left of the record.

Adding A New Row To A Start Option

You may move any of the records from the Merge From start option to the original start option by selecting the left arrow adjacent to the desired row. Once a record is moved it will disappear from the Merge From information and appear in the Original information with the word Merge in the Merge Type column.

Removing An Uncommitted Row From A Start Option

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

Start Option Merge - Characteristics and Quantities

Open **Admin**, and then **Customer**, and then **SA Type Start Option Merge** and navigate to the **Characteristics and Quantities** page to copy rows in the characteristic and contract quantity collections.

Description of Page

The left portion of the **Characteristics** collection initially displays existing characteristics linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Characteristic Type** and **Characteristic Value** for each characteristic are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

The left portion of the **Contract Quantity** collection initially displays existing contract quantities linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Contract Quantity Type** and **Contract Quantity** for each contract quantity are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

NOT_SUPPORTED

Refer to Start Option Merge - Main for more information about how to perform common maintenance tasks for the grids displayed on this page.

Start Option Merge - Contract Option

Open **Admin**, and then **Customer**, and then **SA Type Start Option Merge** and navigate to the **Contract Option** page to copy rows in the contract options collection.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **This tab may not appear.** This tab is suppressed if the interval billing Contract Options module is turned off.

Description of Page

The left portion of the **Contract Option** collection initially displays existing contract option information linked to the original start option. In the **Merge Type**, you will see the word Original, for any of these records. The **Contract Option Type**, **Contract Option ID** and **Description** of the contract option for each contract option row are displayed. In the right portion of the collection, the existing records in the merge from start option are displayed initially.

> **NOT_SUPPORTED**
>
> Refer to Start Option Merge - Main for more information about how to perform common maintenance tasks for the grids displayed on this page.

Start Option Merge - Terms and Conditions

Open **Admin**, and then **Customer**, and then **SA Type Start Option Merge** and navigate to the **Terms and Conditions** page to copy terms and conditions (T&Cs).

Description of Page

The left side of the **Terms and Conditions** grid initially displays the T&Cs linked to the original start option. On the right side, the T&Cs linked to the merge from start option are displayed initially.

> **NOT_SUPPORTED**
>
> Refer to Start Option Merge - Main for a description of how to perform common maintenance tasks for the grids displayed on this page.

Defining SA Relationship Options

We use the term "SA Relationship" to describe functionality that supports the following situations:

- When companies other than your own provide a service to your customers AND you have some type of interaction with these companies. For example, in a deregulated market, customers deal with both distribution and energy supply companies. These companies typically exchange a great deal of information about their joint customers.

Copyright © 2026, Oracle and/or its affiliates.

- When multiple rates are associated with a service (where each rate corresponds with a sub-category of service). For example, most water companies charge for both water and wastewater service using separate rates for each. While it is possible to set up water and wastewater as separate service agreements, the SA relationship functionality allows you to set up a single "master" service agreement (for the water service) and associate with it a "sub" service agreement (for the wastewater service).
- When a party representing a group of customers negotiates a contract that is applied over and above those of the individual service agreements. For example, the head office of a national chain may negotiate for additional discounts that should be calculated together or individually. The SA relationship functionality may be used to track the covered service agreements and to calculate and transfer discounts to the head office's service agreement. Refer to the special discounts section.

> **Warning**
>
> Setting up the SA relationship control tables is as challenging as your organization's business rules. If you don't have requirements similar to those described above, you don't have to set up anything. If you have these types of requirements, your setup process will be taxing as you must design and set up control tables that manage the financial and consumption interactions that take place between you, your customers, and the various service providers.

The topics in this section describe tables that control your SA relationship functionality.

- The Big Picture of SA Relationships and Service Providers
- Designing Your SA Relationship Types and Service Providers
- Setting Up SA Relationship Information
- Negotiated Terms

The Big Picture of SA Relationships and Service Providers

You must set up service providers if companies other than your own provide a service to your customers AND you have some type of interaction with these companies. You will have one service provider for each such company. For example, if you are a distribution company in a deregulated market, you will have a service provider for each company that provides any of the following services:

- Energy (commonly referred to as Energy Supply Companies, Energy Service Providers, Retailers, and Suppliers)
- Meter service (commonly referred to as Meter Service Providers and Meter Agents)
- Meter reading (commonly referred to as Meter Data Management Agencies and Meter Reading Service Providers)
- Billing (commonly referred to as Billing Agents)

The topics in this section provide background information about service providers.

- Persons and Service Providers
- Service Providers Are Linked To Service Agreements

Copyright © 2026, Oracle and/or its affiliates.

- Service Providers May Change Over Time
- How To Set Up SA Relationships On A Customer's Service Agreement
- When Your Company Is A Service Provider
- Service Providers Have To Communicate About Customers
- Relationships Between Service Providers
- A Service Agreement Can Have Many Types Of Relationships
- Billing Relationships
- Consumption Relationships
- Deposits Issues
- Credit and Collection Issues
- An Object-Oriented Perspective Of Service Providers
- How Do You Communicate With Service Providers?
- Off Cycle Switch

Persons and Service Providers

A great deal of information about your service providers is defined using a person. For example, a service provider's name, address, phone numbers, electronic ID's, etc. are all defined on the person object.

In addition, every service provider must have a service provider object created for it. The service provider object contains information about a provider's relationships with the customer and your organization, for example:

- Do you calculate bills for the service provider? If so, do you have their rates or do they interface their charges to you? Refer to Billing Relationships for more information.
- Do you send the customers' consumption to the service provider? Do they send it to you? Refer to Consumption Relationships for more information.
- How are financial settlements between your organization and the service provider implemented (do you pay them when you get paid?, do you purchase the receivable for them?, etc.). Refer to Service Providers Have Service Agreements Too for more information.
- Etc.

NOT_SUPPORTED

Refer to Designing Your SA Relationship Types and Service Providers for more information.

Note

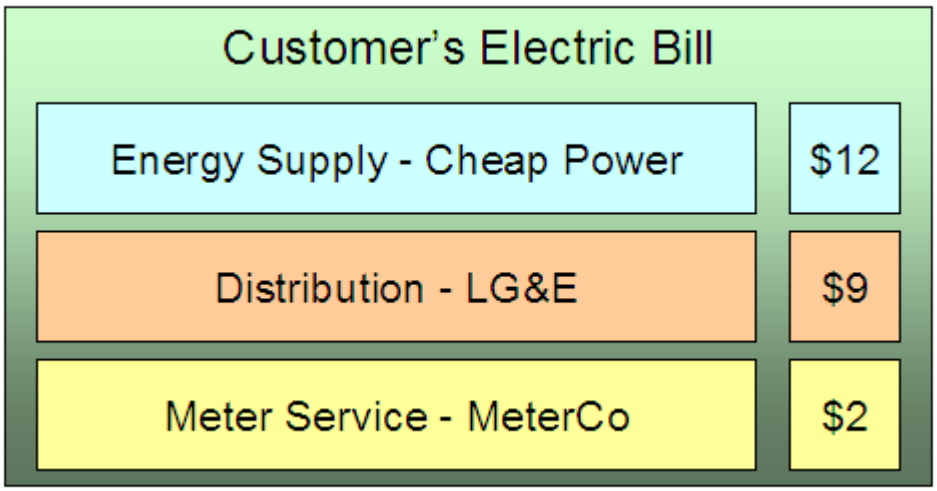
In some situations, you will need to set up a service provider for your own company. Refer to When Your Company Is A Service Provider for the details.

Copyright © 2026, Oracle and/or its affiliates.

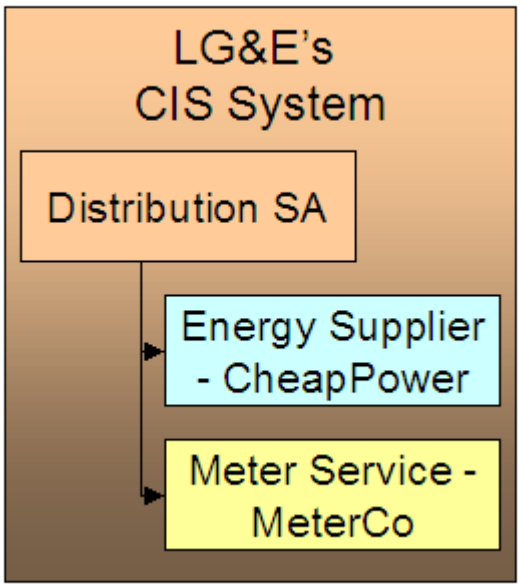
Service Providers Are Linked To Service Agreements

The following diagram illustrates a customer's bill for electric service in a deregulated market. Notice that there are separate sections for energy, distribution and meter service.

> **Note**
>
> **Consolidated billing.** The following is an example of a bill that consolidates charges from many service providers. Rather than receive a consolidated bill, it is possible for the customer to receive 3 separate bills, one from each service provider (we refer to this as Dual billing).

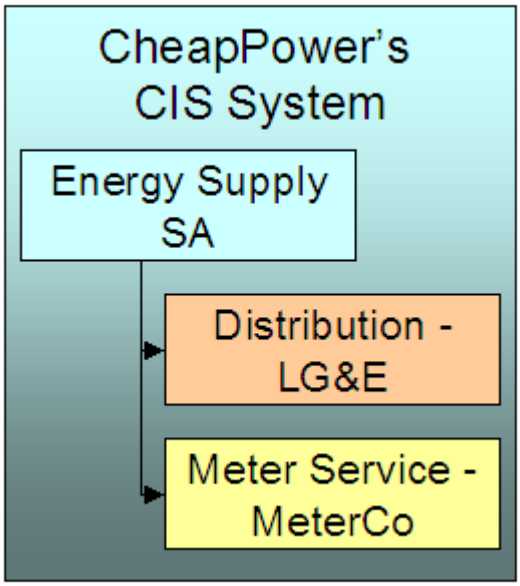


If we were to look at this customer's service agreement in the distribution company's CIS system, we'd find a service agreement for distribution charges, and linked to it would be information about the meter service and energy service providers:

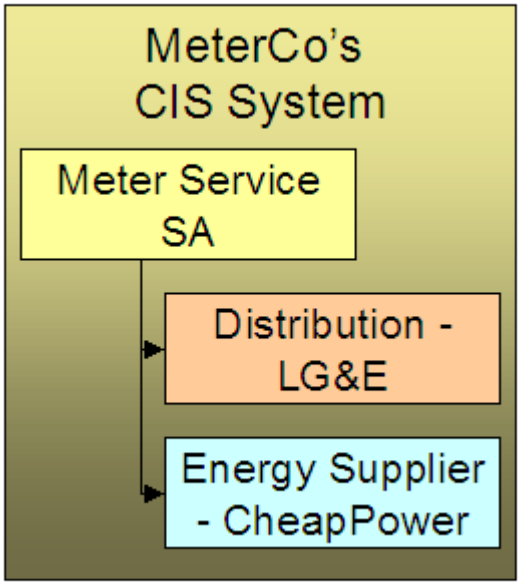


Copyright © 2026, Oracle and/or its affiliates.

If we look at this customer's service agreement in the energy supply company's CIS system, we'd find a service agreement for energy charges, and linked to it would be information about the distribution and meter service providers:



And finally, if we look at this customer's service agreement in the meter service company's CIS system, we'd find a service agreement for meter service charges, and linked to it would be information about the distribution and energy providers:



> **Note**
>
> **Bottom line.** A customer's service providers keep track of the customer and each other in their respective CIS systems. A customer will have a service agreement (or the equivalent) in each service provider's CIS system. A customer's service agreement defines the service providers who supply each type of service.

Copyright © 2026, Oracle and/or its affiliates.

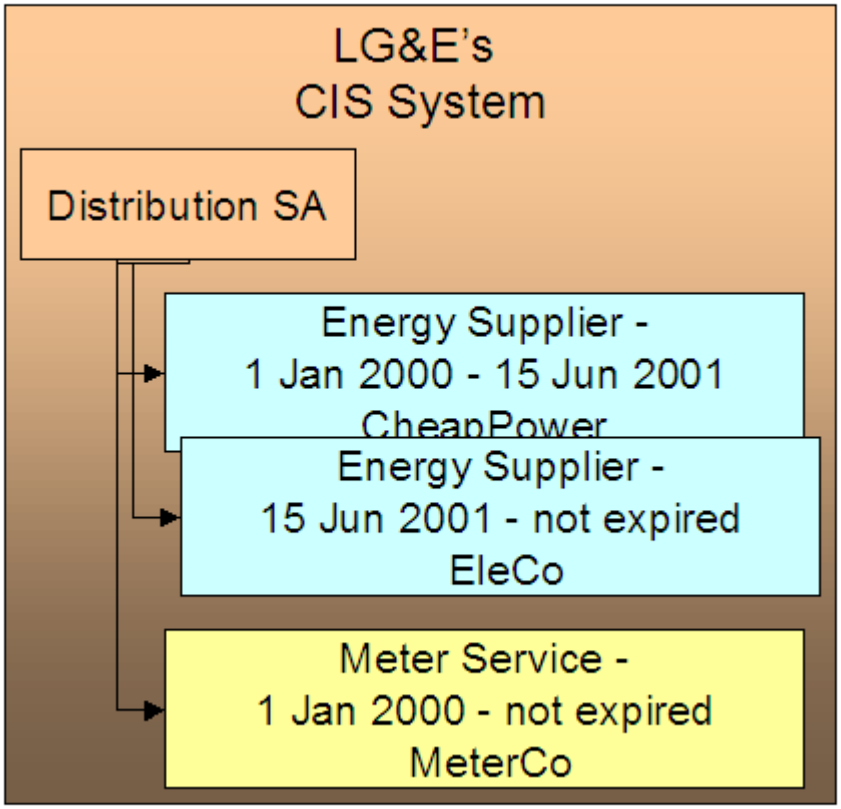
NOT_SUPPORTED

Because information about customers and their services needs to be kept up-to-date in many different CIS systems, there is a need for automated communications between service providers. Refer to Service Providers Have To Communicate About Customers for more information.

Service Providers May Change Over Time

A customer typically has a choice of service providers. Over time, their choice may change. The system keeps track of a customer's service providers throughout time so that it can accurately cancel / rebill historical bills. This means that the service provider relationship is *effective-dated*.

The following diagram illustrates how a distribution company's system keeps track of a customer's service providers (notice the customer changed energy suppliers):



How To Set Up SA Relationships On A Customer's Service Agreement

There are three ways to set up a customer's service providers:

- **Manually.** An operator can manually change a service agreement's SA relationships. The manual method is NOT recommended as changing service providers typically involves

Copyright © 2026, Oracle and/or its affiliates.

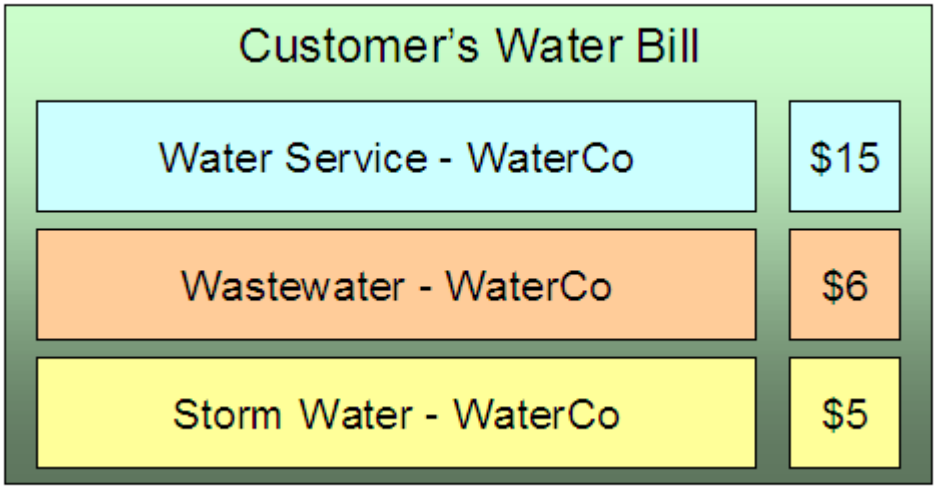
many events (e.g., you have to notify the current service provider that they will be dropped).

- **Use Default Service Providers.** The system will default a service provider on a required SA relationship when a service agreement is activated. Refer to Defaulting Relationship Types And Defaulting Service Providers for more information.

When Your Company Is A Service Provider

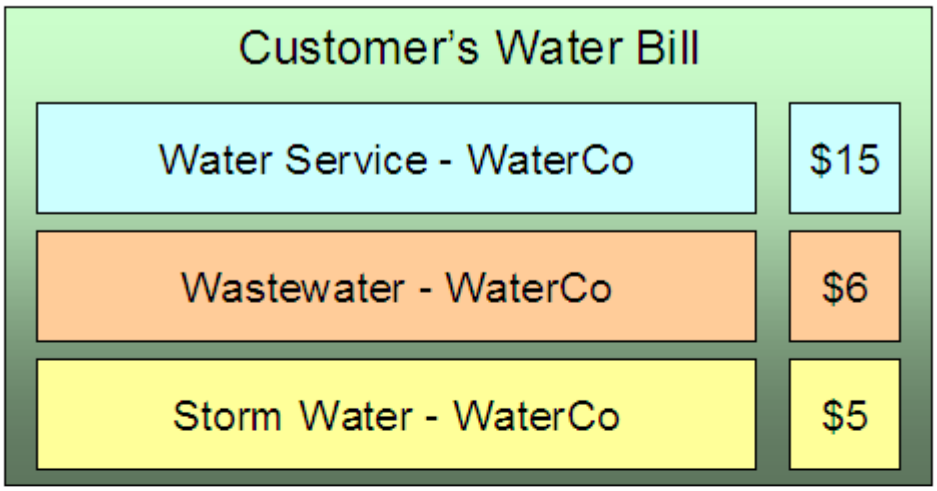
Besides setting up service providers for other companies that provide services to your customer, you may have to set up a service provider for your own company. You have to do this when:

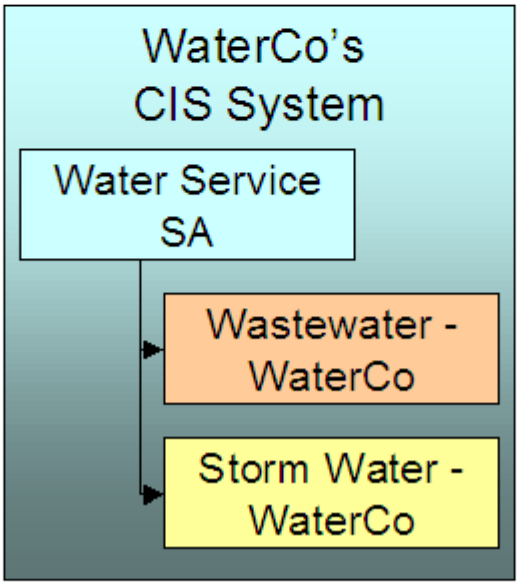
- **Your organization can supply one of the services.** Refer to the illustration in Service Providers Are Linked To Service Agreements. If you are LG&E and you supply energy in addition to distribution, you would need to set up a service provider for your own organization. Why? Because whenever you have a subcategory of service (e.g., energy supply), you must indicate the service provider who provides this service; even when it's you.
- **You decide to break up a service into subcategories** (and have a separate service agreement for each category). For example, a water company may choose to break up service charges into water, wastewater and storm water (they may do this because there are different rates for each category of service). The following is an example of the segregated charges associated with this water company's service charges.



This water company system may benefit by creating a single service (for water) and indicating there are subcategories of service (for wastewater and storm water). Whenever you have a subcategory of service, you must indicate the service provider who provides this service. And, in this example, the water company would be the sole service provider for each subcategory of service.

Copyright © 2026, Oracle and/or its affiliates.





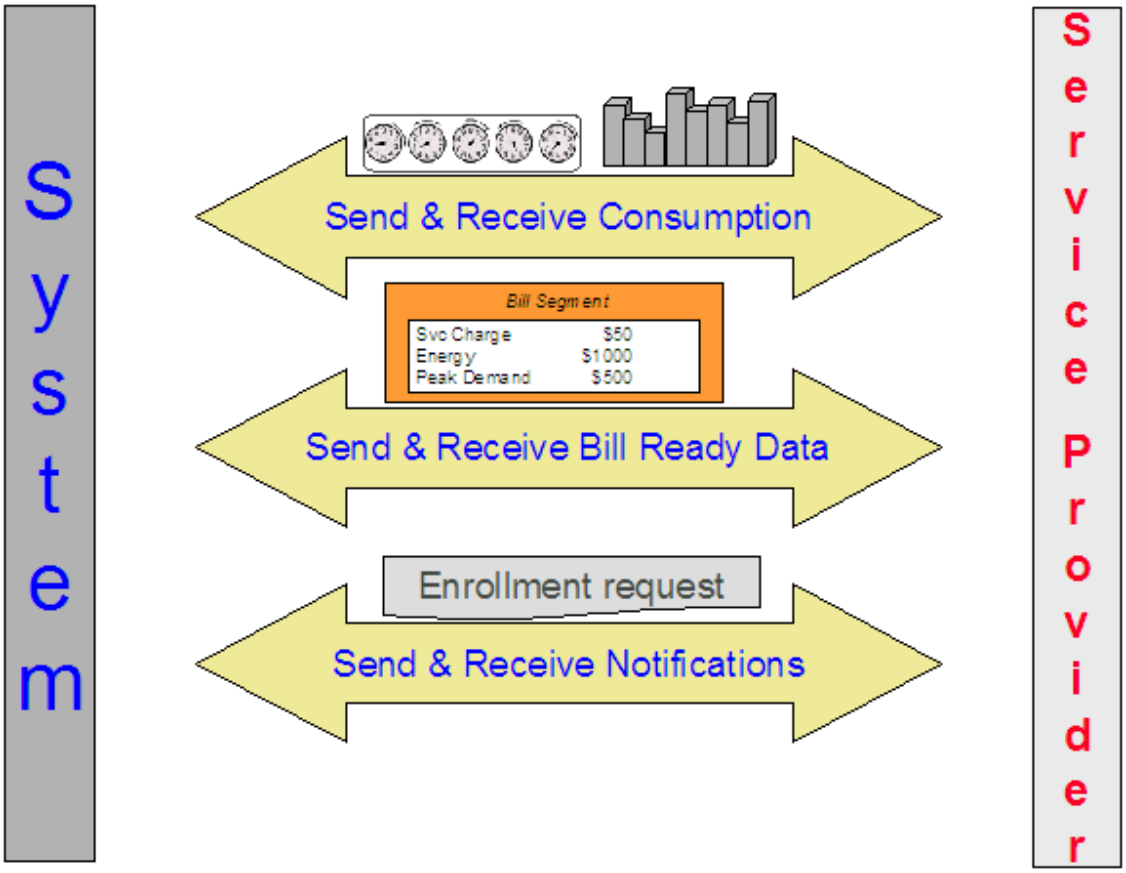
Note

Refer to We Bill For Them - Rate Ready Bill Segments Are Special for restrictions in respect of using subcategories of service.

Service Providers Have To Communicate About Customers

The providers of service typically have to communicate with each other in respect of the customer's service. The following diagram illustrates the major interfaces of information between your system and your service providers.

Copyright © 2026, Oracle and/or its affiliates.



Depending on where your organization fits in the service provider hierarchy, you may:

- Bill for other service providers (or they may bill for you). Refer to Billing Relationships for more information about billing communications.
- Send the customer's consumption to service providers (or they may send it to you). Refer to Consumption Relationships for more information about consumption communications.
- Apprise service providers of the changes to the customers' service (or they may apprise you). Refer to How Do You Communicate With Service Providers? for more information about communications between service providers.

Relationships Between Service Providers

Service providers may arrange contractual relationships with other service providers to provide additional services, e.g., an energy service provider may work with a specific meter data management agency to gather and report interval meter read data.

The system does nothing special to enforce or record these inter-relationships. Why? Because most service provider switches are received by notification records. Notification records indicate all the associated service providers. There is no need for the system to maintain the inter-service provider associations.

A Service Agreement Can Have Many Types Of Relationships

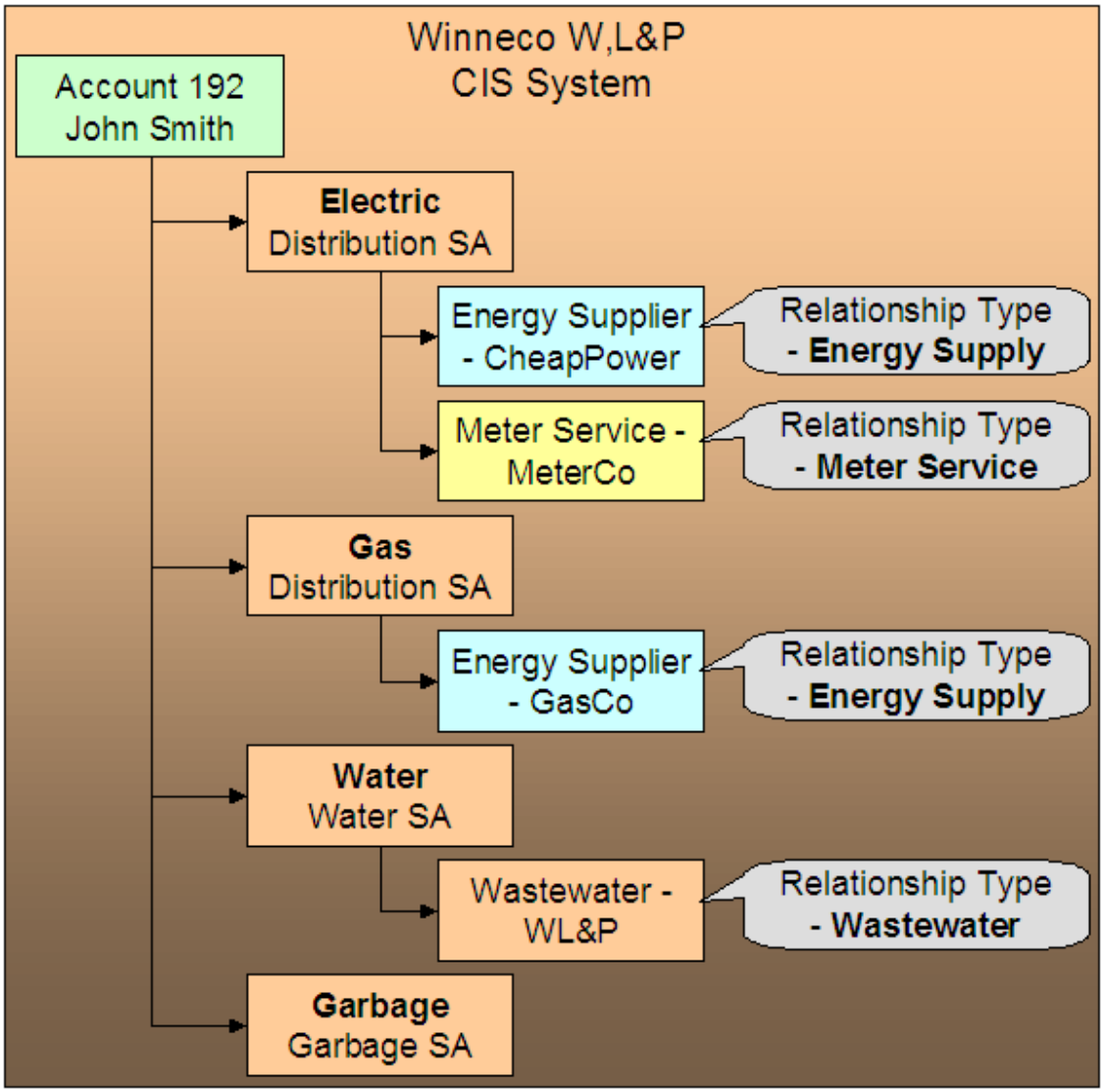
Copyright © 2026, Oracle and/or its affiliates.

As described in the previous sections, a given service can be subdivided into subcategories. Each type of service can have zero, one or more subcategories. We call each subcategory a **SA Relationship Type**. The topics in this section provide information about SA relationship types.

- An Example
- Valid Relationship Types and Service Providers Are Defined On SA Types
- Defaulting Relationship Types And Defaulting Service Providers
- Required Relationship Types and Billing
- Relationship Types Do Not Impact Start/Stop

An Example

We'll use the following example of a customer in a municipal utility's CIS system to explain SA Relationship types:



Note the following:

Copyright © 2026, Oracle and/or its affiliates.

- Electric service has 2 SA relationship types: energy supply and meter service.
- Gas service has 1 SA relationship type: energy supply.
- Water service has 1 SA relationship type: waste water.
- Garbage service has no SA relationship types.

> **Note**
>
> **Bottom line.** Service providers are related to the customer via "your" service agreement. Each service provider linked to a service agreement is defined in respect of a SA Relationship Type. This relationship is effective-dated because we care about how it changes over time.
>
> **Your relationship type is implied.** The SA relationship type of "your" service agreement is implied, e.g., if you are a distribution company, "your" service agreement's implied SA relationship type is "distribution".

Valid Relationship Types and Service Providers Are Defined On SA Types

You control which services have SA relationships (and which don't) when you set up your SA types. Each SA type can have zero, one or more SA relationship types. Each relationship type, in turn, can have one or more valid service providers.

> **NOT_SUPPORTED**
>
> Refer to Setting Up SA Relationships For SA Types for the page used to define the service providers and SA relationship types for each SA type.

Defaulting Relationship Types And Defaulting Service Providers

Please be aware of the following:

- A SA relationship type for a given SA type can be marked as being **Required**.
- A service provider for a given SA relationship type / SA type can be marked as being the **Default**.

If, at activation time, the customer's master SA is missing a **Required** SA relationship that has a **Default** service provider, the activation process automatically creates the SA relationship type and links to it the **Default** service provider. If a master service agreement doesn't have all **Required** SA relationships, the service agreement cannot be activated. This is handy when your organization is the default service provider for a relationship type.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Automatic Creation of Sub SAs for information about how the system will automatically create sub service agreements for the defaulted SA relationship / service provider if your organization provides billing services for the service provider. Refer to Setting Up SA Relationships For SA Types for the page used to define the service providers and SA relationship types for each SA type.

Required Relationship Types and Billing

When the system attempts to create a bill segment for a service agreement whose SA type has **Required** relationship types, it checks if all such relationships are defined for the service agreement. If not, a bill segment error will be generated.

The reason this restriction exists is to handle the situation when your required relationship types change over time. For example, assume on your first day of production you only need energy suppliers defined on electric service agreements. After several months, gas deregulates. When this happens, you will need to change your control tables to indicate that your gas SA types require an energy supplier. If you don't write a default program to update your existing gas service agreements, billing will complain.

Relationship Types Do Not Impact Start/Stop

Customer service representatives (CSR's) are typically not involved with the customer's choice of service providers. Most organizations hear about a customer's service providers from the service providers or from a central body. This means that the start/stop dialog is not impacted by SA relationships. This also means that CSR's are not impacted by SA relationships (unless something goes wrong). If something goes wrong, the CSR's may need to manually correct SA relationships. Refer to How To Set Up Service Providers On A Customer's Service Agreement for more information.

Billing Relationships

When you set up a service provider, you must define your organization's billing relationship with the service provider. The following points provide examples of the billing relationships supported in the system,

- If you are an energy supply company, you may provide billing services for the distribution company. This means that your bill contains both your charges and the distribution company's charges. We refer to this as the **We Bill For Them** billing relationship.
- Alternatively, the distribution company may provide billing services for you. This means that the distribution company's bill contains your charges and their charges. We refer to this as the **They Bill For Us** billing relationship.
- Alternatively, you may both send bills to the customer. We refer to this as the **Dual Billing** relationship.
- Alternatively, if you subcategorize your services OR if your company provides one of the services that is provided by your service providers, then the system will create a separate

Copyright © 2026, Oracle and/or its affiliates.

bill segment for the subcategory of service. We refer to this as the **It's Us** billing relationship.

If you provide billing services for another service provider (i.e., you bill for them), there are two ways to determine the service provider's charges:

- You can load the service provider's rates in your system and calculate the charges for the service provider. We call this the **Rate Ready** calculation method.
- You can let the service provider calculate their own charges and interface them to you. We call this the **Bill Ready** calculation method.

If a service provider provides billing services for you (i.e., they bill for you):

- If the service provider has a suitable CIS system, they can load your rates in their system and calculate your charges for you. We call this the **Rate Ready** calculation method.
- You can calculate your charges and interface them to them. We call this the **Bill Ready** calculation method.

If you don't have a billing relationship with a service provider, you still need a service provider record to define such. Why? Because the system needs to know that it doesn't have to worry about a particular service provider in respect of billing. In addition, you may have other interactions with a service provider that have nothing to do with billing, e.g., you may send or receive consumption.

The topics in this section provide a wealth of information about the various billing relationships and the ramifications of each.

- Sub Service Agreements
- We Bill For Them
- Service Providers Have Service Agreements Too
- They Bill For Us
- Combinations Of Service Provider Billing Methods
- Different Service Providers Can Bill Different Services
- If You Deal With TBFU Service Providers, You Cannot Reopen Bills

Sub Service Agreements

If you provide billing services for another service provider OR if you subcategorize your own charges, there will be a separate service agreement (SA) linked to the customer's account that holds these unique charges. We refer to this new service agreement as a **Sub SA**. We use the term "sub" because this service agreement is subservient to the "master" service agreement. By subservient we mean:

- A sub SA's start and stop dates are the same as the master SA. This statement may seem odd, but it's true – all sub SAs linked to a master service agreement have the same start and stop dates as the master.
- A sub SA's status (i.e., *pending start*, *active*, *pending stop*, *stopped*, etc.) is controlled by its master service agreement. As a rule, the master SA transitions its status first and the sub SA(s) follow.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> There are situations where the master service agreement transitions its status and the sub service agreement(s) does not. One example of this is when both the master and sub SAs are closed and a payment cancellation occurs that only affects the master SA. In this scenario, the master SA becomes reactivated and the sub SA remains closed. Another example is when a straggling billable charge posts to a sub SA; the master SA is not affected.

- Refer to The Rate Ready Calculation Method for additional examples of subservience.

> **Note**
>
> You may find it helpful to keep in mind that sub SAs are only used for service providers with a billing relationship of It's Us and We Bill For Them. This is because these are the only relationships that have implicit billing responsibilities.

The topics in this section provide additional information about sub SAs.

- Only Some Service Providers Have Sub SAs
- Automatic Creation of Sub SAs
- Sub SA State Transition

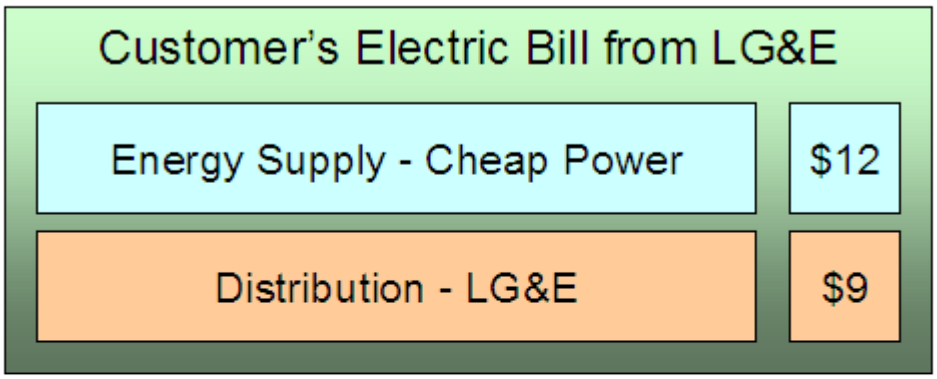
Only Some Service Providers Have Sub SAs

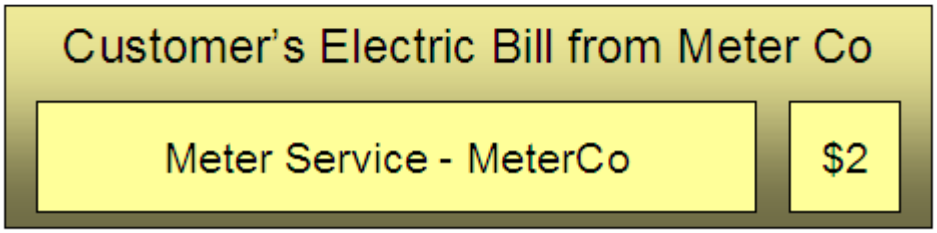
If you do not provide billing services for a service provider, there will be no sub SAs associated with the service provider's SA relationships. Let's use an example to make the point; assume:

- Service is divided into distribution, energy and meter service charges.
- You are the distribution company and you provide billing services for the energy supply company.
- The meter service company bills the customer independently.

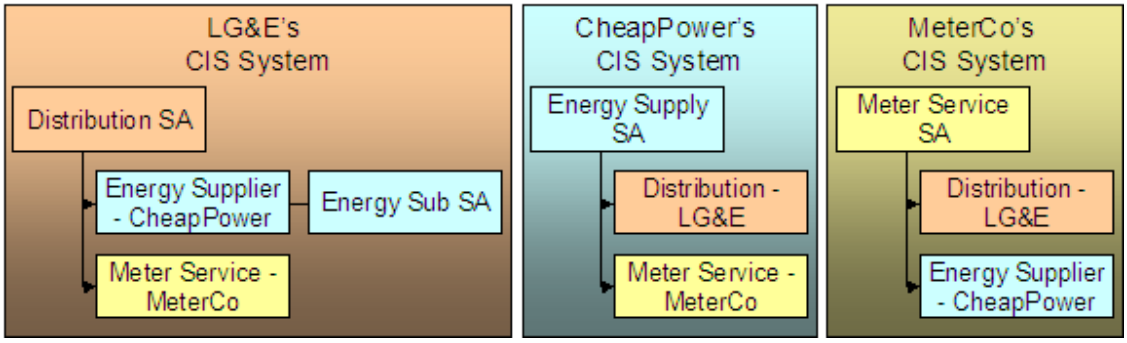
In this situation, the customer will receive two bills: one from you (LG&E), the other from MeterCo. Notice that your bill (LG&E) contains your distribution charges AND CheapPower's energy charges:

Copyright © 2026, Oracle and/or its affiliates.





If we were to look at the customer information in the three service providers' respective CIS systems, we'd find the following service agreements and SA relationships for the customer:



Notice the following:

- The customer has a service agreement in each supplier's CIS system.
- Because LG&E (the distribution company) bills for CheapPower, all customers who have their energy supplied by CheapPower will have a sub SA in LG&E's system. This sub SA maintains the charges (and receivable balance) associated with CheapPower's energy charges.
- Notice that neither CheapPower's nor MeterCo's CIS systems use sub SAs. This is because neither company bills for other service providers.

> **Note**
>
> **Sub SAs are needed if you subcategorize your charges.** The above example shows sub SAs being used when a company provides billing service for another company. Sub SAs are also used when you subcategorize your charges - each sub SA contains the rate associated with each subcategory.

Copyright © 2026, Oracle and/or its affiliates.

Automatic Creation of Sub SAs

The system creates sub SAs for customers choosing service providers where the billing option is It's Us or We Bill For Them. The system creates sub SAs via the following mechanisms:

- The analyze SA relationships background process (ANLYZSAR) monitors newly activated SA relationships. If the respective service provider is It's Us or We Bill For Them, this process creates the sub SA(s) using the information defined on SA Type SA Relationship Type. Refer to Setting Up SA Relationships For SA Types for more details.

> **Note**
>
> The analyze SA relationships background process also activates and stops sub SAs. Refer to Sub SA State Transition for the details.

- A button exists on the SA Relationships page. This button, when pressed, creates sub SAs real-time. This button would only be used if the operator couldn't wait for the background process to run.

The SA type associated with the new sub SAs is defined when you set up each SA Type / SA Relationship Type. Refer to Setting Up SA Relationships For SA Types for more information.

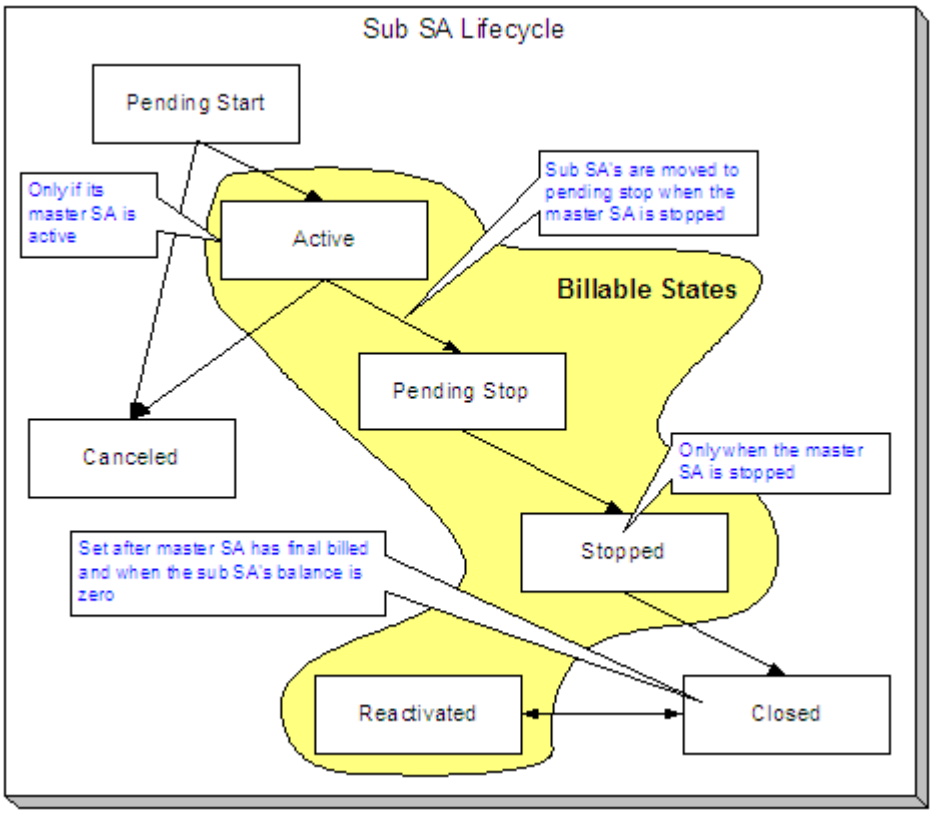
If the SA type uses start options, the start option contains additional values (e.g., rates, contract riders) that are used to populate the newly created sub SAs. Note: start options are typically not used for We Bill For Them - Bill Ready service provider because we don't need to default rates, contract riders and/or contract values on billable charge sub SAs. Refer to Setting Up SA Relationships For SA Types for more information.

> **Note**
>
> **Manually created sub SAs.** In very unusual situations, an operator may create a sub SA manually. An operator would do this using the SA Relationship transaction.

Sub SA State Transition

Sub SAs follow normal SA state transition rules, but have a few additional rules about when the transitions can take place.

Copyright © 2026, Oracle and/or its affiliates.



The following points highlight the additional state transition rules:

- A sub SA can only become Active if its master SA is Active. Most sub SAs are activated by the analyze SA relationships background process. However, a user can manually activate a sub SA using the activate button on the SA maintenance page.
- Sub SAs become Pending Stop when their master becomes Stopped . This typically occurs when the SA activation background process stops the master SA. However, this can happen real time if a user manually stops a master SA using the stop button on the SA maintenance page.
- A sub SA can only become Stopped if its master SA is Stopped . Most sub SAs are stopped by the analyze SA relationships background process. However, a user can manually activate a sub SA using the stop button on the SA maintenance page.
- A sub SA becomes Closed if its master has been final billed and it has a balance of zero. Most sub SAs are closed when their balance becomes zero after the master SA is final billed. However, a user can manually close a sub SA using the close button on the SA maintenance page. Note, if additional billable charges are interfaced after the master has been closed, the sub SA will be Reactivated.

We Bill For Them

If you provide billing services for another service provider, then you have a We Bill For Them billing relationship with the service provider. The topics in this section provide information about this type of billing relationship.

- Sub SAs Are Used When We Bill For Them
- The Rate Ready Calculation Method

Copyright © 2026, Oracle and/or its affiliates.

- The Bill Ready Calculation Method
- Pay at Bill Time Versus Pay at Pay Time
- Paying The Service Provider - Technical Implementation

Sub SAs Are Used When We Bill For Them

As described under Sub Service Agreements, sub SAs are used for We Bill For Them service providers. The sub SAs hold the service providers charges. It's important to note that the system allows more than one sub SA to be created for a given service provider.

The Rate Ready Calculation Method

If you provide billing services for a service provider (i.e., the service provider has a billing relationship of We Bill For Them), you can load the service provider's rates in your system and calculate the charges for the service provider. We call this the **Rate Ready** calculation method.

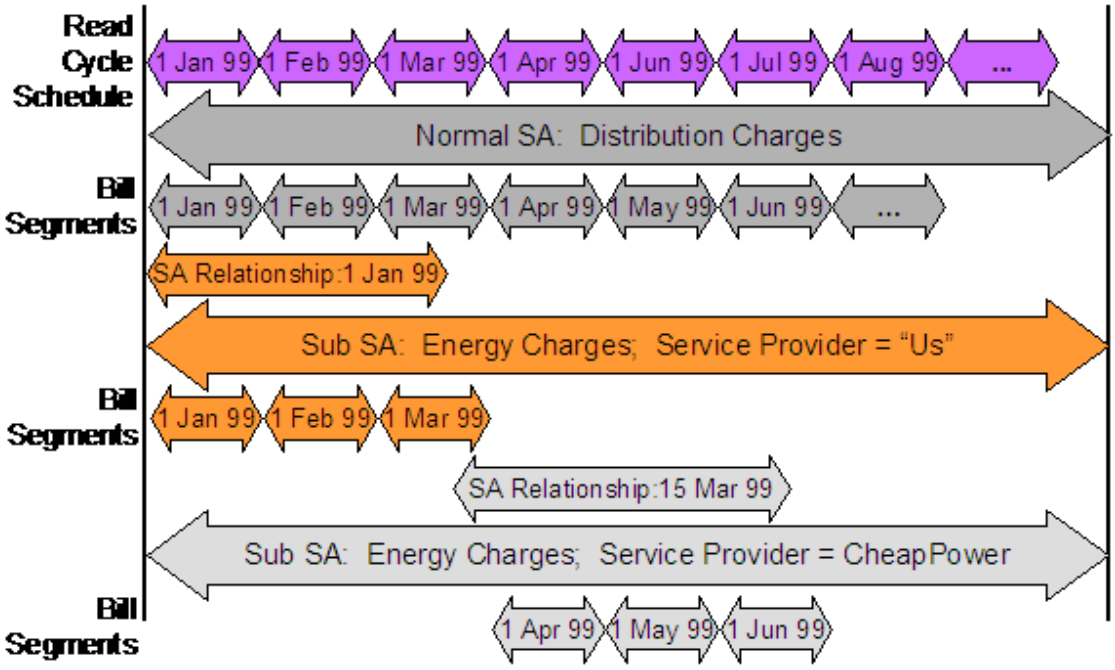
> **Note**
>
> It's Us **can be Rate Ready.** The prior paragraph indicated that the Rate Ready billing option was used for We Bill For Them service providers. If you subcategorize your services and use rates to calculate each subcategory's charges, the service provider(s) set up for your own organization will also be subject to the Rate Ready rules described below.

If a customer uses a We Bill For Them - Rate Ready service provider, the system will create a separate bill segment for the service provider's charges. Keep in mind the following in respect of these types of bill segments:

- The consumption is copied from the "master" service agreement (note: we copy item details and read details after pre-processing calculation groups, if any exist, have been applied). This means that the bill period on the sub SA's bill segment is identical to that on the normal SA. This also means that if the "master" SA is in error, bill segments for the sub SA will not be created.
- The rate defined on the sub SA is applied against this consumption. All pre-processing calculation groups defined on the service provider's rate will be used to manipulate this consumption. If you have pre-processing calculation group on your master SA's rate to apply a line loss or convert a cubic foot to a therm, you don't have to have these pre-processing calculation group on the sub SA's rate (because we copy the consumption from the master to the sub after pre-processing calculation group, if any exist, have been applied).

The following illustration should help:

Copyright © 2026, Oracle and/or its affiliates.



Be aware of the following:

- This example is from a distribution company's perspective.
- From 1 Jan 99 through 15 Mar 99, the distribution company distributes power AND supplies energy (i.e., the distribution company is the service provider of energy).
- On 15 Mar 99, CheapPower becomes the energy supplier. However, the next bill period ends on 30 Mar 99. This means that CheapPower will only start supplying energy on 1 Apr 99. See the following note for why the effective date of the supplier switch is not the true effective date from billing's perspective.

> **Note**
>
> **Service Provider Changes Take Effect On The Next Bill.** If the service provider is changed within a customer's billing period, the system assumes that the service provider in effect at the START of the period is effective the entire billing period. This means that a change of service providers will only take effect on the bill whose start date follows the change date.

Please be aware of the following characteristics of Rate Ready bill segments:

- You cannot cancel a Rate Ready bill segment independently from the master bill segment. If you need to cancel / rebill the sub SA's bill segment, you must cancel / rebill the master SA's bill segment.
- You cannot delete a Rate Ready bill segment independently from the master bill segment. If you need to delete the sub SA's bill segment, you must delete the master SA's bill segment.
- Similarly, if you cancel / rebill the normal SA's bill segment, all Rate Ready bill segments will be cancelled / rebilled.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bottom line.** Creating, freezing, deleting, and canceling a "master" bill segment does the same with all Rate Ready bill segments. You cannot create, freeze, delete or cancel a Rate Ready bill segment independent of its master bill segment.

The Bill Ready Calculation Method

If a customer uses a We Bill For Them - Bill Ready service provider, the service provider calculates their own charges and interfaces them to the system. We then present their charges on our bill; we don't actually calculate anything (sometimes this is referred to as "pass through" billing).

The sub SA linked to a We Bill For Them - Bill Ready must be a billable charge SA. Why? Because billable charge SAs exist to hold bill lines until such time as a bill is created for the customer's account. Refer to Setting Up Billable Charge Templates for more information.

Please be aware of the following characteristics of Bill Ready bill segments:

- Unlike Rate Ready bill segments, Bill Ready bill segments can span different time periods than the master SA. This is because you cannot predict when a service provider will interface their billable charges to you. In fact, a given bill could contain billable charges that span different periods and these charges could have been interfaced from historic and existing service providers.
- Unlike Rate Ready bill segments, Bill Ready bill segments can be created and deleted independently from the master bill segment.
- Unlike Rate Ready bill segments, Bill Ready bill segments can be cancelled independently from the master bill segment.

Sending Consumption And Waiting For The Charges

Uploading Consumption (Rather Than Uploading Calculated Charges)

Calculating Taxes On Uploaded Charges

Sending Consumption And Waiting For The Charges

If your organization supports We Bill For Them - Bill Ready service providers AND you are the source of consumption used by these service provider to calculate their charges, you need to be aware of the following:

- We do NOT recommend sending raw meter reads to service providers. Rather, we recommend sending these service providers the same consumption that you use on your bill segments. Remember, the system maintains a snapshot of billed consumption on bill segments associated with service agreements that are linked to service points.
- But to implement our recommendation (of only interfacing billed consumption to service providers), we need to create a bill segment for the master SA and then wait until the service provider returns the billable charge before sending out the bill. The following points describe how this works:
  - Early in the bill cycle, the system creates a bill segment for the master SA (remember, the system maintains a snapshot of billed consumption on bill segments linked to service points). When the master bill segment is frozen, the system interfaces the snapshot consumption to all service providers associated with the master service

Copyright © 2026, Oracle and/or its affiliates.

agreement who need consumption (this is defined on the service provider). Refer to We Can Send Billed Consumption To Any Service Provider for more information about interfacing consumption to service providers.

- The bill associated with the bill segment will not be completed. Why? Because the bill segments associated with the We Bill For Them - Bill Ready service providers will be in Error. This only happens if you use the Billable Charge bill segment creation algorithm (this algorithm is plugged in on the sub SA's SA type). On the algorithm, make sure to specify a value of Y for the parameter **Wait For The Last Day Of The Bill Cycle**. By doing this, a bill segment in the Error state will exist for the sub SA until the last night of the bill cycle's window.
- On the last night of the bill cycle window, when the system attempts to create a bill segment for the billable charge SA associated with the We Bill For Them - Bill Ready, it will either find recently interfaced billable charges or it won't. If it finds unbilled, billable charges, a bill segment will be created for the sub SA and the billable charges will be swept onto it. If it doesn't find unbilled billable charges, the bill will be completed (i.e., sent out) without the service provider's charges.

> **Note**
>
> **Bottom line.** If a customer's account uses We Bill For Them - Bill Ready service providers, the bill will not be completed (i.e., sent out) until the last night of the bill cycle (if you use the appropriate algorithm). Why? Because we wait until the last night of the bill cycle before trying to sweep on recently interfaced billable charges. If no billable charges have been interfaced from the service provider by the last night of the bill cycle, the bill will be sent out without the service provider's charges.
>
> **Batch versus Online Bill Creation.** If you create a bill online, the system will NOT create an Error bill segment for the We Bill For Them - Bill Ready service provider. Why? Because if you want to create an online bill, either the service provider has interfaced their charges or they haven't. If they have, they should be swept on the bill (via the creation of a bill segment). If they haven't, it shouldn't prevent you from completing the bill.

Top of the Page

Uploading Consumption (Rather Than Uploading Calculated Charges)

We understand this is confusing, but it is quite possible to set up the system so that the We Bill For Them - Bill Ready service provider passes in CONSUMPTION rather than the calculated bill lines. They would only do this if they are not able to calculate the charges in their system and have therefore provided you with their rates. To do this, you would set up everything as described above. In addition, when you upload the billable charges, you must specify the consumption to be rated in the billable charge's service quantity (SQ) collection.

Top of the Page

Calculating Taxes On Uploaded Charges

It is possible to set up the system to calculate taxes for billable charges. You would do this if the service providers are passing through the charges and want you to calculate the taxes.

If you want taxes calculated on top of billable charges:

- Create a service quantity (SQ) on the billable charge that contains the total monetary amount that taxes will be calculated on. Note, you would not have to do this if you have a

Copyright © 2026, Oracle and/or its affiliates.

pre-processing calculation group in your rate that calculates the total monetary amount to which taxes should be applied.

- Specify a rate on the billable charge service agreement. This rate will contain calculation rules that calculate taxes. Note, the calculation rules will be simple SQ calculation rules that apply a percentage to the value of the SQ that represents the monetary amount on which taxes should be calculated.

When the system creates the bill segment for the billable charge, it will call the rate and the rate will calculate the taxes and add additional lines (actually, bill calculation headers) to the bill segment.

Pay at Bill Time Versus Pay at Pay Time

If you provide billing services for another service provider (i.e., the service provider has a billing relationship of We Bill For Them), you will owe them money because you will be receiving money from their customers for their service. You have two options in respect of when the system increases the amount you owe the service provider:

- You can tell the system to increase how much you owe the service provider when you freeze the customer's bill segment. Some people refer to this method as "purchasing the receivable from the service provider". We call this the **Pay At Bill Time** method.
- You can tell the system to increase how much you owe the service provider only when you are paid for by the customer. We call this the **Pay At Pay Time** method.

The method used for a service provider is defined on the service provider's record.

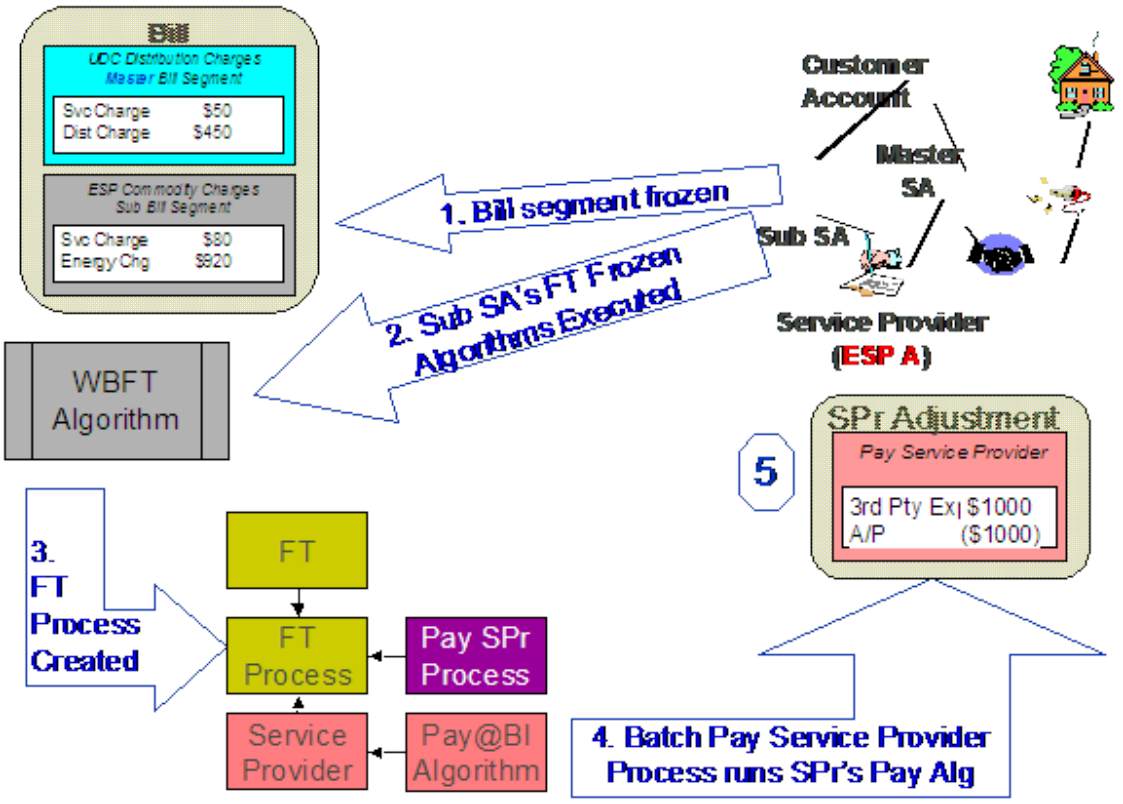
> **NOT_SUPPORTED**
>
> Refer to When We Bill For Them, We Owe Them Money for more information.

Paying The Service Provider - Technical Implementation

> **Warning**
>
> This section describes, technically, how we increase the amount we owe a We Bill For Them service provider. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with increasing how much we owe a We Bill For Them service provider.

Copyright © 2026, Oracle and/or its affiliates.



The following points explain the steps:

- When a financial transaction (FT) is frozen, the system executes the FT Freeze algorithms defined on the SA(s) SA type.
- If you've set up the system properly (i.e., you've put the appropriate FT Freeze algorithm on the sub SA's SA type), one of these algorithms will determine if there is a WBFT service provider associated with the sub SA. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that looks at new FT's and determines if a related "payable" adjustment should be created for the We Bill For Them service provider. This batch process uses the service provider's Payment Relationship and Pay Service Provider algorithm to determine when and how to create these "payable" adjustments.

Service Providers Have Service Agreements Too

Most service providers need service agreements as explained in the following topics.

- When We Bill For Them, We Owe Them Money
- Adjustments and We Bill For Them Service Providers
- When They Bill For Us, They Owe Us Money

When We Bill For Them, We Owe Them Money

When you bill on behalf of a service provider (i.e., the service provider has a billing relationship of We Bill For Them), you will eventually owe them money (because the customer pays you

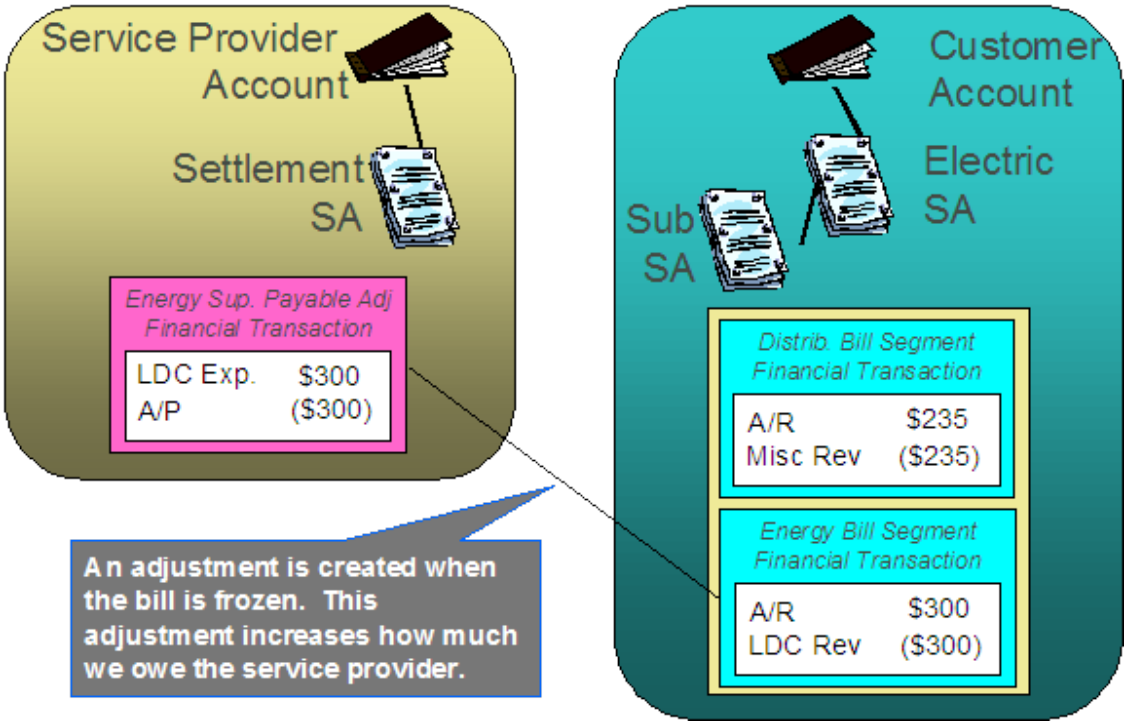
Copyright © 2026, Oracle and/or its affiliates.

for the service provider's service). You have two options in respect of when the system increases the amount you owe the service provider:

- You can tell the system to increase how much you owe the service provider when you create the customer's bill. Some people refer to this method as "purchasing the receivable from the service provider".
- You can tell the system to increase how much you owe the service provider only when you are paid for by the customer.

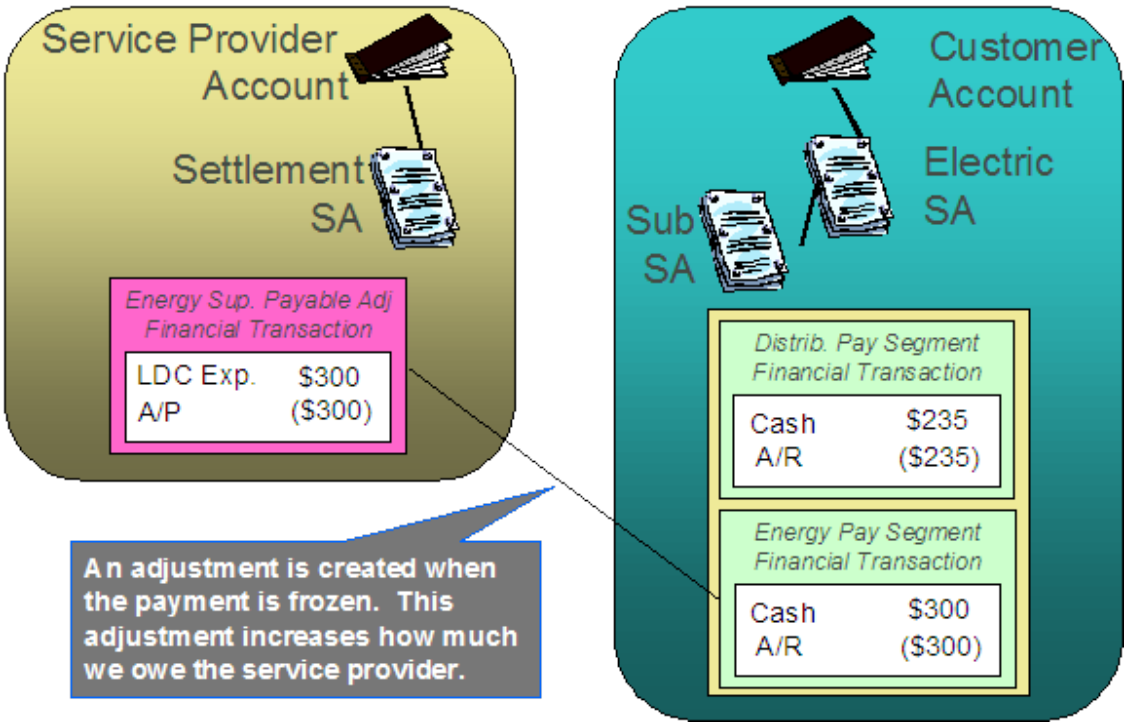
The system keeps track of how much you owe a service provider on a service agreement linked to the service provider's account. The system creates adjustments against this service agreement to increase how much you owe them.

If you "purchase the receivable" (i.e., you owe them when you bill the customer), an adjustment is created when the customer is billed. If you owe them only when you are paid by the customer, an adjustment is created when the customer pays. The following example illustrates an adjustment being created when the bill is frozen (illustrating the "purchase the receivable" scenario):



If you only pay the service provider when you are paid, the example would look as follows:

Copyright © 2026, Oracle and/or its affiliates.



Adjustments and We Bill For Them Service Providers

NOT_SUPPORTED

It's important that you are comfortable with the information described under When We Bill For Them, We Owe Them Money before reading this section.

Adjustments associated with We Bill For Them sub service agreements are tricky. The following points describe how the system "pays" the related service provider when adjustments are issued against the customer's sub service agreement:

- For Pay At Bill Time service providers, most adjustments are treated just like bill segments, i.e., when the adjustment's FT is frozen, a payable adjustment is created for the respective service provider. The reason "most" is underlined in the previous sentence is because A/P adjustments (i.e., adjustments used to interface check requests to your A/P system) are excluded. Why? Because A/P adjustments are used to refund overpayments to the customer. Overpayments are purely between the customer and your company (you never transferred the overpayment to the service provider because it's associated with a Pay At Bill Time service provider).
- For Pay At Pay Time service providers, A/P adjustments are treated just like payment segments, i.e., when the adjustment's FT is frozen, a payable adjustment is created for the respective service provider. All other types of adjustments are ignored. Why? Because A/P adjustments are used to refund overpayments to the customer. Think of it like this - when the customer originally overpaid, you transferred this overpayment to the Pay At Pay Time service provider; therefore, when you refund the overpayment, you get to take the money back from the service provider.

Copyright © 2026, Oracle and/or its affiliates.

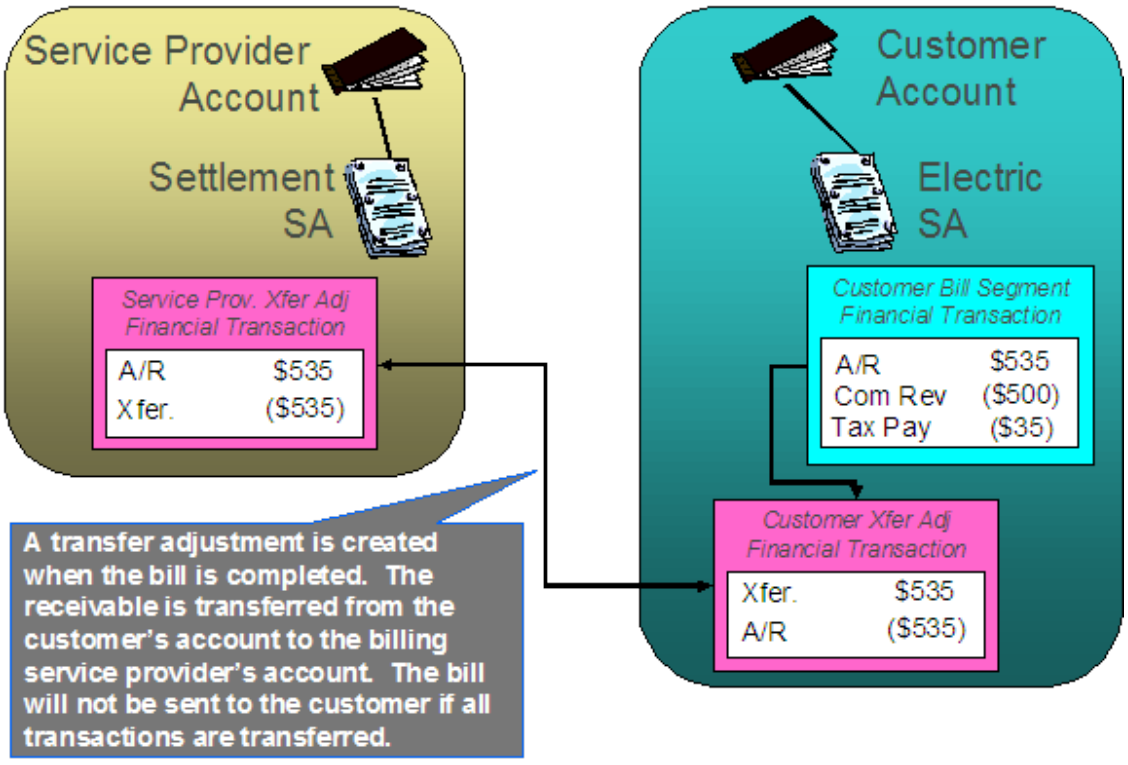
When They Bill For Us, They Owe Us Money

When a service provider bills on behalf of your organization, they will eventually owe you money (because the customer pays them for your service).

The system keeps track of how much a service provider owes you on a service agreement linked to the service provider's account. The question is, How does the system determine how much you are owed when you don't produce a bill? Well, you do produce a bill, it just doesn't get sent to the customer. We understand this is a little confusing, but think about it like this:

- The service provider is presenting your charges on their bill.
- You still have to calculate how much the customer owes your organization otherwise you'll never know how much you are owed by the service provider.

As illustrated below, when a bill is completed, the system determines if there are bill segments and/or adjustments associated with service agreements with a service provider who "bills for us". If it finds these, it transfers the receivable from the customer's service agreement to the service provider's service agreement. If all financial transactions have been transferred to the service provider, no bill is produced for the customer.



> **Note**
>
> **Bottom line.** We always generate a bill for "us", even though we don't send it to the customer.

Copyright © 2026, Oracle and/or its affiliates.

They Bill For Us

If a service provider provides billing services for you, then you have a They Bill For Us billing relationship with the service provider. The topics in this section provide information about this type of billing relationship.

- The Customer Still Needs A Service Agreement
- They Bill For Us - Bill Ready
- They Bill For Us - Rate Ready
- A/R Transfer - Technical Implementation
- Routing Billable Charges To Service Providers - Technical Implementation
- Bill Routings Are Changed

The Customer Still Needs A Service Agreement

If a service provider bills for you, you still need a service agreement for the customer. Why? Because:

- As explained under Service Providers Are Linked To Service Agreements, service providers are defined in respect of a customer's service agreement (therefore the customer must have a service agreement).
- As explained under When They Bill For Us, They Owe Us Money, you still have to calculate bills for the customer.

> **Note**
>
> **A customer's bill history still exists.** Be aware that even when a service provider bills for us, you will still be able to see the customer's billing history. It's just that the customer won't owe you anything because the receivable balance will be transferred to the service provider's account.

They Bill For Us - Bill Ready

At bill completion time, the system determines if there are bill segments and/or adjustments associated with service agreements with a service provider who "bills for us". If it finds these,

- It transfers the receivable from the customer's service agreement to the service provider's service agreement. If all financial transactions have been transferred to the service provider, no bill is produced for the customer.
- Note that payments and A/P adjustments are not transferred. Why? Because payments and A/P adjustments are purely between the customer and your company.
- Each bill segment and adjustment is marked to be interfaced to the service provider (via a separate background process).

Copyright © 2026, Oracle and/or its affiliates.

They Bill For Us - Rate Ready

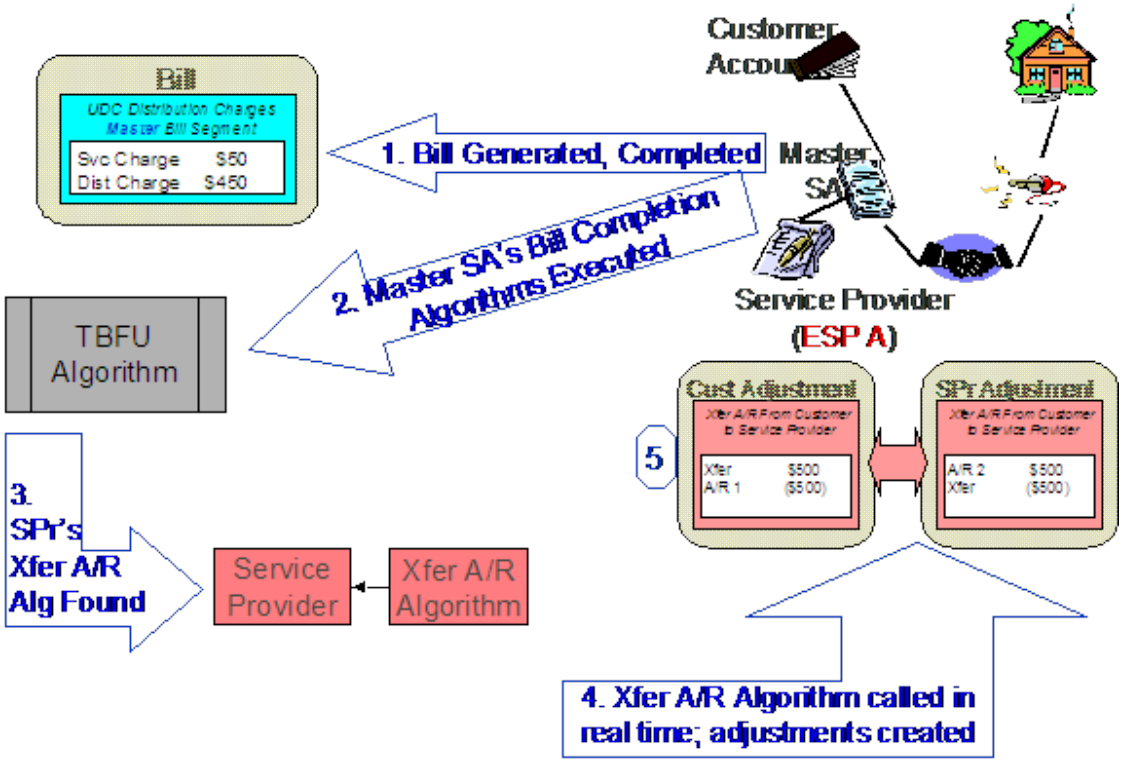
The They Bill For Us - Rate Ready option is not a recommended option. Why? Because you really have to compute how much the customer owes as explained above. If you go to the trouble of figuring out how much the customer owes, then it makes sense to interface this to the billing service provider.

The only difference between this option and They Bill For Us - Bill Ready is that the system will not interface the bill segments and adjustments to the billing service provider.

A/R Transfer - Technical Implementation

> **Warning**
>
> This section describes, technically, how a customer's A/R balance is transferred to a They Bill For Us service provider. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with the transference of a customer's A/R balance to a They Bill For Us service provider.



The following points explain the steps:

- When a bill is completed, the system executes the bill completion algorithms defined on the bill's master SA(s) SA types.

Copyright © 2026, Oracle and/or its affiliates.

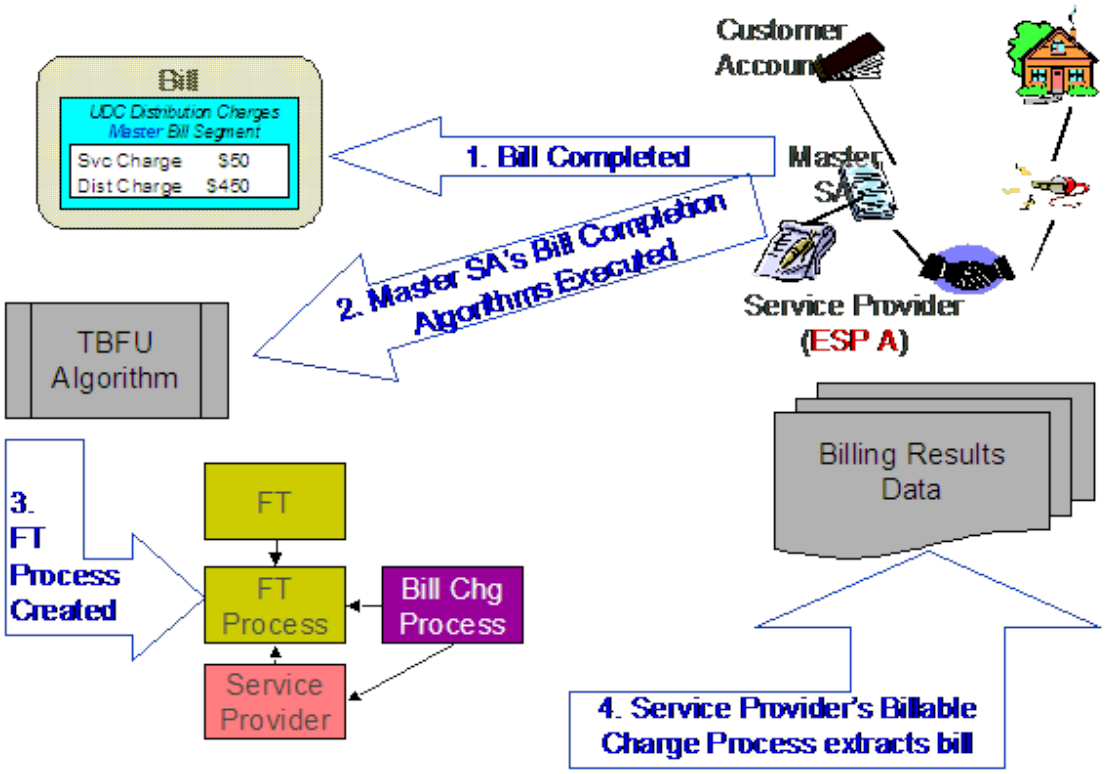
- If you've set up the system properly (i.e., you've put the appropriate Bill Completion algorithm on the master SA's SA type), one of these algorithms will determine if there is a They Bill For Us service provider associated with each master SA on the bill. If so, it will execute the Transfer A/R algorithm defined on the service provider's record. This algorithm causes a transfer adjustment to be created (transferring the financial transaction's affect on the customer's balance from the customer to the service provider).

> **Note**
>
> If there are multiple master SAs on a bill, the financial details associated with each respective master SA could be transferred to different service providers (e.g., one service provider could receive the financial details for gas and another for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Routing Billable Charges To Service Providers - Technical Implementation

> **Warning**
>
> This section describes, technically, how we send billable charges to service providers. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with sending billable charges to service providers.



Copyright © 2026, Oracle and/or its affiliates.

The following points explain the steps:

- When a bill is completed, the system executes the bill completion algorithms defined on the bill's master SA(s) SA types.
- If you've set up the system properly (i.e., you've put the appropriate Bill Completion algorithm on the master SA's SA type), one of these algorithms will determine if there is a They Bill For Us service provider associated with each master SA on the bill. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that downloads billable charges to the service provider. The ID of the batch process that is referenced on the trigger comes from the Service Provider's Billable Charge Download Process.

> **Note**
>
> If there are multiple master SAs on a bill, the financial details associated with each respective master SA could be routed to different service providers (e.g., one service provider could receive the financial details for gas and another for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Bill Routings Are Changed

If all of an account's "master" SAs have a SA relationship with a service provider who bills for us (i.e., the service provider's billing relationship is They Bill For Us), then we have nothing to send to the customer. The system still creates bill routings, but with a couple of differences:

- The Batch Process Id and Run Number are reset.
- The Customer's Name is prefixed with the text from a bill message code 6, 10103. This message code's text is *** Bill not sent.

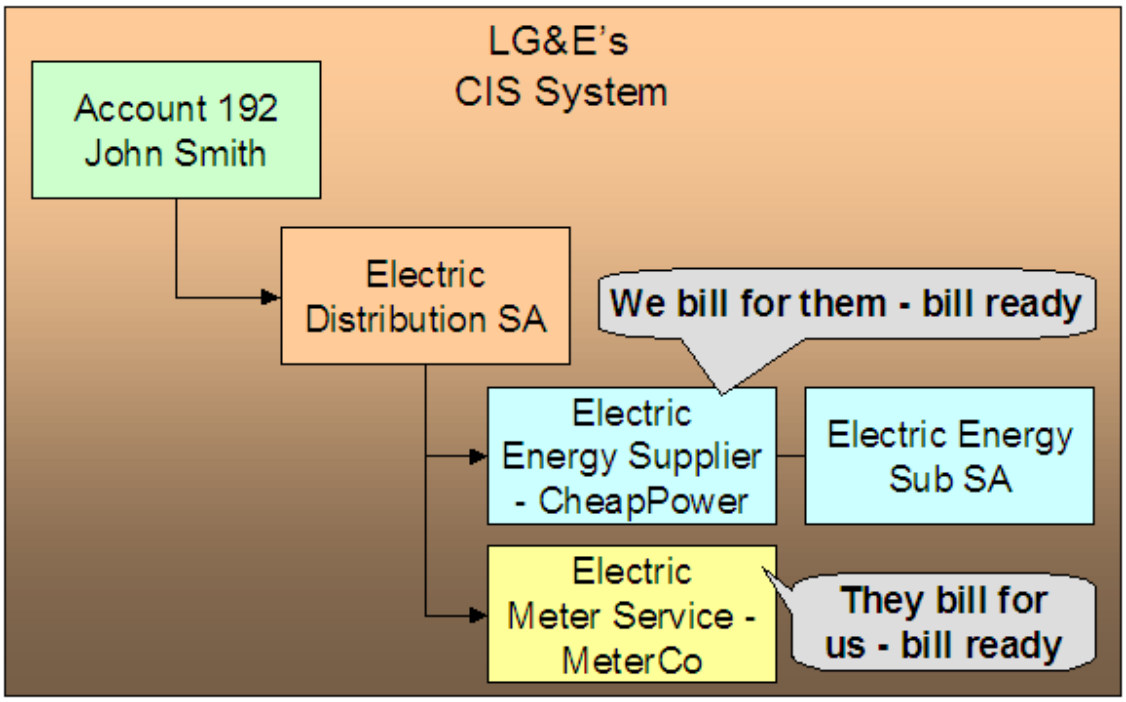
This way, the operators can easily see that that the bill was not routed and why.

> **NOT_SUPPORTED**
>
> Refer to Different Service Providers Can Bill Different Services for information about how the various services under an account could be billed by different service providers.

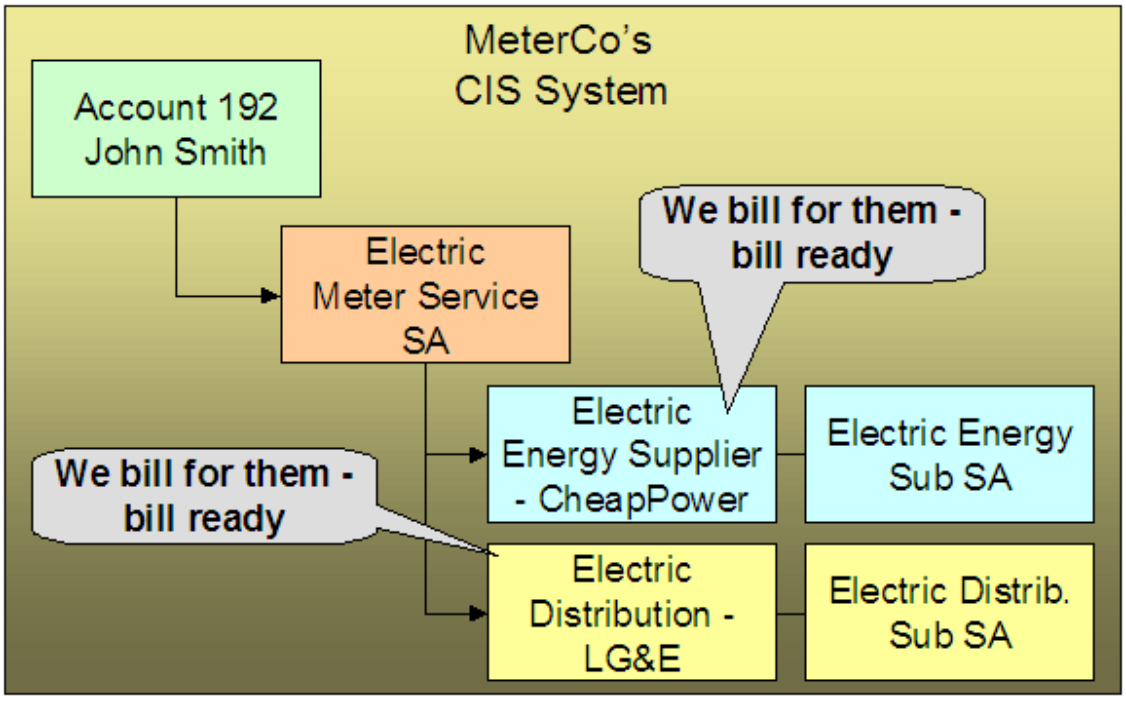
Combinations Of Service Provider Billing Methods

Consider the following situation:

Copyright © 2026, Oracle and/or its affiliates.



In the above example, we are billing for the energy service provider and the meter service provider is billing for us. This means that we will calculate the charges for ourselves, interface charges from CheapPower, and then interface our charges and CheapPower's charges to MeterCo. MeterCo will then produce a bill for the customer that contains our distribution charges, CheapPower's energy charges, and MeterCo's service charges. To help solidify this point, let's look at how this customer would look in MeterCo's CIS system.



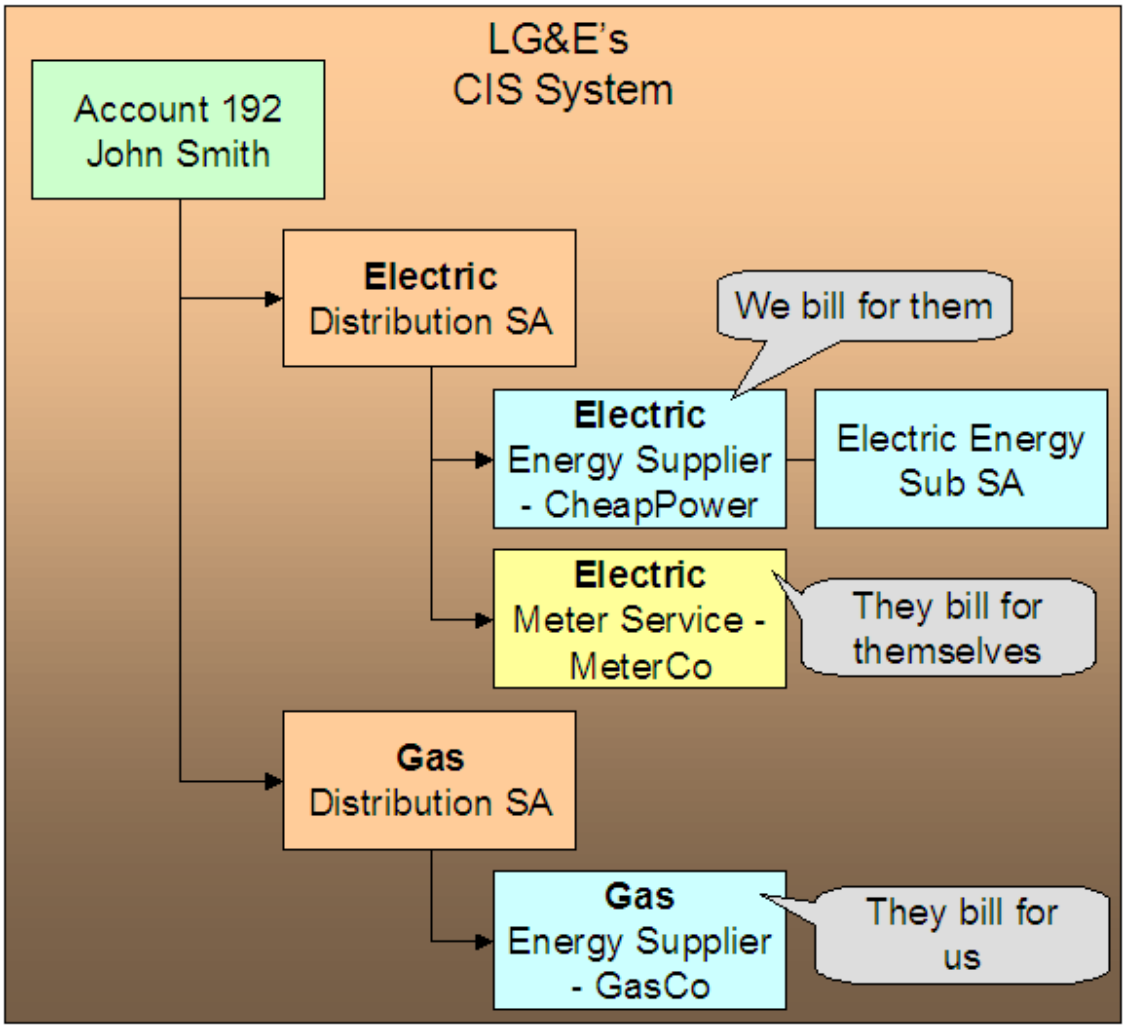
There are some restrictions in respect of permissible combinations of service providers that may supply service to a customer as described in the following points:

Copyright © 2026, Oracle and/or its affiliates.

- If the system encounters a customer with a We Bill For Them (WBFT) - Bill Ready service provider and another service provider that is They Bill For Us (TBFU) - Rate Ready, a billing error will be produced. Why? Because TBFU - Rate Ready means they have everything they need to calculate our bills and therefore we do not interface bill lines to them. If we don't interface our bill lines to them, then we can't interface the charges that were interfaced from the WBFT - Bill Ready service provider. You may wonder why we don't prohibit WBFT - Rate Ready and TBFU - Rate Ready, because it's conceivable for the TBFU service provider to have our rate and the WBFT service provider's rate.
- If the system encounters a WBFT - Pay At Pay Time service provider and another service provider that is TBFU, a billing error will be produced. Why? Because when the system detects a TBFU service provider, it transfers the receivable from the customer to the service provider (and therefore the customer's account will never be paid).

Different Service Providers Can Bill Different Services

Be aware that the system determines billing relationships at the service agreement level, NOT at the account level. To make the point, check out the following customer in a distribution company's system:



Be aware of the following in respect of the above illustration:

Copyright © 2026, Oracle and/or its affiliates.

- The distribution company (LG&E) distributes both electricity and gas.
- The customer has a choice of energy service providers for both gas and electricity.
- This customer - John Smith - purchases electricity from CheapPower, and purchases gas from GasCo.
- LG&E provides billing services for CheapPower.
- GasCo provides billing service for LG&E.

In this situation, LG&E will send bills to the customer that contain both electric distribution and energy charges (but no gas distribution charges). GasCo will also send bills to the customer; these will contain both LG&E's gas distribution charges as well as their own energy charges.

If You Deal With TBFU Service Providers, You Cannot Reopen Bills

NOT_SUPPORTED

Refer to Bill Lifecycle for information about reopening previously completed bills.

If your organization deals with They Bill For Us (TBFU) service providers, a great deal happens when a bill is completed (e.g., the receivable is transferred from the customer to the service provider, we may mark the bill segments and adjustments for routing to the service provider, etc.). These things cannot be undone and therefore the system will not let you reopen bills when these things have occurred.

Note

Technical rule. The specific rule that prevents the reopening of bills is as follows: if a bill contains a service agreement whose SA type has one or more bill completion algorithms, the system will not allow the bill to be reopened. Refer to SA Type - Algorithms for more information about bill completion algorithms.

Consumption Relationships

When you set up a service provider, you must define your organization's consumption relationship with the service provider - a service provider may send the customers' consumption to you, you may send consumption to them, or you may have no consumption relationship with a given service provider.

The topics in this section provide a wealth of information about the various consumption relationships and the ramifications of each.

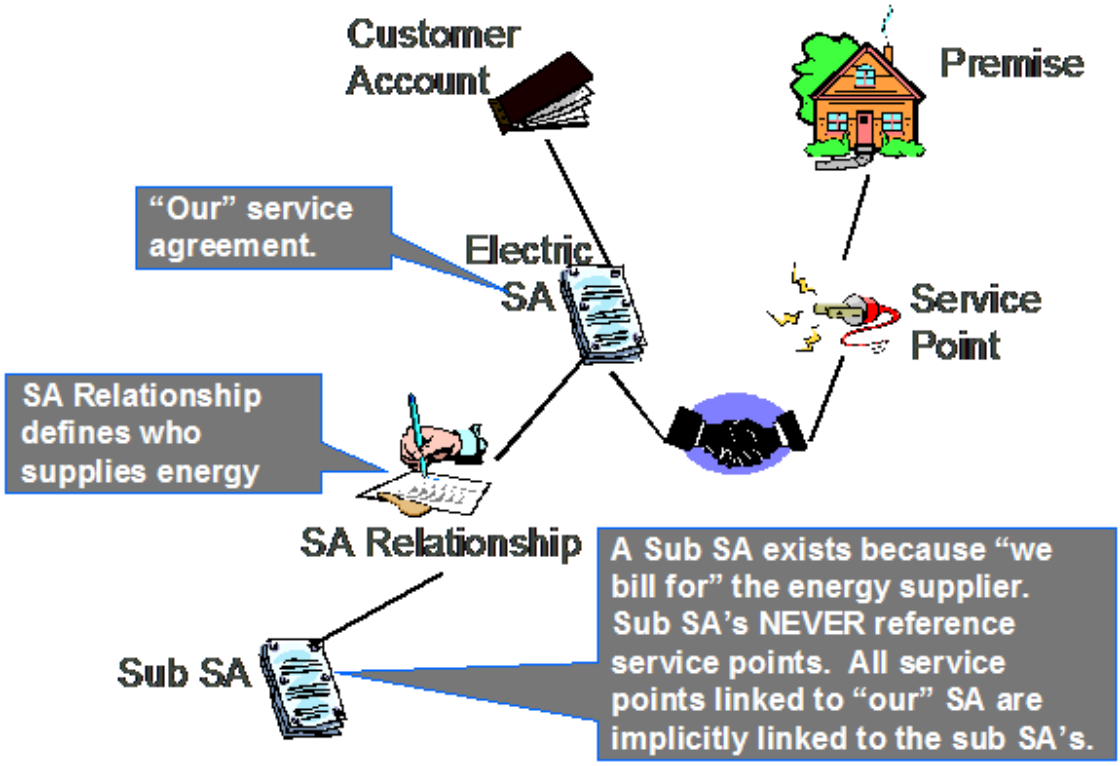
- Only The Master SA Is Linked To Service Points
- We Can Send Billed Consumption To Any Service Provider
- Routing Consumption To Service Providers - Technical Implementation
- We Can Receive Consumption From Service Providers

Copyright © 2026, Oracle and/or its affiliates.

- MDMAs And Service Cycles

Only The Master SA Is Linked To Service Points

The following diagram makes the point that service points cannot be linked to sub SAs (i.e., service agreements that exist to hold charges associated with We Bill For Them service providers). This is because all service points associated with our "master" SA are implicitly linked to all sub SAs.



It's important to understand why the system does not allow sub SAs to reference service points:

- The consumption associated with all service providers should be the same, otherwise the customer will receive inconsistent bills from different service providers.
- The easiest way to ensure consumption is the same for all service providers is to make sure that they all have the same service points (which they must if sub SAs "inherit" their service points from their master).

We Can Send Billed Consumption To Any Service Provider

You can send consumption to any service provider. It doesn't matter what their billing relationship is. Information on the service provider object tells the system if AND how to send consumption to a service provider.

Rather than send raw reads to service providers, we download consumption that has been calculated and snapshot onto the "master" bill segment. We send this billed consumption because:

Copyright © 2026, Oracle and/or its affiliates.

- It is clean and validated
- Register indexes have been subtracted
- Multiple registers have been summed

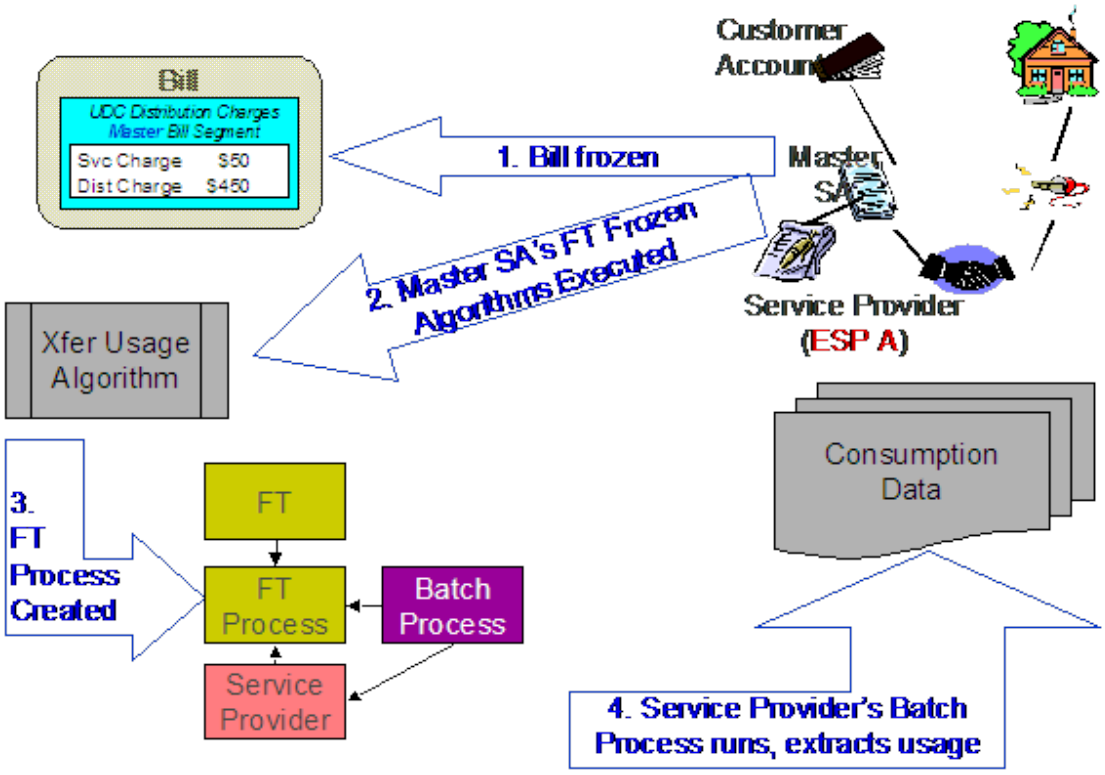
> **Note**
>
> We only send consumption after a bill segment is Frozen. If your organization supports We Bill For Them - Bill Ready service providers AND you are the source of consumption used by these service providers to calculate their charges, please refer to The Bill Ready Calculation Method for an explanation of how the system waits a given amount of time for the recipient of the consumption to interface their charges back to the system before sending the bill to the customer. In other words, the bill sent to the customer should contain the bill segment that triggered the consumption download as well as bill segments containing uploaded billable charges.

Routing Consumption To Service Providers - Technical Implementation

> **Warning**
>
> This section describes, technically, how we send consumption to service providers. If you aren't technically inclined, skip this section.

The following illustration shows the logical steps involved with sending consumption to service providers.

Copyright © 2026, Oracle and/or its affiliates.



The following points explain the steps:

- When a financial transaction (FT) is frozen, the system executes the FT Freeze algorithms defined on the SA(s) SA type.
- If you've set up the system properly (i.e., you've put the appropriate FT Freeze algorithm on the master SA's SA type), one of these algorithms will determine if there are service providers associated with the master SA who need consumption. If so, it will insert a row on the FT Process table.
- Rows on the FT process table are used as "triggers" for batch processes. In this case, the batch process that is triggered is the one that downloads billable consumption to the service provider. If multiple service providers need consumption, multiple rows will be inserted. The ID of the batch process that is referenced on the trigger comes from the Service Provider's Consumption Download Process.

We Can Receive Consumption From Service Providers

We can receive consumption from any source. The system can receive consumption from service providers. Each customer must have:

- A premise
- A service point
- A device with measuring components sufficient to hold the interfaced consumption

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You don't have to perfectly model the service points and meters.** If you are not the distribution company, you may be worried about how to keep devices and service point information up-to-date. Be aware that you don't have to model the SPs and devices perfectly. Why? Because you just need to set up enough information so that consumption can be received accurately.

MDMAs And Service Cycles

> **Warning**
>
> This section uses terminology and concepts described in The Cyclical Meter Read Process.

A Meter Data Management Agency (MDMA) is a service provider who reads meters. In some locales, a meter can be read by a variety of MDMAs, in other locales there is no MDMA as the meter is read by the distribution company. If your organization has MDMAs, then you need to be aware of the following:

- MDMAs must be linked to service agreements as a service provider. Like all service providers, they can change over time.
- When an MDMA is reading a service agreement's meters, the MDMA may override the customer's service cycle (and schedule) with their own cycle. If they do this, the customer's service cycle in the MDMA's system is defined on the service agreement's SA relationship information for the MDMA. Note: whether or not a service provider can override a service cycle is controlled by a switch on the service provider's record. Refer to Setting Up Service Providers for more details.
- A service provider's service cycle schedules are maintained using the system's normal service cycle schedule. The ID of the service provider associated with each schedule is defined on the service cycle; in other words, if a service provider reads meters and they can override the customer's service cycle, the service provider's service cycles must be defined in the system. Refer to Designing Service Cycles, Routes, And Schedules for more information.
- At billing time, the system determines if a service agreement is covered by an MDMA. If so, it uses the service cycle defined on the service agreement's MDMA SA relationship record.
- If your organization ever reads the customers' meters, your regular read cycle should be maintained on your service points. You can think of the service cycle that is defined on a service agreement's MDMA SA relationship record as an override of your service points normal service cycle.
- If the meter read download process detects that a service point is linked to a service agreement with an active MDMA, it still creates a meter read download staging record; however, it marks it as Do Not Need To Read. This means that when a service point is no longer read by an MDMA, the meter read will be downloaded normally.

Copyright © 2026, Oracle and/or its affiliates.

- Refer to We Can Receive Consumption From Service Providers for a description of how a service provider interfaces consumption into the system.

Deposits Issues

> **Warning**
>
> This section uses terminology and concepts described in The Big Picture Of Deposits.

Deposits should be held using normal deposit service agreements (SAs). You should NOT use the Sub Service Agreements (sub SA) functionality to hold or bill for deposits because deposit service agreements do not have the same state transition as do master SAs (e.g., you can activate or stop a deposit independent from its master).

However sub SAs can be covered by a deposit. If so, their SA type must reference a deposit class. To make the point, let's examine a few scenarios:

- Assume you have a sub SA for your own charges (this can happen when we use sub SAs to unbundle charges from the Master SA). In this case, it is likely that the sub SA and master SA will be in the same deposit class. This means that a single deposit SA would cover both the master and the sub SA.
- Assume They Bill For Us (Bill or Rate Ready). In this situation, we still have a master SA for our charges and we transfer the charges to the service provider who does the billing. In this case, it is likely that we would be holding the deposit for the service provider, not on the end-use customer. If we are in a situation where 1) we cannot hold a deposit against the service provider, and/or 2) we are not assured of the service provider paying us when the customer doesn't pay them, then we might want to put the master SA in a deposit class and hold a deposit against the customer's account using a normal deposit SA. We would not expect the service provider to bill the customer for the deposit, so we don't need a sub SA. We bill the customer directly for the deposit using our normal deposit SA.
- Assume We Bill For Them (Bill or Rate Ready). In this situation, we could hold a normal deposit SA for the customer's master SA. For sub SAs, we have two scenarios:
  - **We pay at billing time.** Since we purchase the receivable, we would want to increase our normal deposit to cover the Sub SA. To do this, the sub SA's SA type's deposit class should be the same as our master SA's deposit class.
  - **We pay at payment time.** It seems unlikely that we would want to hold a deposit on behalf of a service provider when we don't purchase the receivable. However, it is possible to do so by putting the sub SA into its own deposit class. If you did this, the system will require a separate deposit SA for the service provider's deposit. The system would calculate and refund such deposits using the algorithms defined on the new deposit SA's SA type's deposit class. It's important to be aware that the deposit is not held with respect to the specific service provider. Rather, it is just held in the system as separate deposit that could be used for any service agreement that belongs to its deposit class.

Credit and Collection Issues

Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> This section uses terminology and concepts described in The Big Picture Of Credit & Collections (C&C).

C&C is only tricky if you deal with We Bill For Them service providers. We'll run through the service provider billing relationships to explain why:

- If we have a Dual relationship with a service provider, we don't have their debt, so we only have a responsibility to tell them when we cut a customer (via a Notification). We don't have to worry about collecting their debt.
- If we have a They Bill For Us relationship with a service provider, there is no debt on the customer's SA because it gets transferred to the service provider (and the service provider's SA will fall into arrears if they don't pay us).
- If we have a We Bill For Them or It's Us relationship with a service provider, the customer's debt associated with the service provider's service is maintained on a sub SA (i.e., it is segregated from our debt). This segregation of debt is both a powerful feature and a cause of administrative difficulties. The topics in this section provide more information about this issues.
- Debt Class Recommendations
- Severing Service

Debt Class Recommendations

A service agreement's debt class is an important element in determining how a customer's debt is collected. In general, we recommend the following:

- If the service provider has a billing relationship of It's Us, we recommend the sub SAs belong to the same debt class as the "master". Why? Because both SAs' overdue debt should probably be grouped together under a single collection process.
- If you buy the receivable from the service provider (i.e., the service provider has a payment relationship of Pay at Billing Time), we recommend the sub SAs belong to the same debt class as the "master". Why? Because both SAs' overdue debt should probably be grouped together under a single collection process.
- If you don't buy the receivable from the service provider (i.e., the service provider has a payment relationship of Pay at Pay Time), you may want to use a different debt class on the sub SA. Why? Because you may collect the service provider's debt differently.

> **Note**
>
> **Bottom line.** If both the "master" and the sub SAs fall into arrears, you will have 1 or 2 collection processes, it all depends on the debt class assigned to each SA type.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Automating Your C&C Activities for information describing how debt class plays a part in this processing.

Severing Service

Sub SAs and severance is tricky. Why?

- Because it's possible for the master SA to be in arrears when the sub SA isn't (for all the standard reasons - directed payments, cancel / rebills, etc.).
- Because it's possible for the sub SA to be on one collection process and the master to be on another (due to different debt classes or different time lines).

Both of these situations could result in severance starting for only one of the service agreements in the master / sub relationship. However, YOU CAN'T CUT SERVICE FOR ONE WITHOUT CUTTING THE OTHER because there is only one service point.

Before we describe how to deal with this conundrum, we'd like to remind you that the system starts a unique severance process for each SA (sub or normal) to be severed. It only creates a severance process for those service agreements linked to a collection process when the collection process' Start Severance event is activated. The type of severance process that is created is controlled by each service agreement's SA Type's severance criteria. Please keep in mind the following when designing these severance processes:

- Only The "Master" Service Agreement Is Linked To Service Points. This means only master SAs should have a "cut for non payment" severance event. Note: typically, such a severance process will expire the "master SA" several days after the cut event if funds are NOT received.
- As described under Sub SA State Transition, a sub SA becomes Pending Stop (and eventually Stopped) when its "master" is stopped. This means the sub SAs will be finaled when the master is finaled.
- If you start severance on a sub SA when the master isn't being severed, you have a problem because you can't cut the sub SA independent from the master SA.

We'll use an example to illustrate how you should design your severance processes to deal with the above challenge. Assume you have a master and a sub SA where both are being managed under the same collection process. Also assume that the Start Severance event kicks off on 18-Dec-1999. In this situation, we'd recommend the following severance processes to be kicked off.

Master SA Severance Process

Trigger Dt.	Event Type	Status
18-Dec-99	Set door hanger	Pending
Pending	Cut for non pay	Pending
Pending	Expire SA	Pending

Sub SA Severance Process

Trigger Dt.	Event Type	Status
3-Jan-00	Create To Do Entry	Pending

Copyright © 2026, Oracle and/or its affiliates.

Notice that the sub SA's severance process contains a single event that generates a To Do Entry on a date in the future of the Expire SA event on the Master SA. This entry should be something like "sub being severed independent of its master". This event will only be triggered if the master SA is paid off and the sub SA isn't. Why? Because if the master SA's Expire SA event is executed, the Sub SA will be Stopped and stopping a SA cancels outstanding severance processes. If the sub gets paid, the system will cancel the sub's severance process.

Let's change the example and assume that the master starts severance and the sub doesn't. In this situation, the master SA will eventually hit the Expire SA event and the sub SA will also stop. There's no alternative.

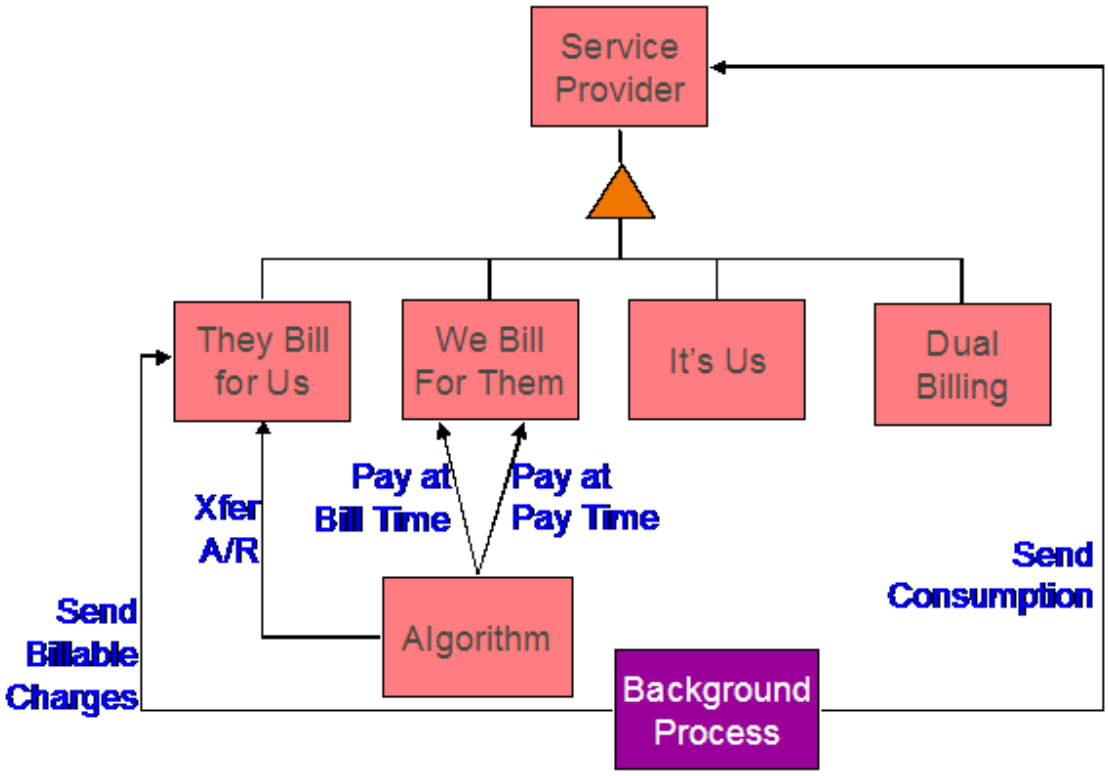
And let's change the example again and assume the sub starts severance and the master doesn't. In this situation, the To Do Entry will only be created X days after the start of severance. If you can't stand this date being X days in the future of the creation of the severance process, create an "Severance Criteria Algorithm" that checks if the master is not being severed or collected and generates a different severance process (with a different start date). Refer to Designing Your Severance Procedures for more information about Severance Criteria Algorithms.

An Object-Oriented Perspective Of Service Providers

> **Warning**
>
> Skip this section if you are not technically inclined.

The following object diagram illustrates in a concise format the various types of service providers and the plug in algorithms and processes available for each.

Copyright © 2026, Oracle and/or its affiliates.



How Do You Communicate With Service Providers?

You communicate with service providers (and they communicate with you) using the following mechanisms:

- You can use traditional interface techniques. For example, if you send consumption to service providers every month (so they can compute their charges), you could use the Consumption Download interface.
- You can use "notifications" to communicate with service providers. Notifications are electronic transactions that service providers exchange to communicate information about a customer. For example, you could use a notification to send a message to an energy supplier when a customer stops service. Note: the term "Direct Access Service Request" (DASR) is synonymous with Notification.

The difference between notifications and traditional interfaces is subtle. Think of notifications as a generic interface that can be used to communicate many different things (e.g., you can use notifications to advise stops, meter exchanges, supplier switches). Traditional interfaces communicate only one thing (and you need one traditional interface for each "thing"). For example, one interface is devoted to downloading consumption, another is responsible for uploading pass through charges, etc.

Off Cycle Switch

If the customer care and billing application integrates with meter data management (mdm) system, it may be set up to generate sub-bill segments for each and every service provider having **We Bill for Them - Rate Ready** or **It's Us** billing relationship, that is effective in the billing period. This requires that the meter data management system can calculate the consumption for each effective SA relationship period.

Copyright © 2026, Oracle and/or its affiliates.

If the switch is from or to a service provider having **They Bill for Us** billing relationship, separate bills have to be generated manually for each relationship effective in the billing period. This restriction is due to requirement for CCB application to generate a bill for a **They Bill for Us** relationship, so that the customer's charges are calculated and eventually transferred to the service provider. The generated bill for **They Bill for Us** billing relationship is not sent to the customer.

- Billing Master Configuration
- Rate Schedule
- Sub-SA Type
- Usage Request
- Sending Consumption to Service Providers

Billing Master Configuration

To enable Off Cycle Switch, navigate to **Admin**, and then **General**, and then **Master Configuration** and add a new entry for Billing Master Configuration.

Specify the values for **Off Cycle Switch Option**, **Implementation Date** and **Off Cycle Switch Option for Old Bill**. Refer to the embedded help on billing master configuration portal for details.

Rate Schedule

Rate schedules assigned to the sub-SA of service provider, having **We Bill for Them - Rate Ready** or **It's Us** billing relationship, must be configured with an SQ rule (for component-based engine) or a pre-processing calculation rule (for rule-base engine) that will obtain the sub-usage's consumption and load it into the bill segment SQ collection during rate application.

For more information, refer to the base package algorithm types provided for this purpose: US (component based engine) and C1-ACRSUBUSG (rule based engine).

Sub-SA Type

When implementing this feature, sub-SA types of service provider having **We Bill for Them - Rate Ready** or **It's Us** billing relationship must be configured as non-bill determinant.

Usage Request

Usage request is sent to the Oracle Utilities Meter Data Management (MDM) system, to obtain the calculated consumption for the billing period - this is used to calculate master bill segment's charges.

In addition, sub-usage request is sent to the mdm system, for each relationship that is effective during the billing period. Calculated consumption on these sub-usage requests is used to calculate sub-bill segment charges.

Sending Consumption to Service Providers

Calculated consumption on these sub-usage requests also serves as the source for consumption download process if consumption is required to be sent to the service provider. A

Copyright © 2026, Oracle and/or its affiliates.

base package algorithm type - C1-STGCNSMP, stages the download by creating an entry in the general process table, capturing the sub-usage id, batch control for consumption download and the batch number.

> **Note**
>
> It is implementation responsibility to build the consumption download batch process that will extract the consumption for sub-usage request as required.

Designing Your SA Relationship Types and Service Providers

As explain in A Service Agreement Can Have Many Types Of Relationships, SA relationship types and service providers are only required when you subcategorize your service agreements.

The topics in this section describe how to design your SA relationship types.

- Designing SA Relationship Types
- Designing Service Providers
- Designing Your SA Types And Start Options For Sub SAs
- Reference Send Consumption Algorithm On Master SA Types
- Reference TBFU Algorithm On Master SA Types
- Designing SA Types For Service Provider Financial Settlements

Designing SA Relationship Types

The easiest way to design SA relationship types is to start with the matrix of SA types designed in SA Types And The Financial Design. For each SA type in the matrix, determine if either of the following questions is true:

- Can companies other than your own provide some subcategory of the service (and do you have some type of interaction with these companies)? For example, this would be true if you are an energy supply company because a different company is responsible for distributing the power to the customer (and you probably exchange consumption and financial transactions with this company).
- Does your organization use different rates for subcategories of the service? For example, this would be true if you use separate rates for water and wastewater service (even though both are based on the customer's water consumption).

If either of the above is true, you will need a SA relationship type for each subcategory of service.

We'll use an example to help make the point. Using the SA types we designed earlier (see SA Types And The Financial Design), we'll assume the following:

- We are designing the system for an electric, gas, water, waste water, and cable utility.
- Residential electric customers have a choice of energy supplier.

Copyright © 2026, Oracle and/or its affiliates.

- Commercial and industrial electric customers can choose an energy supplier and a meter service provider.
- Commercial and industrial gas customers have a choice of energy supplier.

> **Note**
>
> We are not showing most of the SA types that we designed earlier because they do not have subcategories of service.

CIS BU/ SA Type	SA Relationship Type
G/COM	Energy supply
G/IND	Energy supply
E/RES	Energy supply
E/COM	Energy supply
	Meter service
E/IND	Energy supply
	Meter service

> **Note**
>
> Notice that we did not design a SA relationship type for our own distribution service. This is because our relationship type is implied (e.g., if you are a distribution company, you do not have to set up a SA relationship type for distribution service because the customer's "master" service agreement is implicitly associated with distribution service).

Designing Service Providers

After you design your SA relationship types, you need to list every potential service provider for each SA relationship type.

CIS BU/ SA Type	SA Relationship Type	Service Provider
G/COM	Energy supply	AmeriGas
		TransGas
		Green Power
G/IND	Energy supply	AmeriGas
		TransGas
		Green Power
E/RES	Energy supply	Green Power
		Cheap Power

Copyright © 2026, Oracle and/or its affiliates.

		Us
E/COM	Energy supply	ElectriCorp
		Cheap Power
		TeniCorp
	Meter service	MeterCorp
		Us
E/IND	Energy supply	ElectriCorp
		Cheap Power
		TeniCorp
	Meter service	MeterCorp
		Us

Next, list each unique service provider identified above:

Service Provider
AmeriGas
TransGas
Green Power
Cheap Power
Us
ElectriCorp
TeniCorp
MeterCorp

You will have at least one service provider for each entry in the above list. However, you may have to set up more than one service provider in the system for a given company. The topics below explain how this happens.

- Billing Relationship Segmentation
- We Bill For Them - Payment Relationship Segmentation
- Consumption Relationship Segmentation
- Other Segmentations
- Geographic Area Segmentation

Billing Relationship Segmentation

As described under Billing Relationships, a service provider may bill for you, you may bill for them, or you may each send a separate bill to the customer. In the table below, we have shown the assumed billing relationships for each service provider.

Service Provider	Billing Relationship
AmeriGas	We Bill For Them - Bill Ready
TransGas	We Bill For Them - Bill Ready

Copyright © 2026, Oracle and/or its affiliates.

Green Power	We Bill For Them - Rate Ready
Cheap Power - Res	We Bill For Them - Rate Ready
Cheap Power - Com/Ind	Dual Billing
Us - Billable	It's Us
Us - Non billable	None
ElectriCorp	They Bill For Us - Bill Ready
TeniCorp	We Bill For Them - Bill Ready
MeterCorp	Dual Billing

Notice that we had to introduce additional service providers:

- Cheap Power has two service providers - one for residential customers, the other for commercial/industrial customers. This is necessary because we provide billing service for them for residential customers (We Bill For Them - Rate Ready), but for commercial and industrial customers they bill for themselves (Dual Billing).
- Our own service provider (the service provider known as "Us") has two service providers - one for energy supply because we bill for the energy we supply (It's Us), and another for meter service because we don't create bills for meter service (None).

We Bill For Them - Payment Relationship Segmentation

As described under Pay At Bill Time vs. Pay At Pay Time when you provide billing service for a service provider you have to define if you pay the service provider when you bill the customer OR only later, when the customer pays you.

In the table below, we have shown the payment relationships for each service provider.

Service Provider	Billing Relationship	Payment Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time
Green Power	We Bill For Them - Rate Ready	Pay At Billing Time
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time
Cheap Power - Com/Ind	Dual Billing	N/A
Us - Billable	It's Us	N/A
Us - Non billable	None	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time
MeterCorp	Dual Billing	N/A

Notice that we had to introduce an additional service provider for TeniCorp because for commercial customer we purchase the receivable (Pay At Bill Time), but for industrial customers we only pay them when we're paid by the customer (Pay At Pay Time).

Copyright © 2026, Oracle and/or its affiliates.

Consumption Relationship Segmentation

As described under Consumption Relationships a service provider may send the customers' consumption to you, you may send consumption to them, or you may have no consumption relationship with a given service provider.

In the table below, we have shown the consumption relationships for each service provider.

Service Provider	Billing Relationship	Payment Relationship	Consumption Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
Green Power	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Com/Ind	Dual Billing	N/A	N/A
Us - Billable	It's Us	N/A	N/A
Us - Non billable	None	N/A	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time	We Send Consumption
MeterCorp	Dual Billing	N/A	N/A

Notice that we didn't have to proliferate service providers due to consumption relationships.

Other Segmentations

The earlier parts of this discussion described the most common factors that cause the creation of service providers. However, many obscure factors could cause the introduction of more service providers. In this section we explain these more obscure factors.

- Payment Method
- Transfer Receivable Method
- Billable Charge Download Process
- Consumption Download Process
- Notification Upload Processing
- Notification Download Processing
- Financial Settlement Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

Payment Method

As described under Pay At Bill Time vs. Pay At Pay Time, when you provide billing service for a service provider you have to pay the service provider at some point in time. The algorithm that defines the amount to pay (and how the related adjustment is generated) is defined on the service provider record. If a service provider has different payment algorithms for different customer segments, you must split the service provider accordingly.

Transfer Receivable Method

As described under When They Bill For Us, They Owe Us Money, a service provider will owe you money if they provide billing service for you. The algorithm that defines how to transfer the customer's receivable to the service provider is defined on the service provider record. If a service provider has different transfer A/R algorithms for different customer segments, you must split the service provider accordingly.

Billable Charge Download Process

As described under They Bill For Us - Bill Ready, billable charges are interfaced to service providers who provide billing service for you. The background process that performs the interface of billable charges is defined on the service provider record. If a service provider has different billable charge interfaces for different customer segments, you must split the service provider accordingly.

Consumption Download Process

As described under We Can Send Billed Consumption To Any Service Provider, consumption can be sent to any service provider. The background process that performs the interface of consumption is defined on the service provider record. If a service provider has different consumption interfaces, you must split the service provider accordingly.

Notification Upload Processing

A service provider can send you notifications. Whenever a notification is uploaded, the system creates a workflow process to process each such notification. The type of workflow process that's created is controlled by the service provider's workflow process profile. If a service provider requires a different workflow process for a given type of notification (for whatever reason), you must split the service provider accordingly.

Notification Download Processing

The system will send notifications to service providers when something noteworthy happens and when information is needed from a service provider. The type of notification that is sent to a service provider and the background process that interfaces the notification to the service provider is defined on the service provider's notification download profile. If a service provider

Copyright © 2026, Oracle and/or its affiliates.

requires a different type of notification to be sent or they have different interface protocols for a given type of notification, you must split the service provider accordingly.

Financial Settlement Service Agreement

As described under Service Providers Have Service Agreements Too, service providers have service agreements. These service agreements contain how much you owe the service provider (if you bill for them) and how much they owe you (if they bill for you). If you want to have separate service agreements for financial settlements associated with distinct customer segments, you must split the service provider accordingly.

Geographic Area Segmentation

The following table reflects the new service providers that were added since we started Designing Service Providers.

CIS BU/ SA Type	SA Relationship Type	Service Provider
G/COM	Energy supply	AmeriGas
		TransGas
		Green Power
G/IND	Energy supply	AmeriGas
		TransGas
		Green Power
E/RES	Energy supply	Green Power
		Cheap Power - Res
		Us - Billable
E/COM	Energy supply	ElectriCorp
		Cheap Power - Com/Ind
		TeniCorp - Com
	Meter service	MeterCorp
		Us - Non billable
E/IND	Energy supply	ElectriCorp
		Cheap Power - Com/Ind
		TeniCorp - Ind
	Meter service	MeterCorp
		Us - Non billable

Next, determine the postal code ranges in which a service provider is allowed to provide service.

CIS BU/ SA Type	SA Relationship Type	Service Provider	Postal Range
G/COM	Energy supply	AmeriGas	94000 - 95999

Copyright © 2026, Oracle and/or its affiliates.

		TransGas	94000 - 95999
		Green Power	94000 - 95999
G/IND	Energy supply	AmeriGas	94000 - 95999
		TransGas	94000 - 95999
		Green Power	94000 - 95999
E/RES	Energy supply	Green Power	94000 - 95999
			93000 - 93999
		Cheap Power - Res	94000 - 95999
		Us - Billable	94000 - 95999
E/COM	Energy supply	ElectriCorp	94000 - 95999
		Cheap Power - Com/Ind	94000 - 95999
		TeniCorp - Com	94000 - 95999
	Meter service	MeterCorp	94000 - 95999
		Us - Non billable	94000 - 95999
E/IND	Energy supply	ElectriCorp	94000 - 95999
		Cheap Power - Com/Ind	94000 - 95999
		TeniCorp - Ind	94000 - 95999
	Meter service	MeterCorp	94000 - 95999
		Us - Non billable	94000 - 95999

Next, we need to strip off the SA types because the postal ranges are defined for combinations of service provider and SA relationship type. Notice the problem - we have a service provider - Green Power has different postal ranges for the same SA relationship type. You have two ways to fix this problem, you can split your service provider (have one for the gas and another for the electric), or you can split the SA relationship type (have one for the gas and another for the electric). We've chosen the former in our example.

SA Relationship Type	Service Provider	Postal Range
Energy supply	AmeriGas	94000 - 95999
	TransGas	94000 - 95999
	Green Power - Gas	94000 - 95999
Energy supply	AmeriGas	94000 - 95999
	TransGas	94000 - 95999
	Green Power - Gas	94000 - 95999
Energy supply	Green Power - Electric	94000 - 95999
		93000 - 93999
	Cheap Power - Res	94000 - 95999
	Us - Billable	94000 - 95999
Energy supply	ElectriCorp	94000 - 95999
	Cheap Power - Com/Ind	94000 - 95999
	TeniCorp - Com	94000 - 95999
Meter service	MeterCorp	94000 - 95999

Copyright © 2026, Oracle and/or its affiliates.

	Us - Non billable	94000 - 95999
Energy supply	ElectriCorp	94000 - 95999
	Cheap Power - Com/Ind	94000 - 95999
	TeniCorp - Ind	94000 - 95999
Meter service	MeterCorp	94000 - 95999
	Us - Non billable	94000 - 95999

In the table below, we have shown the final list of service providers.

Service Provider	Billing Relationship	Payment Relationship	Consumption Relationship
AmeriGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TransGas	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
Green Power - Electric	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Green Power - Gas	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Res	We Bill For Them - Rate Ready	Pay At Billing Time	N/A
Cheap Power - Com/Ind	Dual Billing	N/A	N/A
Us - Billable	It's Us	N/A	N/A
Us - Non billable	None	N/A	N/A
ElectriCorp	They Bill For Us - Bill Ready	N/A	N/A
TeniCorp - Com	We Bill For Them - Bill Ready	Pay At Billing Time	We Send Consumption
TeniCorp - Ind	We Bill For Them - Bill Ready	Pay At Pay Time	We Send Consumption
MeterCorp	Dual Billing	N/A	N/A

Designing Your SA Types And Start Options For Sub SAs

When you were Designing Service Providers, you defined the service providers that were valid for every combination of SA type and SA relationship type. If you provide billing services for another service provider or if you subcategorize your own services, you another task - you have to design the SA types for your sub SAs.

As described earlier, there will be a separate Sub Service Agreement for every SA relationship for which we calculate a bill segment. Every sub SA must reference an SA type. The following table shows sample SA types (notice that they are only used for We Bill For Them and It's Us service providers).

CIS BU/ SA Type	SA Relationship Type	Service Provider	SA Type(s) for Sub SAs

Copyright © 2026, Oracle and/or its affiliates.

G/COM	Energy supply	AmeriGas	AG1
		TransGas	TG1
		Green Power - Gas	GP-GC1
G/IND	Energy supply	AmeriGas	AG1
		TransGas	TG1
		Green Power - Gas	GP-GI1
E/RES	Energy supply	Green Power - Electric	GP-ER1
		Cheap Power - Res	CP-ER1
		Us - Billable	US-ER1
E/COM	Energy supply	ElectriCorp	Not applicable - they bill for us
		Cheap Power - Com/Ind	Not applicable - dual billing
		TeniCorp - Com	TC1
	Meter service	MeterCorp	Not applicable - dual billing
		Us - Non billable	Not applicable - no billing
E/IND	Energy supply	ElectriCorp	Not applicable - they bill for us
		Cheap Power - Com/Ind	Not applicable - dual billing
		TeniCorp - Ind	TC1
	Meter service	MeterCorp	Not applicable - dual billing
		Us - Non billable	Not applicable - no billing

The design steps required to set up these SA types are similar to those described under Designing SA Types For Service Agreements With Service Points. The following points provide a few suggestions that will help you design your SA types for sub SAs:

- The business unit should be the same as defined for the master service agreement.
- Service type should be set up as per Service Segmentation.
- Distribution code should be set up as per Receivable Segmentation.
- Obviously, the sub SA switch should be turned on.
- Start options:
  - If the SA type is used for a We Bill For Them - Rate Ready service provider or for yourself, the start options should be Required because rates, contract riders and/or contract values will be populated on the sub SA from a start option. Refer to Automatic Creation of Sub SAs for more information.
  - If the SA type is used for a We Bill For Them - Bill Ready service provider, the start options should be not allowed because we don't need to default rates, contract riders and/or contract values on billable charge sub SAs.
- The payment distribution and late payment information should be set up as for any SA type. Refer to Cash Distribution Segmentation and Late Payment Charge Segmentation for more information.

Copyright © 2026, Oracle and/or its affiliates.

- If the SA type is used for We Bill For Them - Bill Ready service provider, the special role should be Billable Charge, otherwise it should not be used. Remember, you should not use sub SAs for Cash Deposits.
- Deposit class should be used if the sub SA is covered by a deposit. Refer to Deposits Issues for more information.
- The one time charge switch should be off.
- Bill segment type:
  - If the SA type is used for a We Bill For Them - Bill Ready service provider, the bill segment type should reference a bill segment creation algorithm that creates bill segments from billable charges (and generates bill segment errors until the last night of the bill cycle). Refer to The Bill Ready Calculation Method for more information.
  - If the SA type is used for a We Bill For Them - Rate Ready service provider, the bill segment type should reference a get consumption algorithm that gets consumption from the master SA, and a bill segment creation algorithm that applies a rate.
- Specify a characteristic premise is required if the sub SA is associated with premise-oriented service.
- The calendar billing options should not be used.
- The recurring charge information should not be used.
- Sub SAs used for We Bill For Them - Rate Ready service providers should NOT be eligible for budgets and therefore the eligible for budget switch should be off. This admonition is given because budget billing causes current amount due to be out-of-sync with payoff amount due and we don't want this to happen for sub SAs. Why? Because we use the bill segments associated with these sub SAs to determine how much we owe the service provider.
- If the SA type is used for a We Bill For Them - Rate Ready service provider or for your own company, rates need to be set up; otherwise they should not be used.
- Sub SAs never reference service points. Refer to Only The "Master" Service Agreement Is Linked To Service Points for more information.
- Adjustment types profiles should be set up accordingly.
- Refer to Credit and Collection Issues for recommendations in respect of the debt class, write off debt class, and severance criteria associated with these SA types.
- Billable charge templates should not be used.
- Completion algorithms cannot be used for sub SAs.
- If the SA type is used for a We Bill For Them service provider, you should link to the SA type the FT freeze algorithm that controls how we pay the service provider. Refer to Pay At Bill Time vs. Pay At Pay Time for how this algorithm is used.
- The creation of bill segments for the sub SAs occurs after the bill segment for the related master SA is created. If you populate a billing processing sequence on an SA type for a sub SA, it is used to control the order in which the sub SAs for a given master SA are processed relative to each other.

Refer to Setting Up SA Types for how to set up these new SA types in the system.

Reference Send Consumption Algorithm On Master SA Types

Copyright © 2026, Oracle and/or its affiliates.

As explained under We Can Send Billed Consumption To Any Service Provider, when a master SA's bill segment is frozen, the system must check if there are any service providers who need the bill segment's consumption. If so, it sets up the data necessary to interface the master SA's consumption (snapshot on the bill segment) to the service provider(s). The system will only do this if you specify an appropriate FT Freeze Algorithm on the master SA types. Refer to SA Type - Algorithm (FT Freeze Algorithm) for more information.

Reference TBFU Algorithm On Master SA Types

As explained under They Bill For Us, when a bill is completed, the system needs to check if there are any service providers who bill for us associated with the bills "master" SAs. If so, it sets up the data necessary to interface the master SA's charges to the service provider and to transfer the receivable balance from the customer to the service provider. The system will only do this if you specify an appropriate FT Completion Algorithm on the master SA types. Refer to SA Type - Algorithm (Bill Completion Algorithm) for more information.

> **Note**
>
> If there are multiple master SAs on a bill, the financial transactions associated with each respective master SA could be routed to different service providers (e.g., one service provider could bill for gas and another could bill for electricity). Refer to Different Service Providers Can Bill Different Services for more information.

Designing SA Types For Service Provider Financial Settlements

As explained in Service Providers Have Service Agreements Too, We Bill For Them and They Bill For Us service providers require a service agreement. You must create SA types for these types of service agreements. The following points provide a few suggestions that will help you design these financial settlement SA types:

- Service type should probably be a non-service oriented service type.
- Distribution code for We Bill For Them settlement SAs should be a payable account (or treat it as a "contra" receivable. Refer to Receivable Segmentation for more information.
- The sub SA switch should be turned off.
- Start options are not allowed.
- The payment distribution and late payment information should be set up as for any SA type. Refer to Cash Distribution Segmentation and Late Payment Charge Segmentation for more information.
- Special role should not be used.
- Deposit class should be used if the settlement service agreement is covered by a deposit. This would probably only be used for They Bill For Us service providers (because they will owe us money).
- The one time charge switch should be off.
- These service agreements are not billable and therefore none of the billing information should be specified.

Copyright © 2026, Oracle and/or its affiliates.

- The characteristic premise switch should be off.
- Rates should not be used.
- Service points should not be used.
- Adjustment types profiles should be set up accordingly.
- Debt class, write off debt class, and severance criteria should be set up accordingly.
- Billable charge templates should not be used.
- Completion algorithms should not be used.
- Freeze algorithms should not be used.

Refer to Setting Up SA Types for how to set up these new SA types in the system.

Setting Up SA Relationship Information

In the previous section, Designing Your SA Relationship Types and Service Providers, we presented a case study that illustrated a mythical organization's SA relationship information. In this section we explain how to set up this information.

- Setting Up SA Relationship Types
- Setting Up Service Providers
- Setting Up SA Types and Start Options For Sub SAs
- Setting Up SA Types For Financial Settlements
- Update Master SA Types With FT Freeze and Bill Completion Algorithms
- Setting Up SA Relationships For SA Types
- Setting Up Market SA Relationship Options

Setting Up SA Relationship Types

Open **Admin**, and then **Open Market**, and then **SA Relationship Type** to define your SA relationship types. Refer to Designing SA Relationship Types for more information.

Description of Page

Enter an **SA Relationship Type** code, **Description** and **Service Provider Model** for every relationship type.

Specify the **Market Service ProviderService Provider Model** if the SA relationships of the type reference a Service Provider Market Option — i.e. Market, Provider Configuration Option and Market Relationship Service Provider are specified in addition to SA Relationship Type and Service Provider.

Otherwise, specify the **Service ProviderService Provider Model** .

Setting Up Service Providers

This portal is used to maintain Service Providers.

Copyright © 2026, Oracle and/or its affiliates.

Refer to The Big Picture of SA Relationships and Service Providers for more information.

You can access the portal from **Admin**, and then **Open Market**, and then **Service Provider**. You are brought to a query portal with options for searching for a specific service provider. Once a service provider is selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Service Provider** zone shows the details of the service provider. Refer to the zone's embedded help for more information.

Setting Up SA Types and Start Options For Sub SAs

The SA types and start options described under Designing Your SA Types And Start Options For Sub SAs must be set up. Refer to Setting Up SA Types for how to do this.

Setting Up SA Types For Financial Settlements

The SA types described under Designing SA Types For Service Provider Financial Settlements must be set up. Refer to Setting Up SA Types for how to do this.

Update Master SA Types With FT Freeze and Bill Completion Algorithms

Refer to SA Type - Algorithm for more information.

Setting Up SA Relationships For SA Types

This portal is used to maintain SA Type SA Relationship Types.

Refer to Designing Your SA Relationship Types and Service Providers for more information.

You can access the portal from **Admin**, and then **Open Market**, and then **SA Type SA Relationship Type**. You are brought to a query portal with options for searching for a specific SA Type SA Relationship Type. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

The **SA Type SA Relationship Type** zone shows the details of the SA Type SA Relationship Type. Refer to the zone's embedded help for more information.

Setting Up Market SA Relationship Options

The following sections describe how to set up Provider Configuration Options, Market Provider Configuration Options and Service Provider Market Options.

- Setting Up Provider Configuration Options
- Setting Up Market Provider Configuration Options

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Provider Configuration Options

This portal is used to maintain Provider Configuration Options.

You can access the portal from **Admin**, and then **Open Market**, and then **Provider Configuration Option**.

The following zones may appear as part of the portal's **Main** page:

- **Provider Configuration Option List** zone lists all Provider Configuration Option records. Broadcast a record to display the details of the selected record.
- **Provider Configuration Option** zone provides information about the selected Provider Configuration Option.
- **Market Provider Configuration Options** zone lists the existing Market Provider Configurations for the selected Provider Configuration Option. Maintenance actions are available on this zone.

Setting Up Market Provider Configuration Options

This portal is used to maintain Market Provider Configuration Options.

You can access the portal from **Admin**, and then **Open Market**, and then **Market Provider Configuration Option**. You are brought to a query portal with options for searching for a specific Market Provider Configuration Option. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** page:

- **Market Provider Configuration Option** zone shows the details of the Market Provider Configuration Option.
- **Service Provider Market Options** zone shows the service providers with their start dates, end dates (if applicable) and statuses. Maintenance actions are provided. A broadcast icon appears if the Service Provider Market Option has sub-SA(s) and/or entries in the Status History log. Clicking on the broadcast icon displays the following zones:
  - **Service Agreements** zone shows the Service Provider Market Option’s sub-SAs.
  - **Status History** zone shows the history of changes to the Service Provider Market Option, which can include changes to Start Date, End Date and/or Status.

Negotiated Terms

The topics in this section describe the use of SA relationship functionality for applying certain types of negotiated terms. It assumes that you are familiar with SA relationship functionality in general.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Negotiated terms are optional.** The functionality described in this section is only relevant if your organization offers this functionality.
>
> **Umbrella agreements**. If the Contract Management module is not turned off, you may also choose to use umbrella agreements to manage the functionality described here.

Imagine that the head office for a multi-site organization negotiates special terms that cover a number of its sites, each of whom have their own account and service agreement. The negotiated terms typically involve discounts. These discounts may be realized

- Under a single service agreement, separate and distinct from the service agreements that are covered, or
- Individually, for each covered service agreement. In this case, two sub-scenarios exist:
  - The discounts may be reflected on the bill segment of each service agreement, or
  - The discounts may be transferred to another service agreement (the group's service agreement). The discount does NOT appear on the individual bill segment for the service agreements that are covered by the negotiated terms.

SA relationships track and manage complex business relationships between a customer and a service provider. You can define the above relationships using service agreement relationship functionality. This is a special type of SA relationship in which the head office is the service provider.

- Examples Of Special Discounts
- Setting Up The System To Enable Negotiated Terms

Examples Of Special Discounts

The following topics illustrate examples of how your organization may configure the system to handle this business functionality.

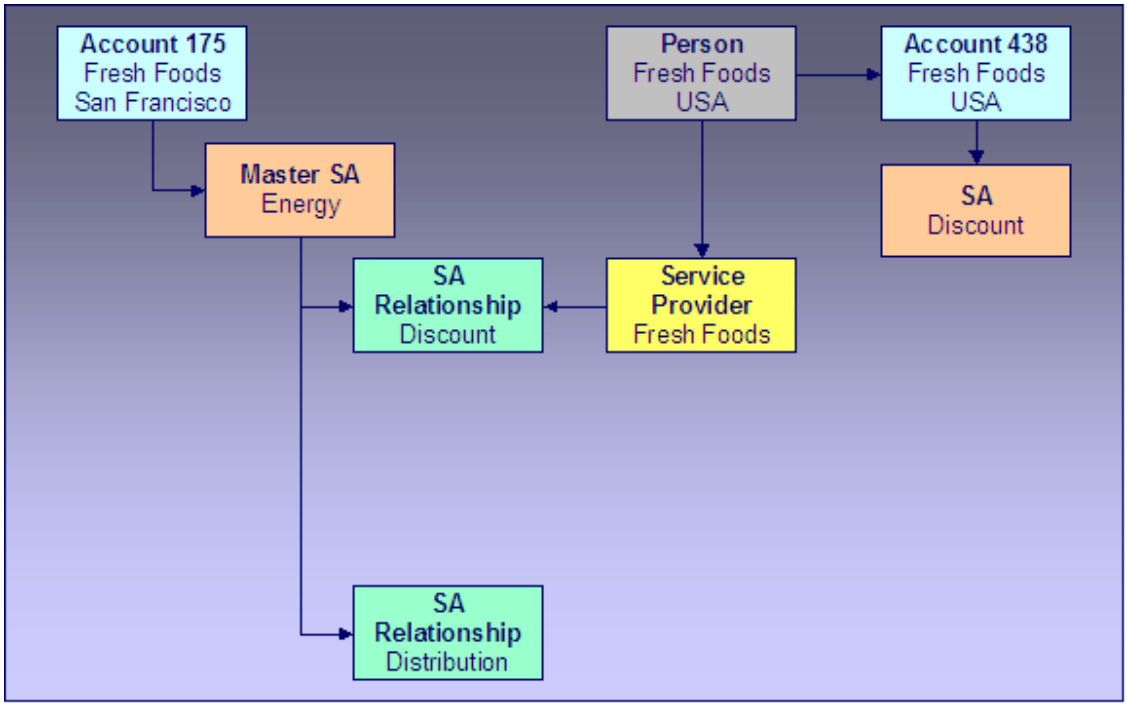
- Example Using Aggregated Consumption
- Example Using Site-by-site Discount

Example Using Aggregated Consumption

A customer, Fresh Foods has 10 stores each with its own account and electric service agreement, i.e., each store is billed separately. Fresh Food's head office decides to negotiate a group discount that applies to one or more of the individual stores.

In the following diagram, one store's account is shown.

Copyright © 2026, Oracle and/or its affiliates.



Example Using Aggregated Consumption

Note the following:

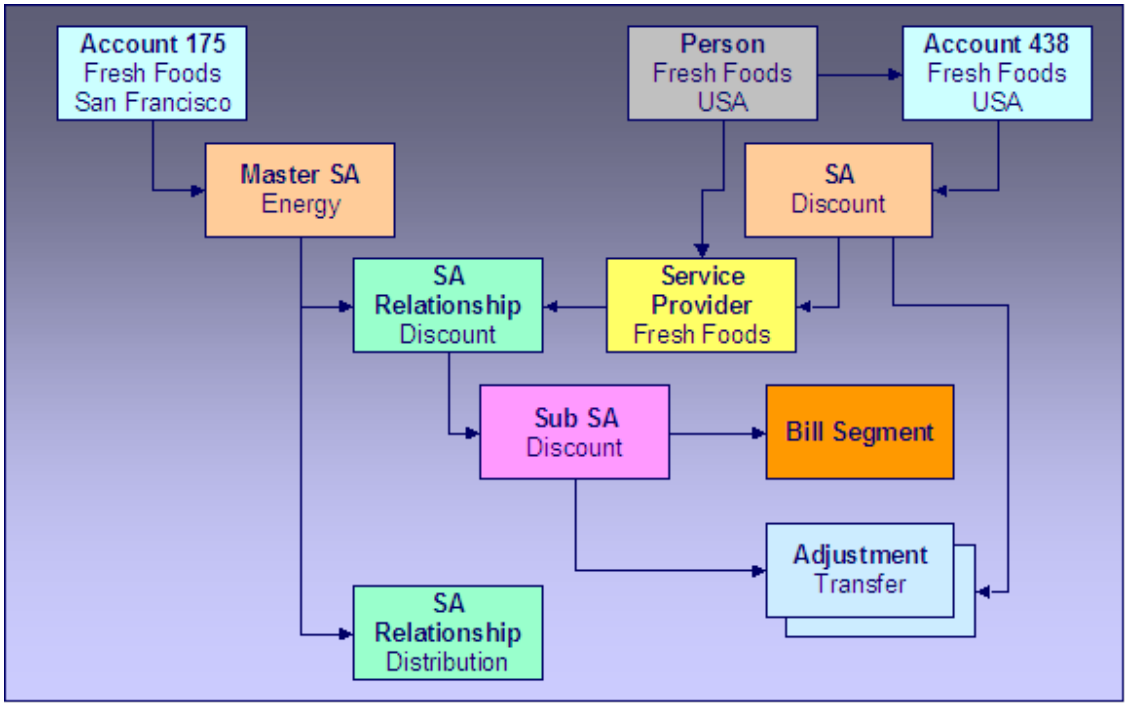
- An SA relationship is created for the energy SA covered by the negotiated terms.
- The service provider in the relationship is the head office.
- This relationship does not have a sub-SA because no additional billing services are provided for each individual covered service agreement.
- Discounts are calculated using the head office discount service agreement and affect only this service agreement and therefore only the head office's account.
- The master SA may have other deregulated relationships, such as the distribution relationship shown.

> **Note**
>
> **Aggregated consumption algorithm.** In this scenario, the service provider's service agreement amalgamates the group's consumption and applies a rate to calculate the discount. This algorithm is not provided - you get to write this algorithm to meet your needs.

Example Using Site-by-site Discount

In this scenario, the discount is calculated on a site-by-site basis.

Copyright © 2026, Oracle and/or its affiliates.



Example Using Site By Site Discount

Note the following differences between this example and the one where consumption is aggregated:

- Each service agreement participating in the discount has an SA relationship with a sub SA.
- Discounts are calculated on a site-by-site basis by the sub SA for each service agreement covered by the discount agreement.
- The discounts may be transferred to the head office service agreement using transfer adjustments. In this case, you may choose to not show the discount on the individual site's bills.

Setting Up The System To Enable Negotiated Terms

The above topics provided background information about how special negotiated terms could be supported in the system. The following discussion summarizes the various setup tasks alluded to above. These notes highlight the setup required in addition to that detailed in Setting Up SA Relationships Information.

- Algorithm
- SA Relationship Type
- Service Provider
- SA Type SA Relationship Type
- SA Type

Algorithm

Copyright © 2026, Oracle and/or its affiliates.

If you use sub SAs to calculate charges / discounts for each covered service agreement individually and you want to transfer the charges / discounts to the service provider's service agreement at bill completion, you will need to set up a bill completion algorithm to do this.

Refer to the algorithm type BCMP-TR for more information.

SA Relationship Type

You will need an SA relationship type that will be used for each type of negotiated term or discount.

Service Provider

Set up a service provider using the principal party of the negotiated discount. For example, if a head office negotiates a special agreement for its sites, you should set up a new service provider representing the head office.

- **Billing Relationship** will depend on the type of negotiated terms you choose to set up. For terms where debits / credits are calculated for each covered service agreement individually, select We Bill For Them, Rate Ready. We are effectively billing on behalf of the principal party. For terms where the discount is calculated on an amalgamated SA there is no billing relationship because no additional billing services are provided for each covered service agreement.
- Select a **Payment Relationship** of Pay SPR Not Applicable if debits / credits to the sub-SA should not be transferred to the service provider. If you want to transfer charges at bill completion time using a bill completion algorithm, you should also select Pay SPR Not Applicable because you will not be creating adjustments to pay the principal party service at payment time or at billing time.
- Enter a **Transfer A/R Algorithm** to transfer the debit / credit from the sub-SA to the principal party's service agreement. If you do not want to transfer the debit / credit to the principally party's service agreement, you do not need to specify a transfer algorithm.

SA Type SA Relationship Type

Associate the SA relationship type setup for negotiated terms with the SA type(s) of the service agreements that are covered by the terms. The following are suggested values for the SA type SA relationship type:

- **Required**. Relationships of this type should not be required for the SA type because the negotiated terms will only cover selected service agreements with that SA type.
- **Gaps in SA Relationship** are Allowed if service agreements of this type do not have to have an SA relationship of this type with a service provider throughout the effective period of the SA.
- Add the service provider set up above to the list of valid service providers for this SA type and SA relationship type combination.

Copyright © 2026, Oracle and/or its affiliates.

SA Type

Set up an SA type for the sub SAs that will be used to calculate discounts. This SA type should use the BCMP-TR bill completion algorithm if you want to transfer the discounts to the head office.

> **Note**
>
> **TBFU Deregulated Relationship with WBFT Negotiated Terms Relationship**. If you have the unusual situation where a master service agreement has a relationship with a TBFU service provider as well as a negotiated term "relationship" with a WBFT service provider, and you transfer charges / discounts from the discount agreement sub SA to the WBFT service provider, you will not be able to use the algorithms provided to transfer receivables to the service providers. This is because bill completion algorithms associated with the SA type of a master service agreement are executed before any bill completion algorithms associated with the SA types of sub service agreements. Consequently the TBFU XFER algorithm associated with the master service agreement will transfer all receivables from related sub SAs to the TBFU service provider before the BCMP-TR algorithm can transfer the discount agreement receivables. You will need to modify the TBFU XFER bill completion algorithm that transfers receivables from the customer to the TBFU service provider to exclude the discount agreement receivables.

Defining Service Credit Options

Some companies allow their customers to participate in a special rewards program. The term "service credits" is used to describe a program that rewards customers for their business. The topics in this section provide details to help you set up the control tables required to support any service credit program that your company supports.

> **Note**
>
> **Penalty Points**. The service credits functionality is described in the documentation with the assumption that it is used for accumulating points to reward your customers. However, if your company has a business need to record penalty points for a customer, the service credits functionality may be used for that purpose as well.

- The Big Picture Of Service Credits
- Designing Your Service Credit Options
- Setting Up Service Credit Options
- Service Credit Examples

Copyright © 2026, Oracle and/or its affiliates.

The Big Picture Of Service Credits

The topics in this section provide background information about service credit functionality.

- Service Credit Membership

Service Credit Membership

Let's look at some examples of special programs that may use service credits functionality:

- Capital credits - When customers receive their utility service from a cooperative, they are considered "members" of the cooperative and may over time receive capital credit allocations from the cooperative based on their service history and the cooperative's profits allocated during that time period.
- Frequent flier miles - Perhaps your company has made an agreement with one or more airlines to allow customers to accumulate frequent flier miles for every x amount spent on service.
- Free pay-per-view movies - Maybe your cable service offers free pay-per-view movies under certain conditions. Using service credits, you can set up your system to accumulate the free pay-per-view movies and use the free movies to offset actual movies viewed by the customer.
- Any other type of loyalty program where the customer earns credits that may later be redeemed in some way.

To participate in a program such as those described above, the customer is linked to a service credit membership. The membership record provides the following functionality:

- It defines the accounts that are linked to the membership.

> **NOT_SUPPORTED**
>
> Refer to Who are the Members? for information about linking persons and accounts to a membership.

- It defines a membership type, which controls certain behavior about the membership.
- It may define an external ID if the membership is associated with an external program, such as a frequent flier mile program.
- It may define a service agreement to use for miscellaneous financial transactions that may get created.
- It may define characteristics used to capture miscellaneous information about the membership.
- Over time, service credit events are created for a membership. The events indicate an amount that either adds or subtracts credit units (i.e., points, miles, movies, dollars, etc) for the membership.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Big Picture of Service Credit Membership for more information about functionality related to a membership.

- How Are Service Credits Earned?
- How Are Service Credits Redeemed?
- Service Agreements For a Membership

How Are Service Credits Earned?

Service credits may be monetary rewards for service or they may be non-monetary rewards such as free movies or frequent flier miles. In any case, how the membership earns the points or rewards depends on the business rules for the program you are offering.

A typical scenario is that the service credits are earned for a membership as a result of other services linked to the membership's accounts. For example:

- Perhaps free pay-per-view movies are earned when signing up for cable service. In this case, the pay-per-view movie membership is related to the membership account's cable service agreement. Refer to Service Credits Earned When Starting Service for more information.
- Perhaps one frequent flier mile is earned for every $10 spent on electricity. In this case, the frequent flier membership is related to the membership account's electricity service agreements (for example, electricity distribution, electricity retail and lamp service). Refer to service credits earned through billing for more information.
- For a capital credits membership, capital credit allocations are calculated based on the amount spent by the customer for standard service, for example electricity and/or gas service. In this case, the capital credit membership is related to the membership account's electricity and/or gas service agreements. For capital credits, a background process is used to calculate the allocated amounts. Refer to Allocating Capital Credit for more information.

It is also possible to earn service credits irrespective of other service for the membership's accounts. (Again, it depends on the business rules for the program you are offering.) For example, perhaps you offer 500 frequent flier miles for signing up for service with your company, regardless of the type of service chosen. In addition, assume that no additional miles are earned for ongoing service. In this example, there is no need to link the membership to any service agreements.

NOT_SUPPORTED

Refer to Service Agreements Contribute to a Membership for more information.

Each earned service credit amount is linked to the membership via a service credit event.

Copyright © 2026, Oracle and/or its affiliates.

How Are Service Credits Redeemed?

Once service credits have been earned for a membership, how may a customer redeem these credits? The method by which the credits are redeemed depends on the business rules for the program you are offering. Here are some examples of how a service credit may be redeemed:

- For free pay-per-view movies, perhaps your customer's monthly cable bill is credited for any pay-per-view movies until all free movies are used. Refer to Service Credits Redeemed Through Billing for more information.
- For frequent flier miles, the information about the earned miles is exported to the appropriate airline. The miles are actually redeemed by the customer through the airline, not through your company. Refer to Interface Membership Information to a Third Party for more information.
- For a capital credit membership, the company periodically (maybe once a year) decides if credits should be redeemed (referred to as "retired") and if so, how much. The company runs a background process to calculate the "retirement" amount. When an amount is retired, the membership balance is reduced by the retirement amount and the amount is transferred to service agreements related to the membership. Refer to Capital Credit Retirement for more information.

> **Note**
>
> **Tracking Membership Balances**. If service credits are redeemed via the system, your membership should probably be configured to keep track of a balance. Refer to Event Amounts May Contribute to a Balance for more information.

Each service credit amount redeemed through the system is linked to the membership via a service credit event.

Service Agreements For a Membership

There are three types of service agreements that may be associated with a service credit membership. The following sections describe these types.

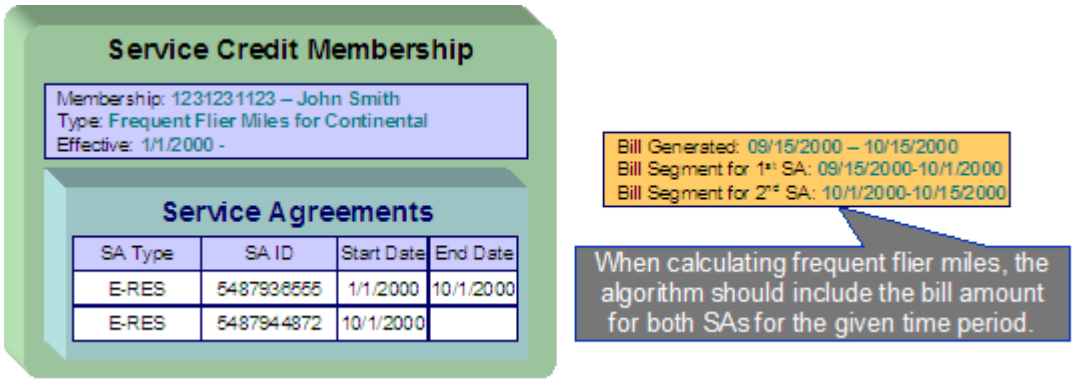
- SAs Contribute to the Membership
- SA Used for Miscellaneous Transactions
- Membership Fee SAs

SAs Contribute to the Membership

As described in how are service credits earned , many memberships are related to specific service agreements for the membership's accounts. We refer to these service agreements as the SAs that contribute to the membership because often the service credit amounts earned for the membership are based on amounts spent by the customer for these services.

Copyright © 2026, Oracle and/or its affiliates.

During the lifetime of a membership, service agreements that contribute to a membership may be stopped and other service agreements started. For example, perhaps you have a frequent flier membership that is related to electric service. Imagine that the customer starts out with a certain rate for electric service, but later decides to opt for a different type of rate that requires expiring the old service agreement and creating a new one. How does this affect your frequent flier mile calculation? It depends on how you design the algorithm that creates the frequent flier events. Essentially, the algorithm must cater for this situation. The following diagram illustrates the scenario.



The service agreements that contribute to a membership are not linked directly to the membership record. This would cause a maintenance burden, requiring links to be updated when service is stopped or started for applicable service agreements. Rather, this link is indirect. The list of service agreement types is defined for the membership type. The system can determine which service agreements are "linked" to the membership by looking at the SA types for the membership type.

NOT_SUPPORTED

Refer to Determine The Types of Service Agreements That Contribute to the Membership for more information on designing your membership type to include appropriate SA types.

SA Used for Miscellaneous Transactions

For some memberships, you may need to define a special service agreement to use for miscellaneous transactions. For example:

If events created for a membership cause an adjustment to be created to affect the general ledger, rather than allowing the system to arbitrarily pick a service agreement to use for this adjustment, the SA to use should be indicated on the membership. Refer to An Event May Cause Other Actions to Occur for more information.

Membership Fee SAs

For some memberships, a membership fee may be applicable. A special service agreement is used to hold the fee. This service agreement is not linked directly to the membership, but is simply a service agreement linked to one of the membership's accounts.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Some fees may be refundable.** The refunding of a fee must be handled by an algorithm. Refer to SAST-RF for information about the algorithm type provided with the base product.

Designing Your Service Credit Options

This section helps you to determine how to design your service credit membership types and service credit event types.

- Designing Your Membership Types
- Designing Your Service Credit Event Types
- How Are Service Credit Events Created?

Designing Your Membership Types

This section discusses the options to consider when designing your service credit membership types.

First consider the type of unit that your membership's service credit events represent.

- Is it related to a currency? If so, ensure that your currency is correctly defined on the currency page.
- Is it a non-currency unit, such as movies, points or miles? If so, you need to define the unit on the credit unit page.

Next, consider other behavior that your membership may exhibit.

- Should your membership calculate an overall balance ?
- Will miscellaneous financial transactions be created for you membership over its lifetime? If so you may need to link a service agreement to your membership.
- Should events linked to your membership reference a fiscal year ?

The table below illustrates three types of memberships: one for capital credits, one for frequent flier miles and one for free pay-per-view movies.

Membership Type	Description	Unit Type / Currency or Credit Unit	Has Balance?	Require SA?	Fiscal Year?
STDCAPCR	Standard capital credit membership	Currency / $	Yes	Yes	Yes
FFDELTA	Delta frequent flier miles	Credit Unit / Miles	No	No	No
FREEPPV	Free pay-per-view movies	Credit Unit / Movies	Yes	No	No

Copyright © 2026, Oracle and/or its affiliates.

The following points explain the settings for each membership above:

- The capital credit membership uses a currency of US dollars for its units. Over time, capital credits are allocated and retired. The overall balance of the credits and debits should be calculated and displayed for information purposes. It is common for the allocation and retirement of capital credits to affect the GL, and as a result, a membership SA is required for posting these financial effects. Finally, in a capital credit situation, the allocation is typically related to a specific fiscal year. When calculating and displaying balances, the balance for each fiscal year must also be available. As a result, the fiscal year must be set to required.
- For the frequent flier membership type, a separate membership type must be created for each different airline. This is because a separate membership must exist for each separate airline in order to keep track of the accumulated miles correctly. The membership type does not have a balance because the accumulated miles are interfaced to the airline and the airline keeps track of the balance. In this example, no SA is required because an assumption is being made that the creation of frequent flier miles does not affect the GL. (Accumulating miles is no liability or expense for the company. The miles are simply accumulated on behalf of a third party.) Finally, the frequent flier miles do not need to indicate a fiscal year.
- For the free pay-per-view movies membership, we assume that the credits are redeemed from within the system. For example, perhaps a calculation rule calculation algorithm or pre-processing calculation rule redeems the free pay-per-view movies over time as customers are billed for the movies. As a result, the membership should have a balance. In this example, no SA is required because an assumption is being made that any affect on the GL is posted at the time the free movies are redeemed (i.e., when calculating a bill). (This is just an example. It's possible that you may want to post to the GL when free movies are accumulated to mark a payable for the company.) Finally, free pay-per-view movies do not need to indicate a fiscal year.

More options must be considered for each membership type.

- Consider Whether the Membership Should Indicate Subcategories
- Determine the Types of Service Agreements Linked to the Membership
- Consider Special Functionality Needed When Adding, Activating or Inactivating a Membership
- Determine Whether The Membership Requires Additional Information

Consider Whether the Membership Should Indicate Subcategories

For some types of memberships, the amount of each service credit event is further grouped by a subcategory. Refer to Events May Indicate a Subcategory for more information.

Use subcategories if multiple subtypes of credits may be accumulated and redeemed and if balances need to be tracked for each subcategory.

For our sample membership types, only the capital credits type should indicate subcategories.

Membership Type	Description	Subcategories
STDCAPCR	Standard capital credit membership	Distribution Transportation
FFDELTA	Delta frequent flier miles	
FREEPPV	Free pay-per-view movies	

Copyright © 2026, Oracle and/or its affiliates.

Determine the Types of Service Agreements Linked to the Membership

Memberships typically exist to reward customers for participating in standard service with the company. Refer to How Are Service Credits Earned? for more information. If memberships of this type are related to standard service for your company, determine the SA types for these service agreements.

> **Note**
>
> **Standard Service SA Types**. These service agreement types are probably different from the type of service agreement you may link as the membership SA. The membership SA is used for miscellaneous charges like general ledger posting. These SA types are related to electricity service, gas service, etc.

For our three examples, assume that the company provides electric service and cable. The capital credits are related only to electric service, frequent flier miles are accumulated for a combination of services, and free pay-per-view movies are related only to cable service.

Membership Type	Description	SA Types
STDCAPCR	Standard capital credit membership	E-RES (Electric Residential) AL-RES (Area Lighting)
FFDELTA	Delta frequent flier miles	E-RES (Electric Residential) AL-RES (Area Lighting) CABLE - RES (Cable Residential)
FREEPPV	Free pay-per-view movies	CABLE-RES (Cable Residential)

Consider Special Functionality Needed When Adding, Activating or Inactivating a Membership

Should a letter or bill message be generated when a membership is created or activated? Should anything happen when a membership is inactivated? If so, you may need to define one or more algorithms to be plugged in on the membership type.

For our examples, let's say that a bill message is generated when a frequent flier or free pay-per-view membership is created. Let's say that a letter is generated when a capital credits membership is activated. We also assume that when a capital credits membership becomes inactive, any outstanding balance is redeemed. For the frequent flier miles and pay-per-view memberships, we assume nothing special occurs when the membership becomes inactive.

> **NOT_SUPPORTED**
>
> Refer to Lifecycle of a Membership for more information about the various status values for a membership.

Copyright © 2026, Oracle and/or its affiliates.

Membership Type	Description	Algorithm System Event	Algorithm
STDCAPCR	Standard capital credit membership	Membership Activation	Send Letter
		Membership Inactivation	Redeem Balances
FFDELTA	Delta frequent flier miles	Membership Creation	Generate Bill Message
FREEPPV	Free pay-per-view movies	Membership Creation	Generate Bill Message

> **Note**
>
> **Sample Algorithms**. The base package does not provide sample algorithms for all the above examples. Refer to Service Credit Membership Type - Algorithm for more information about the algorithms provided with the system.

Determine Whether The Membership Requires Additional Information

Is there any information about your membership that you need to capture that is not already provided by the base system logic? If the answer is yes, you may need to define characteristics for your membership. Use the characteristic collection on the membership type to define the types of characteristics allowed for memberships of this type. You may also define default values for you membership's characteristics.

Designing Your Service Credit Event Types

Now that you have designed your membership types, you need to design the types of service credit events that may be created for your membership.

In many cases, a credit event should cause additional functionality to occur. Algorithms are executed when an event is completed and when an event is canceled and are used to perform additional functionality. The following points illustrate possible algorithms that may be needed when a credit event is completed.

- Validate the event as compared to other events. For example, perhaps a new event should never cause the overall membership balance to fall below zero. You could use an algorithm on the service credit event type to check this condition.
- Create an adjustment that posts to the general ledger. Your credit event may not affect the customer's balance when created, but perhaps it should have an effect on the general ledger. For these types of credit events, you may need to create an adjustment to post to the GL.
- Create adjustments to affect the customer's balance. When "redeeming" a credit, you may need to transfer a monetary amount to one or more of the customer's service agreements.

Copyright © 2026, Oracle and/or its affiliates.

- Create a bill message. Perhaps when a credit is accumulated, you want to inform the customer via a message on the bill indicating the credit amount. A temporary bill message is added to one of the membership's accounts.

> **Note**
>
> **One Account Message of the Same Message Type**. The temporary bill message collection on the account allows only one bill message of the same bill message type. If it's possible for multiple types of events to be generated for the same account, consider creating a different bill message type for each event type.

- Stamp a batch code and batch run number onto the event. This would be used when your event information needs to be interfaced to an external system.

The following points illustrate possible algorithms that may be needed when a credit event is canceled.

- Validate the event as compared to other events. For example, perhaps canceling an event should never cause the overall membership balance to fall below zero.
- Cancel adjustments that may have been created when the event was completed.

To illustrate examples of when to use some of the algorithms above, we'll design event types for the membership types designed above.

- Designing Capital Credit Event Types
- Designing Frequent Flier Event Types
- Designing Pay-per-view Event Types

Designing Capital Credit Event Types

The following event types illustrate typical events for a capital credits membership.

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ALLOCATECCR	Capital credit allocation	STDCAPCR	Event Creation	Create Simple Adjustment
			Event Creation	Generate Bill Message
			Event Cancellation	Validate Balance Not < Zero
			Event Cancellation	Cancel Related Adjustments
RETIRECCR	Retire capital credit (apply to customer's balance)	STDCAPCR	Event Creation	Validate Balance Not < Zero
			Event Creation	Create Adjustments to Affect Customer's Balance
			Event Cancellation	Cancel Related Adjustments

Copyright © 2026, Oracle and/or its affiliates.

FORFEITCCR	Forfeit capital credit (retirement not applied to customer's balance)	STDCAPCR	Event Creation	Validate Balance Not < Zero
			Event Creation	Create Simple Adjustment (affect GL only)
			Event Cancellation	Validate Balance Not < Zero
			Event Cancellation	Cancel Related Adjustments

These event types assume the following:

- When credit events are allocated, the customer is notified via a bill message, the general ledger is affected so a simple GL only adjustment is created. If an event of this type is canceled, any adjustments that were created should be canceled. The event amount should always be positive, so checking that the membership balance does not fall below zero is only checked for event cancellation.
- When capital credits are retired, it's possible that the full membership balance is not applied to the customer's balance. For the portion of the retirement that does affect the customer's balance, you need an algorithm that applies the credits to the customer's service agreements via adjustments. Cancellation of this event should cause any related adjustments to be canceled. It's assumed that the amount of this event is a credit so checking that the membership balance does not fall below zero is only checked for event completion.
- For the portion of the retirement that is not applied to the customer's balance, the event amount should simply affect the GL so a GL only adjustment is created. Cancellation of this event should cause any related adjustments to be canceled. It's assumed that the amount of this event is a credit so checking that the membership balance does not fall below zero is only checked for event completion.

> **Note**
>
> **Service credit event types are independent of subcategories.** A capital credits membership typically uses subcategories. When events are created for different subcategories, the same service credit event type may be used. As a result, all subcategories use the same event completion and event cancellation algorithms.

Designing Frequent Flier Event Types

The following event types illustrate typical events for a frequent flier membership:

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ADDMILES	Add miles to the membership	FFDELTA	Event Creation	Generate Bill Message
			Event Creation	Populate Batch Information

Copyright © 2026, Oracle and/or its affiliates.

	Event Cancellation	Event may not be canceled.

This event type assume the following:

- A new event should generate a bill message.
- Information about the event amount should be interfaced to an external system so batch information should be populated when the event is completed.
- Events of this type may not be canceled because the information is interfaced to an external system. Rather, to reverse an event, simply create a new event whose amount is a credit. This credit amount is also interfaced to an external system.

> **Note**
>
> **Sample Algorithm.** The system does not provide a sample cancellation algorithm that prevents the event from being canceled.

- There is no validation to ensure that the balance does not fall below zero. Recall that this membership was defined as not requiring a balance.

Notice that only one type of event has been defined for this membership. That is because the credits for this membership are not redeemed via this system. Rather they are accumulated on behalf of an external system.

Designing Pay-per-view Event Types

The following event types illustrate typical events for a free pay-per-view movies membership:

SC Event Type	Description	Membership Type	Algorithm System Event	Algorithm
ADDPPV	Free pay-per-view movies	FREEPPV	Event Creation	Generate Bill Message
			Event Cancellation	Validate Balance Not < Zero
REDEEMPPV	Redeem pay-per-view movies	FREEPPV	Event Creation	Validate Balance Not < Zero
			Event Creation	Generate Bill Message

These event types assume the following:

- Any type of new event should generate a bill message.
- For adding free movies, it is assumed that the quantity is positive so when the event is cancelled, the algorithm verifies that the balance does not become negative.
- For redeeming free movies, it is assumed that the quantity is negative so when the event is completed, the algorithm verifies that the balance does not become negative.
- These events do not affect the general ledger and they do not directly affect the customer's balance so no adjustment algorithm is needed.

Copyright © 2026, Oracle and/or its affiliates.

How Are Service Credit Events Created?

Now that you have designed the behavior of your service credit events, an important issue is to determine how these events are created. Consider your business practice for each type of membership.

Let's use our sample memberships to work through different ways that you may create service credits.

- Service Credits for Capital Credit Memberships
- Service Credits for Frequent Flier Memberships
- Service Credits for Free Movies Memberships

Service Credits for Capital Credit Memberships

For a capital credit membership, capital credit allocations are calculated once a year based on billing history and the cost of service. To accomplish this, a background process calculates the amount and creates the service credits. Refer to allocating capital credits for a sample process provided with the base package.

Credits are redeemed via the retirement process. The company determines when to retire capital credits based on analysis of their financial situation. This retirement process is also handled by a background process. Refer to capital credit retirement for a process provided with the base package.

For capital credits memberships, special functionality is required when a member dies. The capital credits are considered part of the person's estate and may need to be retired and applied to a beneficiary's account. This process depends on the company's business practice. However, typically, the membership status would change to inactive so that new capital credit allocations are not created for the membership. The system provides a sample inactivation algorithm called SCMI-RB that transfers part or all of the outstanding credit balance to the member's service agreements. From there, a user can cut a check to the beneficiary.

> **Note**
>
> **Some Credits Are Never Retired**. For many cooperatives, some types of allocated credits are never retired. Refer to Partial Retirement for more information.

Service Credits for Frequent Flier Memberships

In our example, frequent flier miles were related to both the electricity and phone service. Let's assume that frequent flier miles are accumulated every $x spent on electricity and phone service. In this example, a bill completion algorithm creates service credits based on the bill segment amounts for these service agreements. Refer to Service Credits Earned Through Billing for more information.

Copyright © 2026, Oracle and/or its affiliates.

As mentioned before, the frequent flier miles are not redeemed using this system, but are interfaced to a third party for redemption.

Service Credits for Free Movies Memberships

> **Note**
>
> **Sample Algorithms**. No base package algorithm types are provided to support the logic described in this example.

In this scenario, let's suppose that a customer receives three free pay-per-view movies when signing up for new service. To handle this, perhaps an SA creation algorithm creates the membership and the service credits when the service is started. Or perhaps you want to wait until the first bill is generated and a bill completion algorithm is used to generate the first credit. It depends on your business practice.

For redeeming the free pay-per-view movies, it is assumed that the movies are credited during billing after it is determined that the customer has been billed for a movie. The number of movies used for the membership is reduced until all the free movies are used.

> **NOT_SUPPORTED**
>
> Refer to Service Credits Redeemed Through Billing for more information.

Setting Up Service Credit Options

- Setting Up Credit Units
- Setting Up SC Membership Inactive Reasons
- Setting Up Service Credit Membership Types
- Setting Up Service Credit Event Types

Setting Up Credit Units

Credit unit is used for service credit membership types whose events record non-monetary units. Open **Admin**, and then **Service Credit**, and then **Credit Unit** to set up credit units.

Description of Page

The following fields display for each credit unit:

Credit Unit The unique identifier of the credit unit.

Copyright © 2026, Oracle and/or its affiliates.

Description The description of the unit. This also acts as a label for the unit when displaying information about a service credit event.

Symbol / Label Position Indicates whether or not the label for this credit unit appears as a Prefix or as a Suffix to the service credit event amount.

Decimal Positions Indicates the number of decimal positions used for this credit unit. This information should be used by any algorithm or background process that creates a service credit event to determine how to store the event amount. It is also use to correctly display the service credit amounts.

Setting Up SC Membership Inactive Reasons

The service credit membership inactive reason must be specified when the status of a service credit membership changes to inactive. Open **Admin**, and then **Service Credit**, and then **Service Credit Membership Inactive Reason** to set up service credit inactive reasons.

Description of Page

The following fields display for each inactive reason:

Inactive Reason The unique identifier of the service credit membership inactive reason.

Description The description of the inactive reason.

Setting Up Service Credit Membership Types

- SC Membership Type - Main
- SC Membership Type - Algorithm
- SC Membership Type - Characteristics

SC Membership Type - Main

Service credit memberships have a membership type. Open **Admin**, and then **Service Credit**, and then **Service Credit Membership Type**, and then **Add** to define the membership type.

Description of Page

Enter a unique **Service Credit Membership Type** and **Description** for each membership type.

Use the **SA Requirement** flag to indicate whether a miscellaneous SA must be linked to memberships of this type. The possible values are SA Required and SA Not Allowed .

> **Note**
>
> The value defaults to SA Required.

Copyright © 2026, Oracle and/or its affiliates.

Use the **Fiscal Year Requirement** flag to indicate whether events linked to memberships of this type should indicate a fiscal year. The possible values are Fiscal Year Required and Fiscal Year Not Allowed.

> **Note**
>
> **Default Note**. The value defaults to Fiscal Year Not Allowed.

Use the **SCM Event Balance** flag to indicate whether or not a balance of service credit events linked to memberships of this type should be calculated and displayed. The possible values are Has a Balance and No Balance.

> **Note**
>
> **Default Note**. The value defaults to Has a Balance.

Use the **SC Membership Type Unit** flag to indicate whether or not the unit for the event amounts for events linked to memberships of this type is Currency or Credit Unit. When the unit is currency, indicate the **Currency Code**. When the unit is credit unit, indicate the **Credit Unit**.

If events linked to memberships of this type must indicate a subcategory, enter a valid **Subcategory Name** and **Description** for each subcategory applicable to this membership.

> **NOT_SUPPORTED**
>
> Refer to Events May Indicate a Subcategory for more information.

Use the **SA Types** grid to indicate the types of service agreements that contribute to memberships of this type. Indicate the **CIS Division** and **SA Type** for each type of service agreement that is related to a membership of this type.

SC Membership Type - Algorithm

Open **Admin**, and then **Service Credit**, and then **Service Credit Membership Type**, and then **Search** and navigate to the **Algorithm** page to define any algorithms that are associated with a membership type.

Description of Page

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (descriptions of all possible events are provided below).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

Copyright © 2026, Oracle and/or its affiliates.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description
Membership Creation	Optional	Use this system event for algorithms that are executed when a membership is created.

Copyright © 2026, Oracle and/or its affiliates.

Membership Activation	Optional	Use this system event for algorithms that are executed when a membership becomes active.

> **Note**
>
> **Creating Memberships In Active Status**. A membership may be created in either pending status or in active status, based on your business practice. For memberships created in active status, the system executes the

Copyright © 2026, Oracle and/or its affiliates.

		membership creation algorithms and the membership activation algorithms.
Membership Inactivation	Optional	Use a system event of Membership Inactivation for algorithms that are executed when a membership becomes inactive .

SC Membership Type - Characteristics

To define characteristics that can be defined for service credit memberships of a given type, open **Admin**, and then **Service Credit**, and then **Service Credit Membership Type**, and then **Search** and navigate to the **Characteristics** page.

Description of Page

Use the characteristics collection to define characteristics that can be defined for service credit memberships of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on service credit memberships of a given type. Turn on the **Default** switch to default the **Characteristic Type** when service credit memberships of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

Setting Up Service Credit Event Types

Service credit events created for a service credit membership have an event type. Open **Admin**, and then **Service Credit Event Type**, and then **Add** to set up service credit event types.

Description of Page

Enter a unique **Service Credit Event Type** and **Description** for each event type.

Indicate the **Service Credit Membership Type** to which this event type belongs.

Copyright © 2026, Oracle and/or its affiliates.

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (descriptions of all possible events are provided below).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** for which you can define algorithms.

System Event	Optional / Required	Description
Event Completion	Optional	Use a system event of Event Completion for algorithms that are executed when an event is completed. Refer to An Event May Cause Other Actions to Occur for more information.
Event Cancellation	Optional	Use this system event for algorithms that are executed when an event is canceled.

Service Credit Examples

In this section, we provide examples of how to define your control tables to support functionality related to different types of service credit memberships. While your company may not define your environment exactly the same way, this section should help solidify your understanding of how to set up your company's service credit options.

- Defining Control Tables for a Refundable Fee
- Defining Control Tables for a Nonrefundable Fee
- Defining an SA Type for Miscellaneous Transactions
- Using Campaigns/Packages to Set Up Membership
- Defining Another Person for Your Account
- Service Credits Earned When Starting Service
- Service Credits Earned Through Billing
- Service Credits Redeemed Through Billing
- Designing Your Rate Options for Capital Credits
- Partial Retirement
- Interface Membership Information to a Third Party

Defining Control Tables for a Refundable Fee

If your membership requires payment of a fee that is refundable, you must define an SA type to use for the refundable fee.

- Adjustment Type for Refundable Fee

Copyright © 2026, Oracle and/or its affiliates.

- SA Type for Refundable Fee
- Start Option for Refundable Fee
- Algorithm for Refunding the Fee

Adjustment Type for Refundable Fee

This example assumes that the fee is set up similarly to a deposit. The adjustment used to charge the fee would affect the current balance, but not the payoff balance or the general ledger because the fee is not considered a "receivable", rather it is an amount collected and held for the customer.

Create an adjustment type for levying the fee. Indicate the fee amount and indicate the ADJT-CA adjustment FT creation algorithm, which affects the current balance, but not the payoff balance or the general ledger.

> **Note**
>
> **Adjustment Type Profiles**. Be sure to add this adjustment type to an appropriate adjustment type profile and ensure this profile is linked to your SA type.

SA Type for Refundable Fee

Create an SA type to use for the fee. This SA type should be marked as not billable.

Indicate the appropriate adjustment type profile that includes the adjustment type to levy the fee.

Indicate an appropriate payment segment type that references the PSEG-NM payment segment FT algorithm. This algorithm affects the payoff amount and current amount by the payment amount, which should cause the current amount to become zero and the payoff amount to become a credit for the fee amount when the fee is paid.

Start Option for Refundable Fee

For this SA type, define a start option that causes an adjustment to be levied as part of the start service process.

> **NOT_SUPPORTED**
>
> Refer to Including The Membership Fee on campaigns and packages for more information about levying the fee via the order transaction.

Algorithm for Refunding the Fee

Copyright © 2026, Oracle and/or its affiliates.

The base product provides an algorithm type SAST-RF that refunds a fee when service agreements that contribute to the membership are stopped. You must create an algorithm for this algorithm type and enter the SA type created above as an input parameter.

This SA stop algorithm must be plugged in on all SA types that you defined for the membership type.

Defining Control Tables for a Nonrefundable Fee

If your membership requires payment of a fee that is not refundable, you can set this up in two ways:

- You can create a special SA type to handle charging the fee. For this SA type, define a start option that causes an adjustment to be levied as part of the start service process. This adjustment contains your fee. You may use this option if the membership is related to multiple types of services and not all services need to be present in order to create the membership.
- You can levy an adjustment on one of the other service agreements that is being started. To do this, you would use a start option to define an adjustment to be levied as part of the start service process. You may use this option if the membership is related to a single service (for example, free pay-per-view movies).

For our example, we will set up the SA for the nonrefundable fee with a separate 'fee' SA type.

- Adjustment Type for Nonrefundable Fee
- SA Type for Nonrefundable Fee
- Start Option for Nonrefundable Fee

Adjustment Type for Nonrefundable Fee

Create an adjustment type for levying the fee. Indicate the fee amount and indicate the ADJT-NM adjustment FT creation algorithm, which affects the current balance and payoff balance by the adjustment amount, and affects the general ledger.

> **Note**
>
> **Adjustment Type Profiles**. Be sure to add this adjustment type to an appropriate adjustment type profile and ensure this profile is linked to your SA type.

SA Type for Nonrefundable Fee

Create an SA type to use for the fee. This SA type should be marked as not billable.

Indicate the appropriate adjustment type profile that includes the adjustment type to levy the fee.

Indicate an appropriate payment segment type that references the PSEG-NM payment segment FT algorithm. This algorithm affects the payoff amount and current amount by the

Copyright © 2026, Oracle and/or its affiliates.

payment amount, which causes the current amount to become zero and the payoff amount to become a credit for the fee amount when the fee is paid.

Start Option for Nonrefundable Fee

For this SA type, define a start option that causes an adjustment to be levied as part of the start service process.

> **NOT_SUPPORTED**
>
> Refer to Including The Membership Fee on campaigns and packages for more information about levying the fee via the order transaction.

Defining an SA Type for Miscellaneous Transactions

Does your membership require a service agreement to support miscellaneous transactions? If so, you need to consider the SA type to use for this service agreement. This SA type should be marked as not billable.

If you choose to use the order transaction to set up the membership, this SA type must be defined on the algorithm that creates the membership.

It is possible that you may require a start option for this SA type, for example if you want to define a characteristic for service agreements of this type.

Using Campaigns/Packages to Set Up Membership

If enrollment in a membership is a common occurrence for your customers when starting service, you should consider using the order page to start service for the customer and to create the membership as well.

> **Note**
>
> This section only makes sense if you are familiar with the Sales and Marketing chapter.

The recommendation is to use a question/miscellaneous field to ask the customer service representative to indicate the appropriate membership type. Algorithms validate this membership type and use it to create a membership of that type.

This section walks you through how to set up the campaign / package required to support this.

- Column Reference for Membership Type
- Algorithms to Create Membership via Order
- Define a Campaign for Creating a Membership When Starting Service

Copyright © 2026, Oracle and/or its affiliates.

- Including the Membership Fee

Column Reference for Membership Type

In order to ask the customer service representative to define an appropriate service credit membership type, you must define a column reference for the membership type.

Add a new column reference with the following information:

- Column Reference: SCM-TYPE
- Description: SC Membership Type
- FK Reference: SCM TYPE
- Long Description: Service Credit membership type to use when creating a service credit membership at start time.

> **Note**
>
> **Column Reference Algorithms**. This column reference indicates a validation algorithm and a posting algorithm. However, we have not defined them yet so for now simply save this information.

Algorithms to Create Membership via Order

The base product provides two algorithm types to support the creation of a membership record via the order page: a validation algorithm type and a posting algorithm type.

Validate Membership Information

This algorithm is a column reference validation algorithm that checks that an input membership type is valid. Refer to CRVL-ME for more information. You must define an appropriate algorithm for this algorithm type, and on that algorithm you must define the column reference used to identify the membership type. For example:

- Algorithm: VAL MEM TYPE
- Description: Validate SC Membership Type
- Algorithm Type: CRVL-ME
- Parameter1: (Column Reference for Membership Type): SCM-TYPE

Post Membership Information

This algorithm is a column reference posting algorithm that creates a membership using the membership type indicated by the user. Refer to CRPS-ME for more information. You must define an appropriate algorithm for this algorithm type, and on that algorithm you must define the column reference used to identify the membership type. For example:

- Algorithm: CREATE MEMBRSHP
- Description: Validate SC Membership Type
- Algorithm Type: CRPS-ME

Copyright © 2026, Oracle and/or its affiliates.

- Parameter1: (Column Reference for Membership Type): SCM-TYPE
- Parameter2: (Service Credit Membership Status): 10 -Pending or 20 -Active. Refer to Lifecycle of a Membership for more information.
- Parameters3-5: (Division, SA Type, Start Option): Indicate the information needed to create an SA for miscellaneous transactions.

> **Note**
>
> This algorithm first looks for an existing service agreement of this division / SA type linked to the membership's accounts. If one exists, it uses that SA to link to the membership. If an SA does not exist, it creates one with the input division, SA and (optional) start option.
>
> **Update the Column Reference.** Now that you have defined the validation and posting algorithms, return to your column reference for membership type and define the algorithms there.

Although your company may support multiple types of memberships, this column reference and its algorithms have been designed such that only one column reference for membership type would be needed to set up any type of membership. Although the posting algorithm for the membership type column reference indicates an SA type to use for miscellaneous transactions, a service agreement is only created and linked if your membership type indicates that an SA is required. As a result, you may use the same column reference for both memberships that require an SA and those that don't require an SA. However, if you have different membership types that require an SA and each uses a different SA type or a different start option for the membership SA, you need to define more than one posting algorithm and, as a result, more than one column reference.

Define a Campaign for Creating a Membership When Starting Service

Many factors must be considered when designing your campaigns and packages. The possible creation of a membership when using the order page is simply another factor to consider.

If you plan to define a question/miscellaneous field to capture a service credit membership type, the available packages linked to the campaign should be ones that are related to memberships. For example, if your membership is related to electric service, it doesn't make sense to create a membership for a campaign designed to generate a one-time charge.

When defining a question/miscellaneous field, you must indicate its applicability. Consider whether a membership type is required, optional or only applicable for certain packages. This helps ensure that your users capture this information when appropriate.

Including the Membership Fee

You must determine the best way to setup your campaign/package to handle the levying of your membership fee, if applicable. Consider some of the following questions.

- If the membership has a fee, is it a refundable fee or a non-refundable fee?

Copyright © 2026, Oracle and/or its affiliates.

- Is the fee always applicable for the membership? Or is it waived under certain conditions? For example, maybe the fee is applicable if a customer signs up for a single service, but the fee is waived if the customer signs up for two or more services.

The SCM type column reference and question/miscellaneous field are set up to ask the user what type of membership to create. Because the applicability of the fee may differ for each membership type, you should carefully consider the campaign / package setup to levy the fee correctly.

- If a fee is always applicable for a membership, you may consider creating a membership creation algorithm that creates a fee SA with a start option to levy the fee when the membership is created.

> **Note**
>
> **Sample Algorithm**. The base product does not provide a membership creation algorithm to do this.

- If the fee is not always applicable for a membership, you must determine when the fee is applicable. The recommendation is to include the fee SA type in the SAs-to-create collection for the appropriate package. For example,
  - If the fee is applicable when the customer signs up for a single service, you should define a package for each single service that includes one SA to create for the single service and the fee SA as another SA to create.
  - If the fee is waived when the customer signs up for two or more services, you should define a package for each combination of the multiple services. These packages do not include the fee SA in the SAs to create.
- If the customer signing up for service is a former customer who has returned, perhaps the fee was paid earlier when the customer originally had service. In this case, maybe your business practice is to waive the fee at this time. To do this, you should set up a question/ miscellaneous field for the user to mark that this is a returning customer. Based on the answer to this question, perhaps only packages that do not levy any fee are eligible for selection.

Defining Another Person for Your Account

It is common for a capital credit membership to define more than one person for the account being turned on and linked to a membership. A typical example is a married couple. Both spouses are indicated on the account and financially responsible persons and as a result, both are considered members.

Column reference algorithms have been provided by the base product to enable linking a second person to your account via the order page.

This section walks you through how to set up the column reference and campaign to support this.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Linking A Second Person to the Account.** This logic is not restricted to service credit functionality. Any campaign may be designed to include the ability to link a second person to the account being started.

- Column References for Additional Person
- Algorithms to Link Additional Person via Order
- Design a Campaign to Include Linking an Additional Person

Column References for Additional Person

In order to ask the customer service representative to link an additional person for the account, you must define several column references to use as questions/miscellaneous fields.

- We should assume that the person may already exist in the database. As a result, a question to record the person ID is needed.
- If the person does not already exist, the user should capture the person's name, an ID type and an ID number. Questions for these three fields are needed. The system uses this information to create a new person.
- Whether we are using an existing person or creating a new one, the person's link to the account must include an account relationship type. A question to record the account relationship type is needed.

Add a new column reference for person ID:

- Column Reference: PERSON-ID
- Description: Person ID
- FK Reference: PER
- Long Description: Person ID of the additional person to link to the order's account.

Add a new column reference for person name:

- Column Reference: PERSON-NAME
- Description: Person Name
- FK Reference: not applicable
- Long Description: Name of a new person to create.

Add a new column reference for identifier type:

- Column Reference: PER-ID-TYPE
- Description: Person Identifier Type
- FK Reference: ID TYPE
- Long Description: Person Identifier type for the primary ID of the additional person to link to the order's account.

Add a new column reference for identifier number:

- Column Reference: PER-ID-NUM

Copyright © 2026, Oracle and/or its affiliates.

- Description: Person ID Number
- FK Reference: not applicable
- Long Description: Identifier number for the primary ID of the additional person to link to the order's account.

Add a new column reference for account relationship type:

- Column Reference: ACCT-REL-TYPE
- Description: Account Relationship Type
- FK Reference: ACCT REL
- Long Description: Relationship type to use for the link between the additional person and the order's account.

> **Note**
>
> **Column Reference Algorithms**. One of the column references above must indicate a validation algorithm and a posting algorithm. However, we have not defined them yet so for now simply save this information. We recommend using the account relationship type record because that is used for all additional persons.

Algorithms to Link Additional Person via Order

The base product has provided two algorithm types to support the linking of an additional person to the account via the order page: a validation algorithm type and a posting algorithm type.

Validate Addition Person Information

This algorithm is a column reference validation algorithm which checks that either a person id or person name, a valid ID type and ID number are provided and that a valid account relationship type is provided. Refer to CRVL-PE for more information. You must define an appropriate algorithm for this algorithm type and on that algorithm you must define the column reference used for the five fields required for this validation. For example:

- Algorithm: VAL ADD PER
- Description: Validate Additional Person Info
- Algorithm Type: CRVL-PE
- Parameter1: (Column Reference for Person ID): PERSON-ID
- Parameter2: (Column Reference for Person Name): PERSON-NAME
- Parameter3: (Column Reference for Person ID Type): PER-ID-TYPE
- Parameter4: (Column Reference for Person ID Number): PER-ID-NUM
- Parameter5: (Column Reference for Account Relationship Type): ACCT-REL-TYPE

Post Addition Person

This algorithm is a column reference posting algorithm that may link an existing person to the order's account or create a new person and link that person to the order's account. Refer to CRPS-PE for more information. You must define an appropriate algorithm for this algorithm

Copyright © 2026, Oracle and/or its affiliates.

type, and on that algorithm you must define the column reference used for the five fields required for this logic. For example:

- Algorithm: LINK ADDNTL PER
- Description: Link additional person to order's account
- Algorithm Type: CRPS-PE
- Parameter1: (Column Reference for Person ID): PERSON-ID
- Parameter2: (Column Reference for Person Name): PERSON-NAME
- Parameter3: (Column Reference for Person ID Type): PER-ID-TYPE
- Parameter4: (Column Reference for Person ID Number): PER-ID-NUM
- Parameter5: (Column Reference for Account Relationship Type): ACCT-REL-TYPE

> **Note**
>
> **Update the Column Reference.** Now that you have defined the validation and posting algorithms, return to your column reference for account relationship type and define the algorithms there.

Design a Campaign to Include Linking an Additional Person

Any campaign related to a specific account may include the questions and miscellaneous fields defined here to create/link an additional person to the account.

Simply create entries in the questions and miscellaneous fields collection to prompt the user to ask for the required information. A question should exist for each column reference created above. The following table illustrates a possible setup.

Seq	Description	Prompt on Order	Column Reference	Dependency
10	Account relationship type for additional person.	If you would like to link another person to this account, please enter an account relationship type.	ACCT-REL-TYPE	Must have account
20	Person ID for additional person	If the additional person already exists in the database, enter the person id.	PERSON-ID	Must have account
30	Person name for additional person	If you would like to create a new person, please enter the person name.	PERSON-NAME	Must have account
40	ID type for additional person	Please enter the ID type of the new person.	PER-ID-TYPE	Must have account

Copyright © 2026, Oracle and/or its affiliates.

50	ID Number for additional person	Please enter the ID number of the additional person.	PER-ID-NUM	Must have account

Service Credits Earned When Starting Service

For some memberships, you may want to add service credits when starting the program. For example, the customer gets three free pay-per-view movies when signing on for cable service. Or a customer receives 500 frequent flier miles for signing up for a combination of gas and electric service.

There are various ways that you can accomplish this. You should work with your implementers to consider the various options to determine the method that best suits your business practice.

- You could use an SA activation algorithm to create a membership and an event for the initial points. Use this method if the points are related to a single type of service agreement and the points are earned automatically when starting service (i.e., without any human decision to be made).
- You could use a membership creation or membership activation algorithm to generate a service credit event automatically. This method assumes that the decision to create the membership has been made and that the free initial points are always earned for this type of membership (i.e., regardless of the type of service created).
- You could use questions and miscellaneous fields for a campaign/package to determine a customer's eligibility for participation in the membership and for the initial free points. A column reference posting algorithm could create the membership and/or the service credit event for the free points based on the answers to the questions.
- Perhaps the initial free points are only earned after the first bill is generated. Use a bill completion algorithm to generate the initial points.

You may think of other plug-in spots that could be used to generate free initial points based on your business needs.

Service Credits Earned Through Billing

For some memberships, you may accumulate points as a result of billed amounts for other services. For example, perhaps your customers earn one frequent flier mile for every $10 spent on the combined electricity, gas and water bills each month.

To accomplish this, you must design an algorithm to be executed at billing time. There are various plug-in spots executed at billing time that you may use, but the recommended plug-in is a post completion algorithm on the customer class. This plug-in is executed after all bill segments are frozen and most of the completion logic has occurred.

Your algorithm should determine the service agreements that contribute to the membership and calculate the service credit amount for those service agreements bill segments.

This algorithm should also consider what to do when bill segments that contributed to the event are canceled. The algorithm provide with the base product CBCM-SC checks to see if any canceled bill segments are referenced on any previous events. If so, it includes the canceled amount on the new event. This may cause the new event to be a negative amount. The assumption is that over time, earned credits will compensate for the negative event amount.

Copyright © 2026, Oracle and/or its affiliates.

For a membership that interfaces information to a third party, it is assumed that the negative event amount is also interfaced.

Service Credits Redeemed Through Billing

For some memberships, your customer may redeem their earned points by receiving a discount on their bill. For example, if your cable customer has earned one free pay-per-view movie, you can give them a credit the next time they get billed for a pay-per-view movie.

The base product has not provided any algorithms to credit a bill based on earned service credits. This section will identify considerations your implementers should follow when designing algorithms to redeem service credits through billing.

The recommendation is to credit the customer's bill by generating an adjustment using a pre-bill completion algorithm. This algorithm's responsibilities are as follows:

- Determine if the appropriate bill segment(s) contain the charges that need to be credited. For example, if the service credit is for a free pay-per-view movie, determine if the customer has been billed for a pay-per-view movie.
- Determine if the customer's current service credit balance for this program. For example, how many free pay-per-view movie credits are left?
- Create an adjustment to credit the appropriate service agreement by the eligible credit amount.
- Update the service credit membership balance by creating a new SC event with a credit for the number of points redeemed. Link the adjustment to the event as a contributed to FT.
- Consider cancel/rebill situations. If a cancel/rebill has occurred, determine if there is a change to the redeemed credits. For example, if the original bill had one pay-per-view movie that was credited and the new bill also has one pay-per-view movie, no change is needed to the service credit balance. If the original bill had more pay-per-view movies than the new bill, perhaps one or more redeemed service credit points should be reinstated. If the original bill had fewer pay-per-view movies than the new bill, perhaps more points should be redeemed.

You may wonder why we don't recommend crediting the customer's bill while generating the bill segment. For example, use pre-processing calculation rule to determine if any points should be redeemed and use a calculation rule to generate a bill calculation line with the credit amount.

The reason for this is that cancel/rebill logic is not straightforward. Algorithms executed during rate application should NOT perform any updates, such as updating the service credit membership balance. The balance should be updated using a bill segment freeze algorithm or a bill completion algorithm.

When a cancellation occurs, the service credit balance should be updated to reinstate the redeemed points. Again, this should occur when the cancellation is frozen or at bill completion time. If you perform a cancel/rebill, the calculation of the rebill segment does not have the up-to-date information about the service credit balance because the reinstatement of the points by the canceled segment has not occurred yet.

Designing Your Rate Options for Capital Credits

The capital credit allocation background process relies on certain data configuration in order to function correctly. This section identifies the required data setup.

Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> Refer to Allocating Capital Credits for more information.

- Identifying SQ and Sales Information for Historical Bill Segments
- Designing Bill Factors for Credit Allocation

Identifying SQ and Sales Information for Historical Bill Segments

To allocate capital credits, the background process retrieves billing history for each SA that contributes to the membership for the given fiscal year. The process needs to calculate the service quantity (SQ) amount billed and the sales amount billed for the SA in that year.

> **Note**
>
> **Sales Amount**. The sales amount refers to the monetary amount billed. For a capital credit allocation, this amount would generally exclude taxes and may exclude other line item amounts from the bill.

In order to calculate the amounts correctly, the background process must determine which bill calculation lines for each bill segment contain the SQ and/or sales information. Characteristics on the bill calculation lines identify which bill lines should be used.

The rest of this section uses examples to illustrate how you may configure your rate options to support this.

- Define Characteristics for SQ and Sales
- Define Calculation Rules
- Define Batch Control Parameters

Define Characteristics for SQ and Sales

Char Type for identifying bill lines that hold **SQ information**

- Char Type: CCA-SQ
- Description: Capital credit allocation usage
- Type of Char: Pre-defined
- Values: Y
- Char Entities: Calculation Rule, Bill Calculation Line

Char Type for identifying bill lines that hold **sales information**

- Char Type: CCA-SALES
- Description: Capital credit allocation sales
- Type of Char: Pre-defined

Copyright © 2026, Oracle and/or its affiliates.

- Values: Y
- Char Entities: Calculation Rule, Bill Calculation Line

Define Calculation Rules

Identify each rate schedule that may be linked to a service agreement that contributes to a capital credit membership.

For each of these rate schedules, identify the calculation rules whose resulting bill calculation line will contain a billable service quantity that should be included in the SQ calculation for allocating capital credits. For each calculation rule that qualifies, define the CCA-SQ char type (defined above) and a char value of Y.

For each rate schedule, identify the calculation rules whose resulting bill calculation line amount should be included in the sales calculation for allocating capital credits. For each calculation rule that qualifies, define the CCA-SALES char type (defined above) and a char value of Y.

> **Note**
>
> **Characteristic Information**. The system automatically copies characteristic info from a calculation rule to its resulting bill calculation line if the char type entities include both calculation rule and bill calc line.

Define Batch Control Parameters

The background process to allocate capital credits CPCRALOC receives the char type and char value to identify the bill calculation lines that contain the SQ and sales amounts. Once you have your characteristics defined, update your batch control to include these values as default parameter values.

Designing Bill Factors for Credit Allocation

The capital credit allocation process uses an allocation factor in its calculation. A typical capital credit membership may define multiple subcategories, meaning that allocation amounts are calculated each year for the multiple subcategories. The calculation is the same for each subcategory, but the allocation factor differs.

The process has been designed to calculate the allocation for a single subcategory. If your organization requires allocations calculated for multiple subcategories, the process must be run for each subcategory. The allocation process receives a bill factor as an input parameter. As a result, a different bill factor should be set up to define the allocation factor for each subcategory.

For each subcategory, the allocation factor may differ further for the type of customer. For example, the allocation for a commercial customer may differ from the allocation factor for a residential customer. The allocation background process expects the bill factor for each subcategory to define a characteristic type of revenue class. The process determines each service agreement's revenue class by looking at the value defined on its SA type.

Copyright © 2026, Oracle and/or its affiliates.

Following is an example of bill factors set up for a capital credit membership with two subcategories: transportation and generation.

- Characteristic Type for Allocation Bill Factor
- Bill Factor for Transportation
- Bill Factor for Generation

Characteristic Type for Allocation Bill Factor

Char Type: REV-CLASS

Description: Revenue Class

Type of Char: Pre-defined

Values: (define all the valid revenue class values)

Char Entity: Bill Factor

Bill Factor for Transportation

Bill Factor Id: CCAF-TRANS

Description: Transportation Allocation Factor

Char Type: REV-CLASS

Char Source: Characteristic Collection

Char Values: (for each year, the new transportation allocation factor for each revenue class must be defined)

Bill Factor for Generation

Bill Factor Id: CCAF-GEN

Description: Generation Allocation Factor

Char Type: REV-CLASS

Char Source: Characteristic Collection

Char Values: (for each year, the new generation allocation factor for each revenue class must be defined)

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The characteristic source is characteristic collection.** It is the responsibility of the background process to determine the SA's revenue class and to pass this value into the bill factor routines to retrieve the correct bill factor value.
>
> **Estimating Allocation Factors**. Often the company needs to estimate the allocation factors for the new fiscal year and may adjust the values several times until the calculated allocation amounts are satisfactory. Refer to Allocating Capital Credit for more information.

Partial Retirement

In the cooperative business, it is common to never retire certain capital credit allocation amounts. The amounts that do not retire should be assigned their own subcategory.

When executing the retirement background process, the subcategory to retire may be input to the process. If you have certain subcategories that you do not retire, you would simply run the background process for the subcategories that do retire.

Cooperatives typically retire amounts and transfer the amounts to a beneficiary when a member dies. This is known as "estate retirement". Refer to Service Credits for Capital Credit Memberships for more information. If your business practice designates that certain subcategories of allocated amounts do not get retired, this probably holds true for estate retirement as well. If that is the case, your membership inactivation algorithm should be designed to only retire the appropriate amounts by subcategory.

Interface Membership Information to a Third Party

For some memberships, you may accumulate points for a third party, for example accumulating frequent flier miles for an airline. For these types of memberships, you must interface the event information to the third party.

To interface information to a third party, you may choose one of the following options:

- Design an extract program to interface the information
- Use workflow and notification to interface the information via the XAI tool
- Interface Via an Extract Program
- Other Considerations For Interfacing Info to a Third Party

Interface Via an Extract Program

The service credit event may indicate a batch code and batch run number. Design a program to extract event information to a third party. This extract program would select service credit events marked with its batch code and the current run number.

Copyright © 2026, Oracle and/or its affiliates.

Your service credit event must define a completion algorithm that stamps the appropriate batch code and run number. The base product provides an algorithm type to perform this logic. Refer to SCEC-BT for more information.

Other Considerations For Interfacing Info to a Third Party

Because event information is extracted to a third party, you must consider how to handle adjustments to the event amount. For example, if your event is generated based on the creation of a bill segment, what should happen if that bill segment is cancelled?

You may want to prevent these types of events from getting canceled. Validation like this may be added via a user exit or using an event cancellation algorithm.

> **Note**
>
> **Sample Algorithms**. The product does not provide any base algorithms to prevent an event from being canceled.

You should allow negative event amounts to be created so that this information may also be sent to the third party's system.

Defining Loan Options

The topics in this section describe how to set up the system to enable loan functionality.

> **Note**
>
> **Loans are optional.** The system configuration requirements described in this section are only relevant if your organization loans money to customers.

- The Terms Of A Loan Are Stored On A Service Agreement
- Payoff Balance and Current Balance for Loans
- Booking The Principal Amount Using An Adjustment
- Loan Amortization Schedule Calculation
- Billing For Loans And Interest Calculation
- Paying What Is Owed
- Overpayments On Loans
- Adjusting Loan Amounts
- Writing Off Loans
- Distribution Codes for Loans
- Setting Up The System To Enable Loans

Copyright © 2026, Oracle and/or its affiliates.

The Terms Of A Loan Are Stored On A Service Agreement

Loans are initiated by creating a loan service agreement for a customer. The loan service agreement (and its SA Type) contains the loan's terms:

- The **loan amount** is held in the service agreement's Total Amount to Bill.
- The **customer's periodic payment amount** is held in the service agreement's Recurring Charge Amount.
- The **number of amortization periods** (e.g., 36 months, 240 months, etc.) is held in the service agreement's Number of Payment Periods.
- If the **interest rate** is the same for all customers with this type of loan service agreement, the interest rate is defined on the service agreement's SA type (using a bill factor). A specific interest rate can be defaulted from a start option contract value. If a specific interest rate applies to the customer, the SA type's interest rate can be overridden by specifying a bill factor value on the customer's service agreement (where the bill factor value contains the specific interest rate for the customer).
- The SA type controls the **periodicity of the bills** (e.g., monthly or bi-weekly).

Because a loan is defined using a service agreement, the typical functionality that is controlled by the service agreement's SA type is supported, including:

- How and when it is billed.
- How payments are booked in the GL (and the payment priority relative to other service agreements).
- How its debt is monitored by credit and collections.
- How late payment charges are calculated.

Loan service agreements are created using Start/Stop just like all other service agreements. The start/stop transaction has special loan functionality that allows an operator to specify the service agreement-specific loan terms described above. A start option can be specified to override the SA type's interest bill factor.

> **Note**
>
> **Automatic calculation of periodic payment amount / number of periods.** The system calculates a loan's periodic payment amount or number of payments (whichever is left blank). You can have the system do this on Start/Stop Maintenance (using the **Calculate** button that appears on the start service confirmation window), and on Loan - Main (by clicking the **Calculate** button). Regardless of where you do this, the calculation is performed by an algorithm on the loan's SA type. Refer to the LPDA-SI algorithm type for more information about the base package algorithm.

Payoff Balance and Current Balance for Loans

As described under Current Amount versus Payoff Amount, a loan service agreement's current balance and payoff balance differ during the lifetime of the loan. Current balance contains how

Copyright © 2026, Oracle and/or its affiliates.

much the customer owes to remain current (typically their periodic payment amount), and payoff balance contains the amount the customer would have to pay to payoff the loan (typically the principal balance plus any accrued interest charges).

Unlike other SAs, loans have two accounts receivable distribution codes: long term and short term. These two codes allow the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). The current balance for a loan is always the amount of the short-term receivables. The payoff balance for a loan is always the net of the short-term receivables and the long-term receivables.

If the SA has a special role of Loan, the financial transaction algorithms supplied with the base package transfer the current amount between the long-term receivables and the short-term receivables in the GL. For example, when a bill segment is generated for the loan SA, the amount of the periodic payment is transferred from the long-term receivables to the short-term receivables. Don't worry, the examples in the following sections show exactly what these transactions look like.

An operator can see the how the bills, payments and adjustments have affected the GL, current balance and payoff balance using SA Financial History.

The following sections provide examples of how adjustments, bills and payments are recorded in the GL and the subsequent effect on the current and payoff balances. When reading the examples, remember that the payoff balance is always the net of the short-term receivable and the long-term receivable balances.

Booking The Principal Amount Using An Adjustment

When a loan service agreement is activated (i.e., when its status changes from pending start to active), an adjustment is created to book the principal amount. If the customer takes out a loan of 10,000, the adjustment's financial transaction looks as follows:

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated (and an adjustment is created to book the principal)	Long Term Loan Receivable 10,000 Cash <10,000>	0	10,000	0	10,000

This adjustment is issued if:

- The service agreement's SA type indicates a special role of Loan.
- The loan service agreement's Total Amount to Bill contains an amount (i.e., the loan amount).
- The loan service agreement was created using a start option on Start/Stop Maintenance AND the start option references an adjustment type and this adjustment type has been set up as follows:
  - The adjustment type's distribution code should reference the GL account to credit (e.g., Cash).

Copyright © 2026, Oracle and/or its affiliates.

– The adjustment type's FT algorithm reference Payoff Amt = Adj Amt / Current Amt = 0 (booking principal only impacts a customer's payoff balance).

Note that because this financial transaction doesn't have a current amount (the customer doesn't actually owe a current amount yet), there is no need to book anything to the short-term receivables distribution code.

Loan Amortization Schedule Calculation

The amortization schedule is a projection of the amount of principal and interest in each payment over the life of the loan. The amortization schedule may change, for example if the interest rate changes or the customer makes an overpayment (reducing the principal balance).

A loan's amortization schedule is calculated when an operator clicks the **Calculate** button on Loan - Main . Please be aware that when this button is clicked, an algorithm plugged in on the loan SA type actually calculates the amortization schedule (refer to the algorithm type LSCH-SI for more information).

Billing For Loans And Interest Calculation

A bill segment is produced for a loan when its service agreement's account is next billed.

Factors on a loan's SA type controls when a bill segment is produced for a loan. Typically SA types for loan service agreements are set up to use anniversary calendar billing. For this configuration:

- You must indicate the type of anniversary billing in the calendar billing option. Currently, we only support the Anniversary Future Billing (meaning that loans are billed in advance just like a classic home loan is). Refer to Using The Anniversary Method for more information about how this billing method controls the end date of the loan bill segment.
- You must reference an anniversary billing frequency consistent with the recurring charge frequency (e.g., monthly, quarterly, etc.).

To set up for a loan that is billed on a monthly basis, you would define the following fields in SA Type - Billing:

- **Use Calendar Billing**: Anniversary Future Billing
- **Anniversary Bill Frequency**: Monthly
- **Total Bill Amount**: Required
- **Recurring Charge**: Required
- **Recurring Charge Frequency**: Monthly
- **Total Amount To Bill Label**: Loan Amount
- **Recurring Chg Amt Label**: Payment Amount

If your type of loan must be billed on an exact date (for example, always on the 15th of the month) or with an exact number of days between each bill (for example, every 14 days), then your loan should be set up to use the calendar billing option of Use Bill Period. In order for your loan bills to be created on specific dates, the system makes the following assumptions:

- Your bill cycle schedule for the loan's account is defined with the dates that you want the loan to bill and is defined with the window start date equal to the window end date.

Copyright © 2026, Oracle and/or its affiliates.

- You define a bill period schedule corresponding to your bill cycle schedule. Each bill period schedule's bill date should match your bill cycle window start date and each bill period schedule's bill segment end date should be set to the loan period end date.
- Considerations for the first bill. If the loan SA Type's Initial Start Date Option indicates that the first day of the service agreement should be billed, then the loan SA's start date should match the window start date of the next bill cycle schedule for the account. If the loan SA Type's Initial Start Date Option indicates that first day of the service agreement should not be included, then the loan SA's start date should be one day prior to the window start date of the next bill cycle schedule for the account.
- Define your recurring charge frequency to match the frequency of your bill periods.

How the bill segment affects the customer's balance, and how it affects the general ledger are controlled by several algorithms defined on the loan service agreement's SA type and bill segment type:

- The SA type's loan schedule algorithm controls how the loan amortization schedule is calculated.
- The SA type's loan interest charge algorithm controls how interest is calculated.
- The bill segment type's create algorithm controls how the bill lines are constructed.
- The bill segment type's financial algorithm controls how the general ledger is affected by the bill.

The second entry in the following table contains an example of the financial transaction produced by the base package algorithms (note, the first financial transaction in the table was described under Booking The Principal Amount Using An Adjustment).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95

Several important points are illustrated above:

Copyright © 2026, Oracle and/or its affiliates.

- When a bill segment is produced for a loan service agreement, the following takes place (if you use a bill segment creation algorithm of Create Bill Segment for Loans):
  - The loan SA type's bill segment creation algorithm calls the SA type's loan interest charge algorithm.
  - The base package loan interest charge algorithm calculates simple interest based on: 1) unbilled principal (i.e., the service agreement's payoff balance minus the current balance), 2) the number of billing periods covered by the bill, and 3) the interest rate. Refer to the algorithm type LINT-SI for more information about the base package interest calculation algorithm.

> **Note**
>
> **No interest on interest and no interest on past due amounts.** Just like a classic home loan, the base package algorithms do not calculate interest on interest, nor do they calculate interest on past due amounts. If you want to levy a late payment charge, use the SA type's late payment processing. If your organization calculates interest differently, you must develop your own algorithm(s).

- The SA type's bill segment creation algorithm uses the calculated interest to format the bill segment's bill lines. It creates one bill line to show the amount of interest in the payment, and another bill line to show the amount of principal. The principal amount is equal to the service agreement's periodic payment amount minus the amount of calculated interest.
- The financial transaction illustrated above is created if you use a bill segment financial algorithm of Payoff Amt = Interest / Current Amt = Bill Amount on the loan's bill segment type. The following explains how this algorithm works:
  - The SA's current balance increases by the amount of the loan's periodic payment amount (i.e., its recurring charge amount). In other words, the amount the customer thinks they owe increases by 438.71.
  - The SA's payoff balance increases by the amount of interest. In other words, if the customer wanted to payoff the loan, they'd owe 10,041.66.
  - The *Interest* portion of the GL accounting is straightforward (if you're an accountant). It simply takes the amount of interest and debits it to the SA type's receivable distribution code (long-term receivables) and credits it to the distribution code defined on the interest rate's bill factor (typically interest revenue).
  - The *Transfer* portion of the GL accounting transfers moneys from long-term receivables (i.e., the unbilled principal) to short term receivables (the billed portion of the debt). The amount transferred is equal to the FT's effect on the service agreement's current balance, allowing the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Remember that the payoff balance is the net of the short-term and long-term receivables.

Paying What Is Owed

When a payment is made for a loan:

- The service agreement's payoff amount is reduced by the payment amount.
- The service agreement's current amount is reduced by the payment amount.

Copyright © 2026, Oracle and/or its affiliates.

The events that happen when a customer makes a payment against a loan is controlled by the FT algorithm plugged in on the loan service agreement's payment segment type. We'll use an example to help explain how this algorithm works. The 3 rd entry in the following table illustrates the financial transaction produced when a payment is made (note, the first two financial transactions were described above).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000
First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95
Payment is made	Affect Cash Cash 438.71 Long Term Loan Receivable <438.71> Transfer Long Term To Short Term: Long Term Loan Receivable 438.71 Short Term Loan Receivable <438.71>	-438.71	-438.71	0	9,602.95 short-term: 0 long-term: 9,602.95

The financial transaction illustrated above is created if you use a payment segment FT creation algorithm of Payoff Amt = Current Amt = Pay Amt on the loan's payment segment type. The following explains how this algorithm works:

Copyright © 2026, Oracle and/or its affiliates.

- The SA's current balance decreases by the amount of the payment amount. In other words, the customer thinks they owe 0 after the payment. Note refer to Overpayments for information about how an overpayment affects the SA's balances and the general ledger.
- The SA's payoff balance decreases by the amount of the payment. In other words, if the customer wanted to payoff their loan, they'd owe 9,602.95.
- The *Cash* portion of the GL accounting is straightforward (if you're an accountant). It simply takes the amount of the payment and debits it to the payment segment type's distribution code (typically cash) and credits it to the SA type's receivable distribution code (long-term receivable).
- The *Transfer* portion of the GL accounting effectively reduces short-term receivables by the FT's effect on the customer's current balance. This reduction is handled by an offsetting increase to long-term receivables (to make up for reduction made when the cash was applied). Again, this differentiation between short-term and long-term receivables allows the general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term).

Overpayments On Loans

You can determine whether you want to accept loan overpayments. Overpayments reduce the principal amount (the amount owed on the loan), which follows the philosophy adopted by a typical home loan.

When the payment is made, any overpayments are distributed according to the overpayment distribution algorithm defined for the customer class. Any customer classes for which you want to allow loan overpayments should use an overpayment distribution algorithm that keeps the overpayment on the loan SA.

When the payment transaction is frozen, the system checks to see if there is a credit amount on the loan SA's current balance. If a credit exists, the customer has made an overpayment and an adjustment is created to zero out the current balance and transfer the amount of the credit from the SA's short-term receivable to long-term receivable. The adjustment may appear on the customer's next bill to show the additional amount paid against the principal.

The algorithm that controls this adjustment to remove the credit on current balance is plugged in on the loan service agreement's SA type and is applied on the SA Type - Payment Freeze system event.

The third and fourth entries in the following table illustrate an overpayment (note, the first two financial transactions were described above).

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 10000 Cash <10000>	0	10,000	0	10,000 long-term: 10,000

Copyright © 2026, Oracle and/or its affiliates.

First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95
Payment is made (with an overpayment of 200.00)	Affect Cash Cash 638.71 Long Term Loan Receivable <638.71> Transfer Long Term To Short Term: Long Term Loan Receivable 638.71 Short Term Loan Receivable <638.71>	-638.71	-638.71	-200.00	9,402.95 short-term: <200.00> long-term: 9,602.95
Create adjustment to remove SA's credit.	Transfer Short Term Credit to Long Term: Short Term Loan Receivable 200.00 Long Term Loan Receivable <200.00>	200.00	0	0	9,402.95 short-term: 0 long-term: 9,402.95

In the third financial transaction illustrated above, the billed amount of the payment works essentially the same as that illustrated under Paying What Is Owed. If the Keep Overpayment on Loan SA algorithm is plugged in on the overpayment distribution event on the customer class, the overpayment amount is applied to the loan SA, creating a credit on the SA's current balance. The following explains how this algorithm works:

- If the account has an loan SA and there is an excess credit, the credit is applied to the loan SA (as long as this does not cause the loan SA to have a credit payoff balance).

Copyright © 2026, Oracle and/or its affiliates.

- If there is not a loan SA to which the credit can be applied, the algorithm checks to see if there is an open excess credit SA for the account.
  - If so, the excess credit amount is applied to the excess credit SA.
  - If not, the algorithm creates an excess credit SA and applies the amount to this SA.

The fourth financial transaction illustrated above is created if the Create Adjustment to Remove SA's Credit algorithm is plugged in on the loan's SA type's payment freeze event. The following explains how this algorithm works:

- If the SA's current balance is less than zero, the algorithm creates a frozen adjustment that removes the credit by transferring the credit from the short-term receivable to the long-term receivable. This adjustment ID is captured on the pay segment so the adjustment can be canceled if the payment is later canceled. Note that the adjustment cancel reason used by the system in this case is specified on the Financial Transaction Options Feature Configuration using the Adjustment Cancel Reason For Payment Linked To Adjustment option type.
- The SA's current balance increases by the amount of the credit transfer. In other words, the customer thinks they owe 0 after the transfer.
- The SA's payoff balance doesn't change because the payoff balance is always the net of the short-term and long-term receivables. In other words, if the customer wanted to payoff their loan, they'd still owe 9,402.95.

> **Note**
>
> **Overpayments and interest.** The base package interest calculation algorithm (plugged in on the loan's SA type) does not take into consideration the exact date that the overpayment is made when calculating the interest for the period. It only takes into consideration the outstanding principal amount (payoff balance - current balance) at the time of the interest calculation.

Adjusting Loan Amounts

You would issue ad hoc adjustments if you need to change a loan's payoff and/or current balance outside of the normal billing / payment functions.

An adjustment can:

- Reduce a loan's payoff balance.
- Reduce a loan's current balance (i.e., how much the customer thinks they currently owe).
- Reduce both the loan's payoff and current balance.

For adjustments that affect payoff amount only:

- These adjustments are used to change the principal owed, e.g., if an additional amount is loaned.
- The adjustment's adjustment type should reference the Payoff Amt = Adj / Current Amt = 0 FT algorithm.
- GL lines will be generated to reflect the change to principal.

For adjustments that reduce current amount only:

Copyright © 2026, Oracle and/or its affiliates.

- These adjustments can be used to change the amount that the customer must pay as part of the next bill.
- The adjustment's adjustment type should reference the Payoff Amt = 0 / Current Amt = Adj Amount (no GL) FT algorithm.
- Typically, GL lines are not generated for FTs that only affect the customer's current balance. However, moneys must be transferred from long to short in the amount of the adjustment (as described above under Payoff Balance and Current Balance for Loans).

For adjustments that reduce both payoff and current amount:

- These adjustment types can be used to levy additional charges, such as late fees, or to correct interest calculations.
- The adjustment's adjustment type should reference the Payoff Amt = Adj / Current Amt = Adj FT algorithm.
- GL lines are generated to reflect the change to principal. In addition, GL lines must be generated to transfer money from long to short in the amount of the adjustment (as described above under Payoff Balance and Current Balance for Loans).

> **Note**
>
> Adjustments that affect the principal balance (payoff balance - current balance) affect the term of the loan because interest is based on the principal balance.
>
> **Adjustments can cause credit balances to exist.** If you credit the loan SA, it is possible for the current balance to become negative. You may need to create additional adjustments that affect the current amount, depending on whether the customer needs to pay this amount as part of the next payment.

Writing Off Loans

Loans are written off using the standard write-off processing.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Write Off Processing for background information.

We illustrate the classic financial transactions that transpire to financially write-off a loan to help illustrate important points (these are the fourth and fifth entries in the following table):

Event	GL Accounting	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
Loan service agreement is activated	Long Term Loan Receivable 1000 Cash <1000>	0	10,000	0	10,000 long-term: 10,000

Copyright © 2026, Oracle and/or its affiliates.

First bill segment is produced	Interest: Long Term Loan Receivable 41.66 Interest Revenue <41.66> Transfer Long Term To Short Term: Short Term Loan Receivable 438.71 Long Term Loan Receivable <438.71>	438.71	41.66	438.71	10,041.66 short-term: 438.71 long-term: 9,602.95
Payment is made (with an overpayment of 200.00)	Affect Cash Cash 638.71 Long Term Loan Receivable <638.71> Transfer Long Term To Short Term: Long Term Loan Receivable 638.71 Short Term Loan Receivable <638.71>	-638.71	-638.71	-200.00	9,402.95 short-term: <200.00> long-term: 9,602.95
Create adjustment to remove SA's credit.	Transfer Short Term Credit to Long Term: Short Term Loan Receivable 200.00 Long Term Loan Receivable <200.00>	200.00	0	0	9,402.95 short-term: 0 long-term: 9,402.95

Copyright © 2026, Oracle and/or its affiliates.

Sync up current with payoff balance	Transfer Long Term To Short Term: Short Term Loan Receivable 9,402.95 Long Term Loan Receivable <9,402.95>	9,402.95	0	9,402.95	9,402.95 short-term: 9,402.95 long-term: 0
Transfer balance to a write-off SA	Transfer From Loan SA: Write Off Xfer Clearing 9,402.95 Long Term Loan Receivable <9,402.95> Long Term Loan Receivable 9,402.95 Short Term Loan Receivable <9,402.95> Transfer To Write Off SA: Bad Loans Expense 9,402.95 Write Off Xfer Clearing <9,402.95> Note, these will cause a balance of 9,402.55 to exist on the write-off service agreement.	-9,402.95	-9,402.95	0	0

The only unusual portion of the last two financial transactions is the impact on short and long term receivables. Please see the examples above under Billing For Loans And Interest Calculation and Paying What Is Owed to understand how any impact to a loan's current balance causes this type of financial transfer to occur.

Distribution Codes for Loans

As explained above under Payoff Balance and Current Balance for Loans, loans have two accounts receivable distribution codes: long term and short term. These two codes allow the

Copyright © 2026, Oracle and/or its affiliates.

general ledger to differentiate between unbilled loan receivables (long term) and billed loan receivables (short term). Both receivables distribution codes are defined on the loan SA type.

In addition, loans have a distribution code used to book interest revenue. The interest revenue distribution code is defined on the loan's bill factor value for a revenue class (defined on the loan's SA type). For example, on the bill factor you can use one distribution code to book interest revenue from the residential revenue class and another distribution code to book interest revenue from the commercial revenue class. In this example, you create two loan SA types, one for residential revenue and the other for commercial revenue.

Loans may also have a bad loan debt (expense) distribution code that is used when writing off a loan. The bad loan debt distribution code is defined on the loan's write-off service agreement type. Refer to Defining Credit & Collections Options for more information.

Setting Up The System To Enable Loans

The above topics provided background information about how loans are supported in the system. The following discussion summarizes the various setup tasks alluded to above.

- Distribution Code
- Adjustment Types
- Adjustment Type Profile
- Algorithms
- Bill Factor
- Customer Class
- Bill Segment Type
- Frequency
- Bill Period
- Bill Cycle Schedule
- Collection, Severance and Write Off Processes
- SA Type
- Start Options

Distribution Code

You must set up the following distribution codes:

- Long term receivables
- Short term receivables
- Interest revenue
- Bad loan debt

Copyright © 2026, Oracle and/or its affiliates.

Adjustment Types

The following adjustment types are needed:

- Activate a loan. This should reference a distribution code associated with cash or the cash equivalent and an FT algorithm of Payoff Amt = Adj / Current Amt = 0.

> **Note**
>
> **Creating Checks for Loan Amounts.** If you want the system to initiate a check to the customer for the loan amount, the loan activation adjustment type should indicate an **A/P Request Type Code**. Refer to Controls The Interface To A/P & 1099 Reporting for more information.

- Remove Credit (Overpayment). This adjustment should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Even though this algorithm indicates that there is no effect on the GL, there is a special exception built in for loan SAs. The special exception creates GL details to transfer the current balance (short-term receivable) to the long-term receivable. Refer to Overpayments for more information.

> **Note**
>
> **Printing Overpayments on Bills**. If you want the overpayment adjustment to appear on customers' bills, turn on **Print by Default** and enter a **Description on Bill** (e.g. "Additional Principal"). For more information, refer to Controls Information Printed On The Bill.

- Adjustment types to perform any of the adjustments described under Adjusting Loan Amounts.

Adjustment Type Profile

Create an adjustment type profile that references the adjustment types used on a loan. Besides the above adjustment types, it should also reference adjustment types to levy late payment charges (if applicable), levy non-sufficient funds charges (if applicable), refund overpayments (if applicable), sync current balance with payoff balance at write-off time, transfer balances to a write-off service agreement, and write-down small balances.

Algorithms

Add the following algorithms:

- Overpayment Distribution. This algorithm is later plugged in on the customer class for the Overpayment Distribution system event. Refer to the algorithm type OVPY-LO-CSA for more information about the base package algorithm.

Copyright © 2026, Oracle and/or its affiliates.

- Bill Segment Creation for Loans. This algorithm is later plugged in on the loan's bill segment type. Refer to the algorithm type BSBS-LO for more information about the base package algorithm.
- Bill Segment Financial Transaction Creation for Loans. This algorithm is later plugged in on the loan's bill segment type. Refer to the algorithm type BSBF-LO for more information about the base package algorithm.
- Amortization Schedule. This algorithm is later plugged in on the SA type for the Loan Schedule system event. Refer to the algorithm type LSCH-SI for more information about the base package algorithm.
- Interest Calculation. This algorithm is later plugged in on the SA type for the Loan Interest Charge system event. Refer to the LINT-SI algorithm type for more information about the base package algorithm.
- Payment Periods/Payment Amount Calculation. This algorithm is later plugged in on the SA type for the Loan Periods and Amount system event. Refer to the algorithm type LPDA-SI for more information about the base package algorithm.
- Loan SA Payment Freeze. This algorithm has a parameter that must reference the Remove Credit adjustment type defined above. This algorithm is later plugged in on the SA type for the Payment Freeze system event. Refer to the algorithm type STPZ-RMVCR for more information about the base package algorithm.

Bill Factor

You must set up a bill factor that defines the interest rate. If you have different interest rates for different types of loans, you can create a separate bill factor for each or you can use start options to override the interest rate.

On the bill factor's bill factor value, make sure to reference the GL distribution code used to book interest revenue for the revenue class specified on the loan's SA type.

Customer Class

Any customer class on which you want to allow overpayments for a loan must use the overpayment distribution algorithm defined above. The overpayment distribution algorithm keeps the overpayment on the loan SA rather than transferring it to an excess credit SA, allowing a subsequent adjustment to apply the overpayment to the principal balance.

Bill Segment Type

Create a bill segment type that references the bill segment creation algorithm defined above and the FT creation algorithm defined above.

Frequency

Copyright © 2026, Oracle and/or its affiliates.

Create frequency codes to correspond to the frequency of your loans. When setting up your SA types, you must indicate a recurring charge frequency and, if you use the Anniversary Billing Option, you must indicate an anniversary frequency.

Bill Period

If you use the Use Bill Period option, set up a bill period with an appropriate schedule of dates for billing your loan.

Bill Cycle Schedule

If you use the Use Bill Period option, the bill cycle schedule for these types of loans should be defined with an appropriate schedule of dates for billing your loan.

Collection, Severance and Write Off Processes

You should set up the appropriate credit and collections information. Refer to Defining Credit & Collections Options for more information.

SA Type

You must set up a SA type for your loan service agreements (you may need multiple SA types if you have different business rules for different types of loans). The following points describe the minimal requirements for a loan SA type.

- SA Type - Main
- SA Type - Detail
- SA Type - Billing
- SA Type - Rate
- SA Type - SP Type
- SA Type - Adjustment Profile
- SA Type - Credit and Collections
- SA Type - Algorithms

SA Type - Main

Define the following options:

- **Distribution Code** should be a long-term receivable code.
- **Service Type** should references something like "Miscellaneous Service".
- Specify the **Revenue Class** that, together with the interest bill factor, determines the distribution code used to book the loan interest revenue.

Copyright © 2026, Oracle and/or its affiliates.

- **Start Option** should be turned on.
- The Payment Segment Type should reference the Normal Payment. The base package payment segment financial algorithm (Payoff Amt = Current Amt = Pay Amt) used for Normal Payment pay segment types creates the additional GL details to transfer the credit from long-term receivables to short-term receivables if the SA's special role is Loan.
- Turn on **Late Payment Charge** if applicable.
- Define an appropriate **Adj**ustment **Type (Synch Current)** that will cause current balance to be synchronized with payoff balance (if the loan is written off).

SA Type - Detail

Define the following options:

- **Special Role** is Loan.
- Specify the **Interest Bill Factor** set up above. You can use start options to override.
- Use **Override Interest** to indicate whether the interest rate defined on the interest bill factor may be overridden at the SA level. If you select Allowed, the interest rate may be overridden by a contract value on a start option or the SA.
- Use the short-term receivable account defined above as the **Loan A/R Distribution Code**. If you do not want the system to differentiate between short-term receivables and long-term receivables, make the loan A/R distribution code the same as the distribution code (above).

SA Type - Billing

> **NOT_SUPPORTED**
>
> For an overview of some of these options, be sure to refer to Billing For Loans And Interest Calculation for more information.

Define the following options:

- The Bill Segment Type should reference the value created above.
- **Characteristic Premise Req**uired should not be checked. (A loan SA is not a premise-based service. SA types that have this box checked are filtered out of the SA type search when the start method is Start a SA, so users will never be able to start a loan SA that requires a characteristic premise.)
- **Use Calendar Billing** must equal Anniversary Future Billing or Use Bill Period.
- **Bill Period** is required for the Use Bill Period option.
- **Anniversary Billing Frequency** is required for the Anniversary Future Billing option and must equal the periodicity of the bills (monthly, weekly, etc.).
- **Total Bill Amount** is required (it holds the principal).
- **Total Amount To Bill Label** should reference something like "Loan Amount".

Copyright © 2026, Oracle and/or its affiliates.

- **Recurring Charge** is required (it holds the periodic payment amount).
- **Recurring Chg Amt Label** should reference something like "Payment Amount".
- **Recurring Charge Frequency** is required (it defines the periodicity associated with the recurring charge amount) and must be the same as the bill period frequency or the anniversary billing frequency (based on your **Use Calendar Billing** option).

SA Type - Rate

Turn off the **Rate Required** switch as loans do not use rates.

SA Type - SP Type

Turn off the **Service Points Required** switch as loans do not have service points.

SA Type - Adjustment Profile

Adjustment Type Profile should reference the profile set up above.

SA Type - Credit and Collections

The credit and collections information should reference a **Severance Process Template** that simply expires the loan. The **Debt Class** and **Write Off Debt Class** should reference an appropriate value consistent with your credit and collections rules. Refer to Defining Credit & Collections Options for more information.

SA Type - Algorithms

The Loan Schedule, Loan Interest Charge, Loan Periods, and Amounts and Payment Freeze algorithms defined above must be set up.

Start Options

Loans require a start option to define the adjustment type that is used to create the adjustment to book the initial principal amount when the loan is activated. Loan start options can also specify default values for loan amount, payment amount, and number of periods.

Create at least one start option for each loan SA type. The following information should be defined:

- **Adjustment Type** should reference the adjustment type defined above to book principal.
- **Recurring Charge Amount** (Payment Amount) should only be defined if you have standard loan payments.

Copyright © 2026, Oracle and/or its affiliates.

- **Total Amount To Bill** (Loan Amount) should only be defined if you have standard loan amounts.
- **Number of Payment Periods** should be the number of payment periods in the loan (if you have a standard loan period).
- If you want to define a set of standard APRs for a loan SA Type, set up a **Contract Value** on the Rate Info page. The **Bill Factor** should match the interest bill factor specified on the SA Type.

Defining Non-Billed Budget Options

A Non-Billed Budget (NBB) is a payment plan that allows your customers to pay set amounts at specified intervals (e.g. every two weeks). Non-Billed Budgets are typically used when your company bills on an infrequent basis and you want to provide your customers with a mechanism to make smaller payments more frequently. As the name suggests, bills are not created for the Non-Billed Budget's scheduled payments; customers must remember to make their payments at the scheduled intervals. The topics in this section describe how to design and set up non-billed budgets.

> **Note**
>
> **Non-Billed Budgets are optional.** The system configuration requirements described in this section are only relevant if your organization offers Non-Billed Budgets.

- What Is A Non-Billed Budget?
- The Financial Impact Of Non-Billed Budgets
- Financial Transactions For Unmonitored Non-Billed Budgets
- Designing Non-Billed Budgets
- Non-Billed Budget Recommendation Rule
- Setting Up The System To Enable Non-Billed Budgets

What Is A Non-Billed Budget?

A Non-Billed Budget is a special type of budget or payment plan that encompasses three major elements:

- A set of scheduled payments
- The business rules used to recommend and potentially renew the payment schedule
- The business rules that govern the financial impact on the current and payoff balances of the SAs covered by the payment schedule

Non-Billed Budgets are managed via a service agreement whose SA type has a special role of Non-Billed Budget. If an SA type has a special role of Non-Billed Budget it must have one or more recommendation rules, and SAs of that type are allowed to have payment schedules.

Copyright © 2026, Oracle and/or its affiliates.

The Financial Impact Of Non-Billed Budgets

When you set up the Non-Billed Budget SA type, you can indicate whether the Non-Billed Budget is monitored by the account debt monitor process. The following sections contain examples of financial transactions for Non-Billed Budgets that are monitored by the account debt monitor.

> **Note**
>
> **Unmonitored Non-Billed Budgets.** SAs that are covered by unmonitored Non-Billed Budgets are subject to different financial treatment than those that are monitored. The financial transactions relevant for unmonitored Non-Billed Budgets are discussed later.

- Current Amount For SAs Covered By A Non-Billed Budget
- Scheduled And Actual Payments On The Non-Billed Budget
- Overpayments for Non-Billed Budgets
- Underpayments For Non-Billed Budgets
- Billing For SAs Covered By The Non-Billed Budget
- Distributing Non-Billed Budget Credit
- Stopping an SA Covered By a Non-Billed Budget

Current Amount For SAs Covered By A Non-Billed Budget

As described under Current Amount versus Payoff Amount , the values of the current balance and payoff balance are the same for most financial transactions. One exception is for SAs that are covered by a Non-Billed Budget in which case the current balance is always zero and the payoff balance is always the amount the customer really owes.

When a Non-Billed Budget is activated or an SA is added to a Non-Billed Budget, the system creates adjustments for all affected SAs to set the current amount equal to zero. The adjustment type that references the Payoff Amt = 0 / Current Amt = Adj Amount (no GL) algorithm is taken from the SA's SA type.

> **Note**
>
> **Non-Billed Budgets and Open Item Accounts.** If the Non-Billed Budget is for an open item account, no match event is created for the financial transaction that reduces the current amount to zero. This FT must be manually matched if required.

The following example shows the effect of activating a Non-Billed Budget that covers an electric SA with a current balance of 25. The transactions for the electric SA are illustrated on the right and the transactions for the non-billed budget SA are illustrated on the left.

Copyright © 2026, Oracle and/or its affiliates.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				

By setting the current balance for the covered SAs to zero, the SAs are insulated from the regular debt monitoring process. Depending on the Non-Billed Budget recommendation algorithm , the payoff balance can be ignored, divided evenly between the scheduled payments or added to the first scheduled payment.

Activating a Non-Billed Budget has no affect on its own current or payoff balances.

Scheduled And Actual Payments On The Non-Billed Budget

When a scheduled payment is due, an adjustment is created to increase the Non-Billed Budget's current balance by the expected amount. The current balance on the SA can be monitored to ensure that payments are being made on time.

When the payment is made, the Non-Billed Budget's current balance is reduced to zero and the Non-Billed Budget's payoff balance reflects the accumulated credit from the payment. This accumulated credit is transferred to the covered SAs when the next bill for the account is completed.

The following example illustrates scheduled and actual payments for the Non-Billed Budget in the previous example. The amount of the scheduled Non-Billed Budget payments is 10. Note that the first two transactions were described above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0
Payment					-10	-10	0	-10

An algorithm (Process NBB Scheduled Payment) plugged in on the Non-Billed Budget SA type creates the appropriate financial transactions when the scheduled payment is due. The

Copyright © 2026, Oracle and/or its affiliates.

algorithm is called by the Non-Billed Budget Scheduled Payment Processing (NBBPS) background process.

The normal payment processing handles the adjustments created when the payment is made.

> **Note**
>
> **Non-Billed Budget Payment Cancellation.** If a payment is canceled, the financial transaction is reversed.
>
> **Automatic Payments And Non-Billed Budgets.** Users may set up a Non-Billed Budget to use automatic payments by setting up the account's auto pay options. Users may also exclude the Non-Billed Budget from automatic payment if the account is set up for automatic payment. The NBB Scheduled Payment Automatic Payment Create background process calls the Auto Pay Creation algorithm to create the auto payment for a Non-Billed Budget.

Overpayments for Non-Billed Budgets

Typically, payments in excess of the Non-Billed Budget's current balance are credited to an overpayment (excess credit) SA. When the next adjustment is created for a scheduled payment, the credit on the overpayment SA is used to relieve the non-billed budget current balance.

> **Note**
>
> **Overpayment Distribution Algorithm.** The overpayment distribution is a function of the overpayment distribution algorithm plugged in on the account's customer class. We strongly recommend that the Non-Billed Budget SA type be set up so that overpayment is not allowed. Any excess payments should go to an overpayment SA. For more information about overpayment distribution, refer to Overpayment Segmentation.

In the example below, the customer pays 20 for a scheduled payment instead of 10. The non-billed budget SA is illustrated on the right and the overpayment SA is illustrated on the left. The electric SA is not illustrated in the example below because scheduled payments, payments and overpayments have no effect on covered SAs. The first four transactions were illustrated above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Non-Billed Budget SA				Overpayment SA			
Starting balance								
Non-Billed Budget is activated								

Copyright © 2026, Oracle and/or its affiliates.

First scheduled payment is due	10	0	10	0				
Payment	-10	-10	0	-10				
Second scheduled payment is due	10	0	10	-10				
Over-payment	-10	-10	0	-20	-10	-10	-10	-10
Third scheduled payment is due	10	0	10	-20				
Transfer Adjustment	-10	-10	0	-30	10	10	0	0

After the Process NBB Scheduled Payment algorithm creates the next scheduled payment, it looks for a credit amount on the overpayment SA and creates an adjustment to transfer the credit balance (or the amount of the payment if the credit is more than the scheduled payment amount) from the overpayment SA to the non-billed budget SA. The overpayment transfer adjustment type and the overpayment SA type are specified as parameters to the algorithm.

Underpayments For Non-Billed Budgets

An insufficient payment or a canceled payment leaves a current balance on the Non-Billed Budget SA.

In the example below, the customer makes a payment of 7 for the fourth scheduled payment. The example below does not show the overpayment SA (illustrated above). The first eight transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0

Copyright © 2026, Oracle and/or its affiliates.

Payment	-10	-10	0	-10
Second scheduled payment is due	10	0	10	-10
Over-payment	-10	-10	0	-20
Third scheduled payment is due	10	0	10	-20
Transfer Adjustment	-10	-10	0	-30
Fourth scheduled payment	10	0	10	-30
Underpayment	-7	-7	3	-37

The Process NBB Scheduled Payment algorithm creates a trigger to ensure that the current balance is monitored by the account debt monitor. Refer to Credit And Collections And Non-Billed Budgets for more information.

Billing For SAs Covered By The Non-Billed Budget

When the next bill for the account is completed, the credit on the Non-Billed Budget is transferred to the covered SAs. The credit is prorated over the covered SAs according to the relative payoff balances on each SA.

In the example below, the electric SA's bill is 33. The first ten transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0

Copyright © 2026, Oracle and/or its affiliates.

Payment					-10	-10	0	-10
Second scheduled payment is due					10	0	10	-10
Over-payment					-10	-10	0	-20
Third scheduled payment is due					10	0	10	-20
Transfer Adjustment					-10	-10	0	-30
Fourth scheduled payment					10	0	10	-30
Underpayment					-7	-7	3	-37
Bill	0	33	0	58				
Bill completion (transfer adjustment)	0	-37	0	21	0	37	3	0

When a bill segment financial transaction is created, the current amount is set to zero for SAs that are covered by the Non-Billed Budget (if you use a bill segment FT algorithm of Payoff Amt = Bill Amt / Current Amt = Amt Due). Though not evident by the name, the Payoff Amt = Bill Amt / Current Amt = Amt Due algorithm does set the current amount to zero for monitored Non-Billed Budgets. (For SAs with other roles, the current amount is equal to the amount due or the recurring charge.)

A bill completion algorithm transfers money from the Non-Billed Budget to the covered SAs (if you plug in the NBB Credit Transfer bill completion algorithm on the Non-Billed Budget SA type). The algorithm type supplied with the base package distributes the credit using the method described in Distributing Non-Billed Budget Credit.

> **Note**
>
> **Canceled Bill Segments.** No new processing occurs when a bill segment is canceled; any credit balance remains on the covered SA.

Distributing Non-Billed Budget Credit

Copyright © 2026, Oracle and/or its affiliates.

Both the NBB Credit Transfer bill completion algorithm type and the Non-Billed Budget SA stop algorithm type supplied with the base package use the same method of distributing a credit from a Non-Billed Budget SA to the covered SAs. The following points describe how the credit is distributed:

- Covered SAs that are already in credit (due to some other circumstance, such as a cancellation and rebill) are excluded from the distribution.
- The distribution to each covered SA will not exceed its total payoff to ensure that none of the covered SAs have a credit balance.
- The credit is prorated over the covered SAs according to the relative payoff balances on each SA.
- The calculation of the payoff balance is adjusted to exclude the current balance to ensure that the credit is prorated over the debt covered by the budget, not any ad-hoc debt for the SA.

> **Note**
>
> The current balance on covered SAs should always be zero. The only exception occurs if an adjustment has been added that directly affected the SA balance. In this case, the distribution assumes that the balance is outside the non-billed budget and needs to be paid separately.

- Any excess credit remains on the Non-Billed Budget SA until the next distribution takes place or until the Non-Billed Budget SA is stopped.
- The type of adjustments created is determined by the **Adjustment Type (Xfer)** specified on the Non-Billed Budget SA type.

The examples in the table below illustrate the points above.

NBB SA Credit	SA 1 Payoff	SA 1 Current	SA 2 Payoff	SA 2 Current	SA 1 Credit Xfer	SA 2 Credit Xfer
-100.00	-100.00	0.00	-200.00	0.00	0.00	0.00
-100.00	150.00	0.00	-50.00	0.00	-100.00	0.00
-300.00	150.00	0.00	50.00	0.00	-150.00	-50.00
-100.00	150.00	0.00	250.00	0.00	-37.50	-62.50
-100.00	150.00	0.00	250.00	50.00	-42.86	-57.14

The first example above shows that covered SAs with a credit balance are excluded from the distribution. Any excess credit remains on the Non-Billed Budget.

The second example shows that one covered SA has a credit balance, so the entire credit is distributed to the remaining SA.

The third example shows the amount distributed to a covered SA does not exceed its payoff balance. Again, any excess credit remains on the Non-Billed Budget SA.

The fourth example illustrates how the credit is prorated based on the payoff balance. The fifth example illustrates the same prorating but with a current balance on one of the SAs (SA 2). (Remember that the prorating excludes any current balance.)

Copyright © 2026, Oracle and/or its affiliates.

The prorated amount is calculated by subtracting the current balance from the payoff balance then multiplying the result by the distribution amount and dividing by the total payoff owing of all covered SAs.

Stopping an SA Covered By a Non-Billed Budget

If a service agreement covered by a Non-Billed Budget is stopped, the system must bring the current balance and payoff balance of the covered SA back in synch. The system creates an adjustment using the Synch Current adjustment type from the SA's SA type.

In the example below, the electric SA's payoff balance is 21. The first twelve transactions are discussed above.

Event	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance	Effect On Current Balance	Effect On Payoff Balance	Current Balance	Payoff Balance
	Electric SA				Non-Billed Budget SA			
Starting balance	0	0	25	25				
Non-Billed Budget is activated	-25	0	0	25				
First scheduled payment is due					10	0	10	0
Payment					-10	-10	0	-10
Second scheduled payment is due					10	0	10	-10
Over-payment					-10	-10	0	-20
Third scheduled payment is due					10	0	10	-20
Transfer Adjustment					-10	-10	0	-30
Fourth scheduled payment					10	0	10	-30
Underpayment					-7	-7	3	-37
Bill	0	33	0	58				

Copyright © 2026, Oracle and/or its affiliates.

Bill completion (transfer adjustment)	0	-37	0	21	0	37	3	0
SA is stopped	21	0	21	21				

> **Note**
>
> In addition to synching the current and payoff balance, the SA being stopped is removed from the covered SAs collection. When the last covered SA is removed from the collection, the Non-Billed Budget is also stopped.

Financial Transactions For Unmonitored Non-Billed Budgets

Some companies have the concept of non-billed budgets where the payments made by the customer are optional. This functionality is implemented as an unmonitored Non-Billed Budget. Unmonitored Non-Billed Budgets allow a customer to make optional prepayments towards a bill.

As explained previously, unmonitored Non-Billed Budgets receive different financial treatment than monitored Non-Billed Budgets. The financial transaction algorithms use the **Non-Billed Budget Monitor** flag on the Non-Billed Budget's SA type to create the appropriate financial transactions. The following table describes the differences in the financial treatment of monitored and unmonitored Non-Billed Budgets.

System Event	Monitored Non-Billed Budgets	Unmonitored Non-Billed Budgets
When a Non-Billed Budget is activated or an SA is added to a Non-Billed Budget...	The system creates adjustments for all affected SAs to set the current balance equal to zero.	The current balances for covered SAs are not zeroed. The activate SA and Non-Billed Budget maintenance processing do not create any adjustments if the Non-Billed Budget is unmonitored.
When a scheduled payment is due...	An adjustment is created to increase the Non-Billed Budget's current balance by the expected amount.	There is no change to the Non-Billed Budget's current balance; it is always zero. The scheduled payment processing background process does not execute the NBB process scheduled payment algorithm.
When the payment is made...	The Non-Billed Budget's current balance is reduced to zero and the Non-Billed Budget's payoff balance reflects the accumulated credit from the payment.	Typically the payments are distributed to an excess credit (overpayment) SA. Refer to Distributing Payments for Unmonitored Non-Billed Budgets for more information.

Copyright © 2026, Oracle and/or its affiliates.

When a bill segment financial transaction is created...	The current amount is set to zero for SAs that are covered by the Non-Billed Budget.	The covered SAs' current amounts are equal to their payoff amounts.
At bill completion...	Money from the Non-Billed Budget is transferred to the covered SAs. Customers may receive a bill for information purposes, but they are not required to pay it.	Money from the overpayment SA is transferred to the account's SAs. The customer is still liable to pay the outstanding balance.
When the Non-Billed Budget is stopped or an SA is removed from a Non-Billed Budget...	The system creates adjustments for all affected SAs to synchronize their current balances with their payoff balances, thus removing the zero current balance and replacing it with the actual current balance.	There is no change to the current balances for covered SAs as the balances already reflect the actual current balance.

Unmonitored Non-Billed Budgets also receive different credit and collections treatment. Refer to Credit and Collections and Unmonitored Non-billed Budgets for more information.

- Distributing Payments for Unmonitored Non-Billed Budgets
- Transferring Credit from Unmonitored Non-Billed Budgets

Distributing Payments for Unmonitored Non-Billed Budgets

For Non-Billed Budgets, payments are distributed according to the payment and overpayment algorithms on the customer class. The base package payment distribution algorithm applies a payment first towards any SAs that have overdue or current balances (refer to Distributing A Payment for more information). Since the unmonitored Non-Billed Budget SA doesn't have a current balance, it is not considered by the payment distribution algorithm. If there aren't any SAs with a current balance, the overpayment distribution algorithm handles the remaining credit. You can elect to:

- Apply the overpayment to an excess credit SA. This is the method that we strongly recommend because all financial transactions are then a function of the normal payment, overpayment and billing processes.
- Apply the overpayment to the highest priority SA that is eligible for overpayment (as specified on the SA type). You can use this method to apply the overpayment to the unmonitored Non-Billed Budget SA. If you use this method, you must also set up the system to transfer the credit from the unmonitored Non-Billed Budget.

> **Note**
>
> **Use An Excess Credit SA.** We strongly recommend that payments for unmonitored Non-Billed Budgets are distributed to an excess credit SA. In this case, the Non-Billed Budget is just a shell to hold the covered SAs and recommend a payment schedule; all financial transactions are a function of the normal payment, overpayment and billing processes.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to Overpayment Segmentation for a detailed discussion of overpayment distribution options.

Transferring Credit from Unmonitored Non-Billed Budgets

If you distribute an overpayment to an unmonitored Non-Billed Budget SA (i.e. the unmonitored non-billed budget maintains a credit balance instead of an overpayment SA), you must plug-in a bill completion algorithm on the SA type to transfer the credit balance to the covered SAs at bill completion time. The bill completion algorithm type (BCMP-NB) supplied with the base package transfers the credit balance to the covered service agreements when the bill is completed. Additionally, the **Adjustment Type (Xfer)** on unmonitored Non-Billed Budget SA types should reference a FT algorithm of Payoff Amt = Adj / Current Amt = Adj to ensure that the credit is removed from both the current and payoff balances.

Warning

You must create your own SA stop algorithm type for correctly stopping an unmonitored non-billed budget that maintains a credit balance. The SA stop algorithm that is supplied with the base package does NOT transfer remaining credit from the unmonitored Non-Billed Budget SA. (The base package SA stop algorithm transfers the remaining credit using the overpayment distribution algorithm on the customer type, which you have set up to transfer to back to the unmonitored non-billed budget.)

Designing Non-Billed Budgets

The topics in this section describe functionality that you must consider when designing non-billed budgets.

- Making SAs Eligible For Non-Billed Budgets
- Designing Recommendation Rules

Making SAs Eligible For Non-Billed Budgets

A billable SA may be covered by a Non-Billed Budget when its SA type is flagged as Eligible for Non-Billed Budget.

All SAs that are eligible for non-billed budgets should reference a bill segment type that uses the Payoff Amt = Bill Amt / Current Amt = Amt Due bill segment FT algorithm. This algorithm sets the SA's current amount to zero if it is covered by a monitored Non-Billed Budget.

A list of the SAs covered by a Non-Billed Budget is maintained with the non-billed budget SA. This list is used at bill completion to determine the financial transactions that should occur.

Copyright © 2026, Oracle and/or its affiliates.

Designing Recommendation Rules

Users (and the renewal process) ask the system to recommend the scheduled payments for a Non-Billed Budget. In general, this recommendation process must establish:

- The amount to be paid
- The dates on which the payments are due

We envision many different types of recommendation rules. For example:

- Recommend 26 scheduled payments to be made on a fortnightly basis that are due on Tuesdays.
- Recommend monthly payments that are due on the nearest workday after the 10th of the month.
- Recommend scheduled payments where the customer pays their annual charges in 10 out of 12 months where the payments are not due in November and December.
- Recommend bi-monthly payments where the payments are due on the first workday following the 5th and 20th of the month.

Additionally, the recommendation rules must determine how to handle any outstanding payoff balances for covered SAs. The true-up rule provided with the base package payment schedule algorithm type can ignore the payoff balance, divide the payoff balance evenly between the scheduled payments, or add the payoff balance to the first scheduled payment.

A recommendation rule comprises three elements:

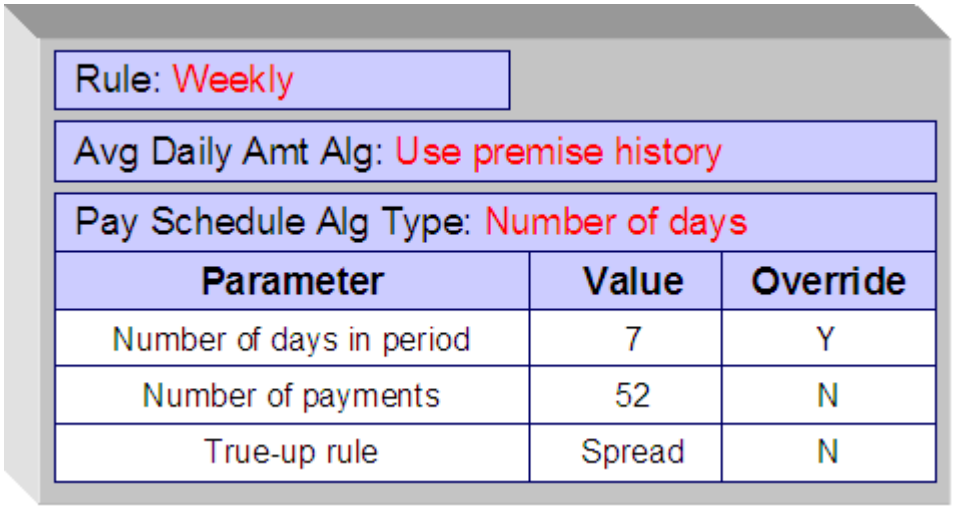
- An algorithm to calculate an average daily amount (the NBDA-DA algorithm type provided with the base package uses premise billing history)

> **Note**
>
> **Calculating an Amount for Non-utility SAs.** The algorithm type supplied with the base package only handles service-point oriented SAs. For example, the average daily amount algorithm calculates an average daily amount. For non-utility SAs, you must develop the appropriate algorithm types.

- Two algorithm types are provided with the base package to calculate a schedule of payments. NBPS-MON calculates a monthly schedule and NBPS-PS calculates a scheduled based on a specified number of days.
- A collection of default parameter values for the payment schedule algorithm type

A recommendation rule (based on the algorithm types provided with the base package) is illustrated below.

Copyright © 2026, Oracle and/or its affiliates.



Rule: Weekly

Avg Daily Amt Alg: Use premise history

Pay Schedule Alg Type: Number of days

Parameter	Value	Override
Number of days in period	7	Y
Number of payments	52	N
True-up rule	Spread	N

Weekly Payments Recommendation Rule

> **Note**
>
> **Additional Parameters.** Not all of the parameters associated with the weekly payment schedule algorithm type are illustrated. Refer to the NBPS-PS algorithm type for a detail description of the parameters.

The default parameter values for the payment schedule algorithm type may change over time, so the collection contains an effective date. If default values are changed, these changes do not affect Non-Billed Budgets already in effect. Existing Non-Billed Budgets keep the parameter values that were used when the Non-Billed Budget was started.

A user may override the default parameter values for the payment schedule algorithm type to customize the schedule if an override is allowed for a parameter. Additionally, a user may edit the payment schedule details at any time (provided the payment has not yet been processed).

The parameter values used for the recommendation rule are kept with the Non-Billed Budget SA, so that any customized parameter changes can be re-applied to a renewed Non-Billed Budget. For example, the parameter that determines the payment due day may default to the first of the month. To customize the schedule, this value may be changed to the fifth of the month. This amended value is kept with the Non-Billed Budget SA to ensure that the renewed budget follows the same monthly schedule.

> **Note**
>
> Normally parameter values for an algorithm type are kept with the algorithm. Because the parameters may vary for each Non-Billed Budget, the parameter values are kept with the SA in the case of Non-Billed Budgets.

Refer to Non-billed Budgets Recommendation Rule - Main for information on creating recommendation rules.

- Example Recommendation Rules
- Activating Non-Billed Budgets

Copyright © 2026, Oracle and/or its affiliates.

- Renewing Non-Billed Budgets
- Expiring Non-Billed Budgets
- Stopping Non-Billed Budgets
- Automatic Payment and Non-Billed Budgets
- Credit and Collections and Non-Billed Budgets
- Credit and Collections and Unmonitored Non-Billed Budgets
- Non-Billed Budget Status
- Alerts For Non-Billed Budgets

Example Recommendation Rules

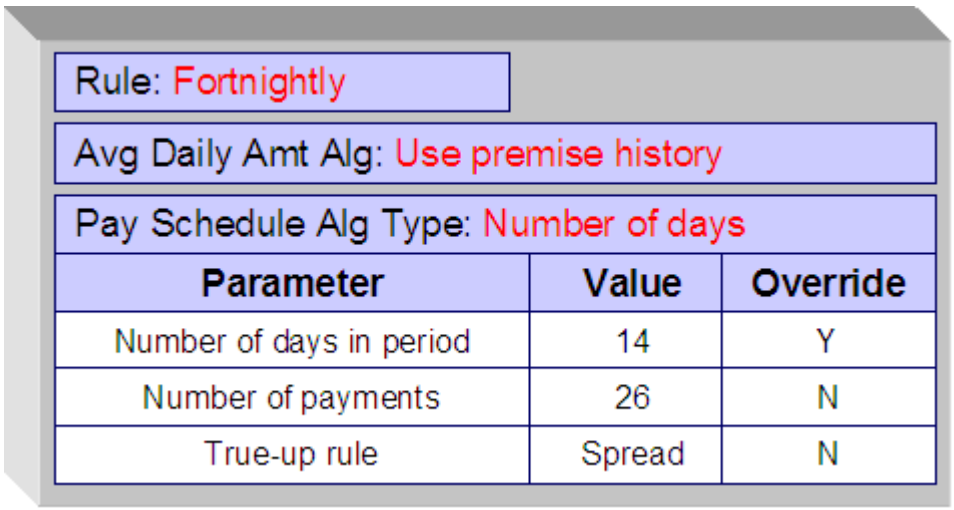
The following examples may be helpful in designing and implementing your recommendation rules.

> **Note**
>
> **Developing Your Own Payment Schedule Algorithms.** The base package comes supplied with a monthly payment schedule algorithm type. You can use this algorithm as an example when creating payment schedule algorithm types for your implementation.

- Fortnightly Payments Recommendation Rule Example
- Monthly Payments Recommendation Rule Example
- Ten Out of Twelve Months Recommendation Rule Example

Fortnightly Payments Recommendation Rule Example

The following diagram illustrates a recommendation rule where the customers pay every two weeks. The current balance for any covered SAs is added to the first payment.

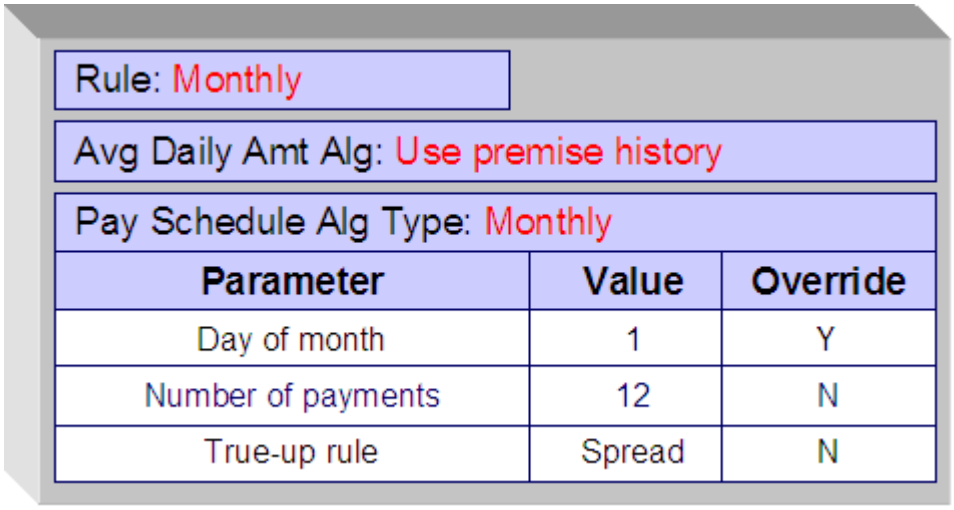


Copyright © 2026, Oracle and/or its affiliates.

Fortnightly Payments Recommendation Rule

Monthly Payments Recommendation Rule Example

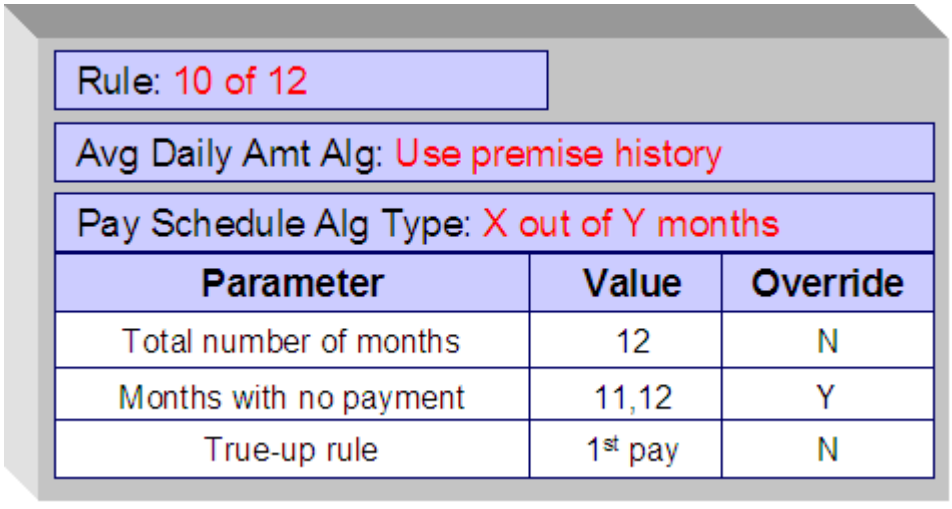
The following diagram illustrates a recommendation rule where the customers pay twice monthly on the first of the month. The current balance for any covered SAs is spread out over the scheduled payments.



Monthly Payments Recommendation Rule

Ten Out of Twelve Months Recommendation Rule Example

The following diagram illustrates a recommendation rule where the customers pay one a month except for months during a holiday season (November and December). The current balance for any covered SAs is added to the first payment.



Ten Out of Twelve Months Recommendation Rule

Copyright © 2026, Oracle and/or its affiliates.

Activating Non-Billed Budgets

You can plug in an algorithm (of type SACR-AT) on the SA Creation system event on the Non-Billed Budget SA type to automatically activate the Non-Billed Budget (i.e. transition it from pending start to active status). If you don't use a SA Creation algorithm to activate the Non-Billed Budget, it is activated the next time the SA activation background process runs.

When a Non-Billed Budget is activated, you can perform special processing using an algorithm plugged in on the SA Activation system event on the Non-Billed Budget SA type. The special processing can be developed to do anything that you would like, for example you could:

- Create a customer contact that with an appropriate letter template can generate a letter to inform the customer of their payment amount and payment schedule.
- Initiate the creation of a payment coupon book for a customer.

The system comes supplied with a sample algorithm type (called SAAT-CC) that simply creates a customer contact to indicate that the Non-Billed Budget is activated.

Renewing Non-Billed Budgets

A Non-Billed Budget can be renewed either manually or via a background process. When the Non-Billed Budget SA is created, the expiration date, renewal date and the recommendation rule used to create the initial budget are kept with the SA. A renewal flag on the Non-Billed Budget SA type controls if a renewal is required, optional or not allowed. If renewal is required, a user must specify a renewal date when creating the service agreement. The renewal date is defaulted on to an SA based on the valued of the **Days Before Expiration for Renewal** field on the SA type.

An algorithm on the SA type can customize the processing required to renew an SA.

The SA renewal background process:

- Executes the SA renewal algorithm (specified on the SA type) when the renewal date is reached (i.e., it is on or before the process date). The base package comes with an algorithm type (SARN-NB) that determines the current recommendation rule for a non-billed budget and executes the associated payment schedule algorithm using the Non-Billed Budget SA-specific parameter values to generate a new schedule. It returns new expiration and renewal dates.
- If the renewal algorithm is successful, the renewal and expiration date fields on the SA are updated with the new values.
- If the renewal process is not successful, a To Do list entry (of type TD-SARN) is created for the account and SA.

The new payment schedule that is returned from the renewal process for a Non-Billed Budget is appended to the current schedule.

A user can manually launch the renewal process for a Non-Billed Budget SA by clicking the **Renew NBB** button on the Non-Billed Budget maintenance page .

Expiring Non-Billed Budgets

Copyright © 2026, Oracle and/or its affiliates.

Non-Billed Budget service agreements may specify an expiration date. The SA expiration background process initiates the stop process for all pending start or active SAs where the expiration date is reached (before or on the process date).

Stopping Non-Billed Budgets

When a Non-Billed Budget stop is initiated, either on request or because it has expired and is not being renewed, the Non-Billed Budget is transitioned to pending stop status. You can plug in an algorithm (of type SAIS-ST) on the SA Stop Initiation system event on the Non-Billed Budget SA type to automatically finalize and stop the SA (i.e. transition it to stopped status). If you don't use a SA Stop Initiation algorithm, the Non-Billed Budget is stopped the next time the SA activation background process runs.

To finalize a pending stop SA, the system first calls the stop SA algorithm plugged-in on the SA Stop system event on the SA type. The stop SA algorithm type (SAST-NB) supplied with the base package:

- Distributes any credit on the Non-Billed Budget to the covered SAs (using the method described in Distributing Non-Billed Budget Credit)
- Distributes any excess credit remaining on the non-billed budget using the overpayment distribution algorithm for the account's customer class and the overpayment transfer adjustment type (specified as a parameter to the algorithm)
- Creates a trigger to cause the account to be reviewed by the account debt monitor
- Creates a customer contact (if the customer contact class and customer contact type parameters are populated)

> **Warning**
>
> If you do not plug in an SA stop algorithm that transfers the credit balance from the non-billed budget to its covered SAs (or an excess credit SA), the stopped Non-Billed Budget may have a credit balance. You must then manually distribute this credit.

After the SA stop algorithm is finished, the SA stop processing performs the following steps if the SA type has a special role of Non-Billed Budget:

- If the Non-Billed Budget is monitored, create adjustments to synchronize the current and payoff balance of covered SAs using the **Adj. Type (Synch Current)** adjustment type from the covered SAs' SA types
- Remove the covered SAs from the Non-Billed Budget
- Remove all the scheduled payments from the non-billed budget
- Create an adjustment to synchronize current and payoff on the Non-Billed Budget SA using the **Adj. Type (Synch Current)** adjustment type from the Non-Billed Budget's SA type

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Synchronizing current and payoff effectively sets the current amount to zero on the Non-Billed Budget SA, as the payoff amount should have been reduced to zero by the distribution and overpayment processing in the algorithm for SA Stop.

> **NOT_SUPPORTED**
>
> Refer to The Lifecycle Of A Service Agreement for more information about how a pending stop SA is stopped and closed.

Automatic Payment and Non-Billed Budgets

If a customer wants to pay their non-billed budget scheduled payments automatically, the account must be set up for automatic payment. In addition, the Non-Billed Budget must indicate that automatic payment is being used.

> **NOT_SUPPORTED**
>
> Refer to How To Set Up Automatic Payment For A Non-Billed Budget for more information.

When this is done, a background process referred to as NBBAPAY creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

> **Note**
>
> You must ensure that your auto pay creation algorithm supports Non-Billed Budget scheduled payments. The APAY-CREATE algorithm type supplied with the base package supports non-billed budget scheduled payments.

Credit and Collections and Non-Billed Budgets

Unless the Non-Billed Budget is unmonitored, the account debt monitor (ADM) monitors a Non-Billed Budget SA's current amount just as it does for any other SA. The scheduled payment algorithm creates a trigger to ensure that the account debt monitor reviews the account the next time it runs. The review date on the trigger record is set to the process business date.

A separate debt class is needed for Non-Billed Budget SA types, thus allowing you to define collection class controls, debt criteria and collection process templates specifically for Non-Billed Budgets. The debt criteria should be set up to trigger a collection process when the arrears amount exceeds $0.01 for more than n payment periods plus the number of grace days that you want to allow.

Copyright © 2026, Oracle and/or its affiliates.

The collection process template can perform any of the events in standard collection processes, such as sending letters to customers and creating severance processes. At a minimum, the collection process template should be set up to start a severance process for all service agreements in the debt class. (Since the debt class is specifically for Non-Billed Budgets, the Non-Billed Budget is the only SA that will be subject to a severance process.)

The Non-Billed Budget severance process template should include the following event types:

- Populate a characteristic on the SA to indicate that the SA is broken. The base package comes supplied with a severance event algorithm type (SVEV-NB) that sets an SA characteristic to indicate that it was "severed".
- Expire Severance Agreement to move the SA to the pending stop state.

When the system subsequently stops the Non-Billed Budget, the system removes the covered SAs from the Non-Billed Budget and synchronizes their current balances with their payoff balances. Since the SAs have current balances again, they are subject to the account debt monitor, which can start subsequent collection processes for any of the SAs that meet the debt criteria for their debt class.

NOT_SUPPORTED

Refer to Stopping Non-billed Budgets for a complete description of the events that occur when a Non-Billed Budget is stopped.

Customers can catch up on their payments and avoid having their Non-Billed Budget broken as long as their current balance doesn't violate the debt criteria for the Non-Billed Budget's collection process.

Credit and Collections and Unmonitored Non-Billed Budgets

If the **Non-Billed Budget Monitor** flag on the Non-Billed Budget's SA type is set to Unmonitored, the NBB Scheduled Payment Processing background process does not create a trigger for the account debt monitor. Additionally, the Non-Billed Budget's current amount is always equal to zero, so it never violates any debt collection criteria. We recommend using a debt class that has the **Eligible for Collection** flag turned off, such as the N/A debt class.

For unmonitored Non-Billed Budgets, the current balance is kept on the covered SAs so they are subject to the account debt monitor and any debt criteria for their SA types' debt classes. For more information, refer to Financial Transactions for Unmonitored Non-Billed Budgets.

Non-Billed Budget Status

A Non-Billed Budget SA's status is just like any other SA's status. In addition, you can use a characteristic to keep an explicit status relevant to the Non-Billed Budget:

- A base package severance event algorithm type (SVEV-NB) creates a characteristic value to indicate if a Non-Billed Budget is "broken" (i.e. stopped as a result of a severance process).
- A base package break Non-Billed Budget algorithm type (NBBR-BRK) creates a characteristic value to indicate if a Non-Billed Budget is "canceled" (i.e. manually stopped by a user).

Copyright © 2026, Oracle and/or its affiliates.

An active Non-Billed Budget implicitly has a "kept" status (i.e. all scheduled payments have been made).

Alerts For Non-Billed Budgets

The system provides alerts to highlight the existence of Non-Billed Budgets. These alerts are important to assist the customer service representatives:

- An alert is displayed if the account has a Non-Billed Budget that is not stopped (e.g., pending start, active or pending stop).
- When a user denies a Non-Billed Budget (for whatever reason), the user should create a customer contact with a given customer contact type. This type of alert prevents the customer from shopping around. An existing alert algorithm type (CC BY TYPCL) can highlight these customer contacts.
- For customers who are permanently forbidden from having a Non-Billed Budget, the user should put a permanent alert on the account.
- Use an algorithm to highlight cancelled or severed non-billed budgets with an entry in the alert zone. The algorithm type to do this is not provided. Use PP BY STATUS and CCAL-WF as examples of how to create this type of algorithm.

> **NOT_SUPPORTED**
>
> For more information about introducing alert conditions on Control Central, refer to Installation Options - Algorithms.

Non-Billed Budget Recommendation Rule

Recommendation rules are used to recommend scheduled payments for Non-Billed Budgets. For information about designing recommendation rules, refer to Designing Recommendation Rules.

To define recommendation rules, navigate to **Admin**, and then **Financial**, and then **NBB Recommendation Rule**, and then **Add.**

Description of Page

Enter an easily recognizable **Recommendation Rule** code and **Description** for each recommendation rule.

Specify the **Average Daily Amount Algorithm** used to calculate the average daily amount for this recommendation rule.

Specify the **Payment Schedule Algorithm Type** used to create the recommended payment scheduled for Non-Billed Budgets that use this recommendation rule. The Payment Schedule Algorithm Type cannot be modified if a Non-Billed Budget SA that is not stopped or cancelled is using this recommendation rule.

Payment Schedule Parameters enables you to define collections of default parameter values for the payment schedule algorithm type that are effective dated. For each collection:

Copyright © 2026, Oracle and/or its affiliates.

- **Effective Date** defines the date on which the collection of parameter values becomes effective.
- **NBB Rule Parameter Value** specifies the default value of each parameter supplied to the algorithm. Note that the payment schedule algorithm type controls the number and type of parameters.
- **Override** indicates whether the user can override the default value for the parameter.

Setting Up The System To Enable Non-Billed Budgets

The above topics provided background information about how Non-Billed Budgets are supported in the system. The topics in this section describe how to set up the system to enable Non-Billed Budget functionality.

> **Note**
>
> **Example Setup.** This section describes typical non-billed budget configurations. Your set up and configuration may differ depending on your business needs. This section is provided for guidance only. Read the descriptions of Non-Billed Budget functionality above to understand the implications of the described setup.

- NBB Distribution Codes
- NBB Adjustment Types
- NBB Adjustment Type Profiles
- NBB Characteristic Types
- NBB Customer Contact Class And Types
- NBB Algorithms
- NBB Debt Class And Collection Process
- SA Types for SAs Covered by NBBs
- NBB Recommendation Rules
- Non-Billed Budget SA Types
- NBB Background Processes

NBB Distribution Codes

You must set up a Non-Billed Budget Clearance distribution code that is used on the Non-Billed Budget SA type to credit non-billed payments.

NBB Adjustment Types

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Non-Billed Budget Financial Transaction Algorithms.** The Non-Billed Budget adjustment types use the standard FT algorithm types that are provided with the base package. If you have not yet defined algorithms for these types in your system, do so before creating the Non-Billed Budget adjustment types.

The following adjustment types are needed:

- Add SA To Non-Billed Budget. This adjustment type should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Because the adjustment affects the current balance only, there is no entry in the GL. This adjustment type is referenced on **Adjustment Type (Current=0)** on SA types that are eligible for Non-Billed Budgets. Note that this adjustment type is never used if the SA is added to an unmonitored Non-Billed Budget.
- Bill Complete for Monitored Non-Billed Budget. This adjustment type should reference a distribution code used for balance transfer clearing. It should reference a FT algorithm of Payoff Amt = Adj / Current Amt = 0. This adjustment type is referenced on **Adjustment Type (Xfer)** on SA types that are monitored Non-Billed Budgets.
- Synch Balance for Non-Billed Budget. This adjustment type should be set up for **Sync. Current Amount** and should reference a FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). This adjustment type is referenced on the **Adj. Type (Synch Current)** on Non-Billed Budget SA types. It is used when a Non-Billed Budget SA is stopped to synch the current balance with the payoff balance.
- Scheduled Payment. This adjustment type should reference an FT algorithm of Payoff Amt = 0 / Current Amt = Adj Amount (no GL). Because the adjustment affects the current balance only, there is no entry in the GL. This adjustment type is referenced on the Non-Billed Budget process scheduled payment algorithm.
- Overpayment Transfer. This adjustment type should reference a transfer distribution code and a FT algorithm of Payoff Amt = Adj / Current Amt = Adj. This adjustment is referenced on the non-billed budget process scheduled payment algorithm and the stop non-billed budget algorithm. Note that if there is already a transfer adjustment type created in your system, you do not need to create a new one.

If you are setting up an unmonitored Non-Billed Budget that maintains a credit balance (as opposed to maintaining the credit balance on an overpayment SA), you need to create an adjustment type for Bill Complete for Unmonitored Non-Billed Budget. (Refer to Transferring Credit from Unmonitored Non-billed Budgets for more information.) The adjustment type should reference a transfer distribution code and a FT algorithm of Payoff Amt = Adj / Current Amt = Adj. This adjustment type is referenced on **Adjustment Type (Xfer)** on SA types that are unmonitored Non-Billed Budgets.

NBB Adjustment Type Profiles

Create an adjustment type profile for Non-Billed Budgets that references the following adjustment types:

- Bill complete for monitored Non-Billed Budget
- Bill complete for unmonitored Non-Billed Budget (if used)
- Synch balance for Non-Billed Budget

Copyright © 2026, Oracle and/or its affiliates.

- Scheduled payment
- Overpayment transfer

Create an adjustment type profile for eligible SAs that references the following adjustment types:

- Add SAs to monitored Non-Billed Budgets
- Bill complete for monitored Non-Billed Budget
- Bill complete for unmonitored Non-Billed Budget (if used)

> **Note**
>
> **Bill Complete Adjustment Types.** Because the bill complete adjustment types transfer amounts between two SAs, they must be in profiles for both Non-Billed Budget and eligible SA types.
>
> **Overpayment Transfer Adjustment Type.** The overpayment transfer adjustment type created above is used to transfer funds from an excess credit SA to a Non-Billed Budget when the scheduled payment is processed. It is also used to transfer excess funds from a Non-Billed Budget that is being closed to an excess credit SA. The transfer adjustment should therefore be added to an adjustment type profile that is referenced on the excess credit SA type.

NBB Characteristic Types

Create a Non-Billed Budget status characteristic type that specifies Service Agreement as its characteristic entity. The characteristic type should include the following predefined values:

- Non-Billed Budget Canceled
- Non-Billed Budget Severed

NBB Customer Contact Class And Types

Create a Non-Billed Budget customer contact class. The contact class may include the following customer contact types:

- Non-Billed Budget Activate
- Non-Billed Budget Renewal
- Non-Billed Budget Stop

> **Note**
>
> **Customer Contact Letters.** If you want to send letters to your customers when a contact of any of these types is created, you must create an appropriate letter template and attach it to the contact type.

Copyright © 2026, Oracle and/or its affiliates.

NBB Algorithms

You must define the following algorithms:

- On NBB recommendation rule, the Non-Billed Budget Daily Amount Calculation. Refer to the NBDA-DA algorithm type for more information about the base package algorithm.
- On SA type:
  - Non-Billed Budget Process Scheduled Payment. This algorithm has parameters that must reference the Scheduled Payment and Overpayment Transfer adjustment types defined above. Another parameter references an overpayment SA type (that you may need to create if there is not already one in your system). Refer to the NBPA-PS algorithm type for more information about the base package algorithm.
  - Non-Billed Budget SA Renewal. Refer to the SARN-NB algorithm type for more information about the base package algorithm.
  - Break Non-Billed Budget SA. Refer to the NBBR-BRK algorithm type for more information about the base package algorithm.
  - SA Activation - automatically activate SA (if you want to automatically activated Non-Billed Budgets when they are created). Refer to the SACR-AT algorithm type for more information about the base package algorithm. Note that this same algorithm may be used on many SA types.
  - SA Activation - create customer contact (if you want the system to create a customer contact when NBB SAs are activated). Refer to the SAAT-CC algorithm type for more information about the base package algorithm.
  - SA Stop - automatically stop SA (if you want to automatically transition Non-Billed Budgets from pending stop to stop when their stop is initiated). Refer to the SAIS-ST algorithm type for more information about the base package algorithm. Note that this same algorithm may be used on many SA types.
  - SA Stop - Stop Non-Billed Budget. Refer to the SAST-NB algorithm type for more information about the base package algorithm.
  - Bill Completion - Non-Billed Budget Credit Transfer. Refer to the BCMP-NB algorithm type for more information about the base package algorithm.
- On bill segment type, the Bill FT Algorithm (for all SAs that are eligible for NBB). Refer to the BSBF-BA algorithm type for more information about the base package algorithm. Note this is the standard bill FT algorithm type used for common bill transactions. It supports bill FT for non-billed budgets and other SA types. You only need to create it if it doesn't already exist in your system.
- On severance event type, an algorithm for Non-billed Budget Severance. Refer to the SVEV-NB algorithm type for more information about the base package algorithm.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Payment Schedule Algorithm Types.** For Non-Billed Budget payment schedule algorithm types, you need to define the algorithm types (if you add your own algorithm types). You do not need to define algorithms because the parameter values for the algorithm are defaulted on the recommendation rule and stored with the Non-Billed Budget SA. (Normally the algorithm holds the parameter values.) Refer to the NBPS-MON algorithm type for more information about the base package payment schedule algorithm.

NBB Debt Class And Collection Process

Set up a separate debt class, collection class control, collection process template and severance process template for Non-Billed Budgets according to the information in Credit and Collections and Non-Billed Budgets.

SA Types for SAs Covered by NBBs

You must modify the SA type for any SAs that you want to allow to be covered by a Non-Billed Budget.

- Verify that the specified **Bill Segment Type** (on the Billing page) references a financial transaction algorithm to set the current amount to zero for monitored Non-Billed Budgets. The FT creation algorithm type BSBF-BA (i.e., Payoff Amt = Bill Amt / Current Amt = Amt Due) supplied with the base package sets the current amount to zero for SAs that are covered by monitored non-billed budgets, though this is not evident by the name. (For SAs that are not covered by a Non-Billed Budget, the current amount is equal to the amount due or the recurring charge.) Refer to Billing For SAs Covered By The Non-Billed Budget for more information.
- Set the **Eligible for Non Billed Budget** flag (on the Billing page) to Eligible for Non-Billed Budget .
- Populate the **Adjustment Type (Current = 0)** (on the Detail page) to indicate the adjustment to be used to zero out the current amount on the covered SAs when the Non-Billed Budget SA is activated. Use the Add SA To Non-Billed Budget adjustment created above. This adjustment type is only called for SAs that are covered by a monitored Non-Billed Budget.
- Reference an **Adjustment Type Profile** (on the Adjustment Profile page) that includes the Add SA To Non-Billed Budget adjustment type referenced in the **Adjustment Type (Current = 0)** field above.
- If not already specified (for write off), an **Adj. Type (Synch Current)** (on the Main page) is also required. It is used to synchronize (make equal) the current amount with the payoff amount when the SA is removed from (i.e. no longer covered by) a Non-Billed Budget.

NBB Recommendation Rules

Copyright © 2026, Oracle and/or its affiliates.

Set up any NBB Recommendation Rules that you want to be available for your Non-Billed Budget SAs. Use any Non-Billed Budget Daily Amount Calculation algorithms defined above. Also use the Non-billed Budget Monthly Payment Schedule (NBPS-MON) algorithm types and specify the default parameters.

Non-Billed Budget SA Types

You must set up a SA types for your Non-Billed Budget service agreements. You may need multiple Non-Billed Budget SA types if you have different business rules for different types of Non-Billed Budgets, for example, if you have both monitored and unmonitored Non-Billed Budgets or if you support Non-Billed Budgets with different renewal requirements. The following points provide guidelines for creating a non-billed budget SA type.

- SA Type - Main (NBB)
- SA Type - Detail (NBB)
- SA Type - Billing (NBB)
- SA Type - Rate (NBB)
- SA Type - SP Type (NBB)
- SA Type - Adjustment Profile (NBB)
- SA Type - Credit and Collections (NBB)
- SA Type - Algorithms (NBB)
- SA Type - NBB Recommendation Rule (NBB)

SA Type - Main (NBB)

Service Type should reference something like "Miscellaneous Service".

Distribution Code should be the one you set up to book credits for Non-Billed Budgets.

Revenue Class should be set to N/A. (Revenue classes are not applicable because Non-Billed Budgets do not apply a rate and revenue classes are only relevant for SA types that use a rate.)

The **Payment Segment Type** should reference the Normal Payment.

Do Not Overpay should be on. Any excess payments should go to the overpayment SA, not the Non-Billed Budget SA.

Late Payment Charge is not applicable and should not be turned on because Non-Billed Budgets are not billed.

Adj. Type (Synch Current) should reference the Synch Balance for Non-Billed Budget adjustment type created above.

SA Type - Detail (NBB)

- **Special Role** is Non-Billed Budget.

Copyright © 2026, Oracle and/or its affiliates.

- **Adjustment Type (Xfer)** must be populated to indicate the adjustment to be used for transferring accumulated credit from the Non-Billed Budget SA to the covered SAs. This field is not used for unmonitored Non-Billed Budgets.
- **Renewal** may be optional, not allowed, or required depending on your business processes.
- If Renewal is required, specify the **Days Before Expiration for Renewal**.
- **Non-Billed Budget Monitoring** must indicate whether the Non-Billed Budget is monitored by the account debt monitor.

SA Type - Billing (NBB)

Non-Billed Budget SAs do not get billed, so the **Eligible for Billing** flag should be off.

Additionally, the **Characteristic Premise Required** should not be checked for Non-Billed Budgets.

SA Type - Rate (NBB)

Non-Billed Budget SAs do not use rates, so the **Rate Required** flag should be off.

SA Type - SP Type (NBB)

Non-Billed Budget SAs do not have service points, so the **Service Points Required** flag should be off.

SA Type - Adjustment Profile (NBB)

Adjustment Type Profile should reference the adjustment type profile for Non-Billed Budgets (set up above).

SA Type - Credit and Collections (NBB)

The credit and collections information should reference a **Severance Process Template** that calls the Non-Billed Budget Severance algorithm created above and expires the Non-Billed Budget SA. The **Debt Class** should reference an appropriate value consistent with your credit and collections rules. Typically, monitored Non-Billed Budgets should be in their own debt class. Unmonitored Non-Billed Budgets should have a **Debt Class** that is not Eligible for Collection.

The **Write Off Debt Class** is not applicable because the Non-Billed Budget contains no debt to be written off. Reference a debt class such as N/A. Refer to Defining Credit & Collections Options for more information.

Copyright © 2026, Oracle and/or its affiliates.

SA Type - Algorithms (NBB)

The SA type algorithms defined above must be set up:

- The Non-Billed Budget Credit Transfer algorithm created above should be specified for the Bill Completion system event (not used on unmonitored Non-Billed Budgets).
- The Break Non-Billed Budget SA algorithm created above should be specified for the Break NBB SA system event.
- The Non-Billed Budget Process Scheduled Payment algorithm created above should be specified for the Process NBB Scheduled Payment system event (not used on unmonitored non-billed budgets).
- The Non-Billed Budget SA Renewal algorithm created above should be specified for the SA Renewal system event.
- If you created the Automatic SA Activation algorithm above, it should be specified for the SA Creation system event.
- The Non-Billed Budget SA Activation algorithm created above should be specified for the SA Activation system event.
- If you created the Automatic SA Stop algorithm above, it should be specified for the SA Stop Initiation system event.
- The Stop Non-Billed Budget algorithm created above should be specified for the SA Stop system event.

SA Type - NBB Recommendation Rule (NBB)

Add the recommendation rules defined above that are valid for SAs of this type. Also, indicate which recommendation rule should be used as the default.

NBB Background Processes

Ensure that the following background processes are scheduled:

- Non-Billed Budget Scheduled Payment Processing (NBBPS)
- Non-Billed Budget Scheduled Payment Automatic Payment Create (NBBAPAY)
- Service Agreement Renewal (SARENEW)
- Stop Expired Service Agreements (SAEXPIRE)

Defining Quotation Options

This section describes tables that must be set up before quotations can be created.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, refer to The Big Picture of Quotations.

- Setting Up SA Types For Quotes
- Setting Up Quote Route Types
- Setting Up Terms and Conditions
- Setting Up Decline Reasons
- Setting Up Customer Classes For Quotes

Setting Up SA Types For Quotes

The topics in this section describe additional setup responsibilities required on SA types that can have proposal service agreements.

Note

Assumption. We have assumed that you've already designed your SA types for the services that you sell. If you haven't done this, refer to Defining Service Agreement Types .

- Enabling The Automatic Generation Of Billing Scenarios
- Enabling The Generation Of Simulated Bill Segments
- Enabling The Creation Of A Real SA When A Quote Detail Is Accepted

Enabling The Automatic Generation Of Billing Scenarios

As described under Proposal SAs Contain Billing Scenarios And Template Consumption, a proposal SA requires billing scenarios before a quote detail can be generated. The system can automatically create billing scenarios when a proposal SA is created. If this is required for your implementation, create a Proposal SA Creation algorithm type, and plug in the appropriate algorithm on each applicable Proposal SA type. Refer to PSAC-CBS for an example algorithm ***(note, this algorithm is not applicable for C2M but may be used as a reference only)***.

Enabling The Generation Of Simulated Bill Segments

As described under Creating Quotes And Quote Details, in order for the system to generate simulated bill segments for a proposal SA, you must plug-in a Proposal SA Bill Segment Generation algorithm on the proposal SA's SA type. Refer to CBSP-AR for an example algorithm that creates a simulated bill segments for each billing scenario linked to the proposal SA by calling rate application.

Copyright © 2026, Oracle and/or its affiliates.

Enabling The Creation Of A Real SA When A Quote Detail Is Accepted

As described under Accepting / Declining Quote Details, in order for the system to create a real SA when a proposal SA is accepted, you must plug-in a Proposal SA Acceptance algorithm on the proposal SA's SA type. Refer to PSAA-PS for an example algorithm that creates a real SA by copying the proposal SA.

> **Note**
>
> **Interval pricing service agreements.** are not supported. PSAA-PS has a parameter for creating interval profiles, time-of-use maps and contract options when the real SA is created. This specific parameter/option is not applicable for C2M.

Setting Up Quote Route Types

Quote route types control how quotes are routed to customers and prospects. To define a quote route type, open **Admin**, and then **Sales & Marketing**, and then **Quote Route Type.**

> **NOT_SUPPORTED**
>
> Refer to Printing Quotes for more information about how quotes are routed to customers and prospects.

Description of Page

Enter a unique **Quote Route Type** and **Description** for every quote route type.

Quote Routing Method controls the type of information that may be defined when the respective **Quote Route Type** is selected on Account - Person Information. The following options are available:

- Postal: Use this method if the routing is via the postal service.
- Fax: Use this method if the routing is via fax.
- Email: Use this method if the routing is via email.

> **Note**
>
> The values for **Quote Routing Method** are customizable using the Lookup table. This field name is QTE_RTG_METH_FLG .

- The next two fields control how quotes that are routed using this method are printed (both in batch and online).

Copyright © 2026, Oracle and/or its affiliates.

- Use **Batch Control** to define the process that creates the flat file that is passed to your quote printing software. Refer to Tec hnical Implementation of Printing Quotes In Batch for more information about these processes.
- Use **Extract Algorithm** to define the plug-in that constructs the "flat file records" that contain the information merged onto quotes routed using this method. Refer to Printing Quotes for more information.

Setting Up Terms and Conditions

As described under Legal Terms and Conditions May Be Specified On SAs , your SA type start options can reference terms and conditions (T&C's) that should be defaulted onto new service agreements (both real and proposal). Each T&C is identified with a terms and condition code. To define a terms and conditions code, open **Admin**, and then **Sales & Marketing**, and then **Terms and Conditions**, and then **Add.**

> **Note**
>
> **T&C print order.** The value of the T&C code controls the order in which the T&C appears on the printed quote. This means you should assign these codes in some type of structured format (e.g., 01...) if you would like them to appear in a certain order.

Description of Page

Enter a unique **Terms and Condition** code and **Description**. Use **Text** to describe the exact terms.

Setting Up Decline Reasons

A proposal SA decline reason must be supplied when a proposal SA is declined. Open **Admin**, and then **Sales & Marketing**, and then **Proposal SA Decline Reason** to define your reason codes.

Description of Page

Enter an easily recognizable **Decline Reason** and **Description** for each proposal SA decline reason.

Where Used

Decline reasons are used when a quote detail is declined.

Setting Up Customer Classes For Quotes

An optional plug-in spot exists on customer class where you can introduce additional logic to be executed when a quote is completed for an account that belongs to this customer class. The base package comes supplied with a sample algorithm that creates a workflow process when a quote is completed (refer to the algorithm QTEC-WP for more information).

Copyright © 2026, Oracle and/or its affiliates.

Defining Case Management Options

Case management functionality is a highly configurable tool your organization can use to manage many situations, including (but certainly not limited to) the following:

- a high-bill complaint,
- a bankruptcy,
- an inspection of a premise,
- a customer's request for literature,
- an application for new service,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change service providers on a future date,
- the processing of a market message in a deregulated environment,
- ... (the list is only limited by your time and imagination)

Obviously the steps involved in the resolution of the above cases are very different. The topics in this section describe how to configure the system to manage your cases as per your organization's desires.

> **Note**
>
> **Separate module.** The Case Management functionality is associated with a separate module. If the Case Management module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

> **NOT_SUPPORTED**
>
> Refer to Case Management for a description of how end-users use cases.

- The Big Picture Of Cases
- Setting Up Case Management Options

The Big Picture Of Cases

The topics in this section provide background information about how to configure the system to support your case management requirements.

- Case Type Controls Everything
- Scripts and Cases

Copyright © 2026, Oracle and/or its affiliates.

- To Do's and Cases
- Examples of Case Types

Case Type Controls Everything

Whenever a user creates a case, they must specify the type of case (e.g., high-bill complaint, literature request, etc.). The case type controls how the case is handled.

Case types hold the business rules that control cases. Since these business rules can sometimes be quite complicated, setting up case types requires planning and foresight. The topics in this section describe the type of business rules that can be configured on your case types.

- Person / Account / Premise Applicability
- Contact Information Applicability
- Business Object Association
- Additional Information
- Access Rights
- Lifecycle
- Status-Specific Business Rules
- Responsible User Applicability

Person / Account / Premise Applicability

Some types of cases may be person-oriented, others may be premise-oriented, and still others may be account-oriented. For example:

- Cases used to keep track of a literature request would reference the person who requested the literature.
- Cases used to keep track of the inspection of a property would reference the premise being inspected.
- Cases used to keep track of a high-bill complaint would reference the account associated with this bill(s) being disputed.

When you set up a case type, you define if its cases must reference a person, account, and/or premise. Note, any combination of these objects is permitted on a case.

Contact Information Applicability

When a case is created, you may want to keep track of how to contact its originator. For example, you may want to record the originator's email address or phone number. When you set up a case type, you define if contact information is required, optional or not allowed on its cases.

Copyright © 2026, Oracle and/or its affiliates.

Business Object Association

A case type may reference a Business Object, which serves as a link between cases of that type and the options that are associated with the business object.

Additional Information

Some of your cases may require additional information (in the form of characteristics). For example, a high-bill complaint may require at least one bill. When you set up a case type, you can define the additional fields that are required. In addition, you can define default values for these fields.

The case functionality also allows you to require characteristics when a case enters a given state. Refer to Required Fields Before A Case Enters A State for the details.

> **Note**
>
> **Requiring supporting documents.** Because any type of characteristic can be referenced on a case, you can require references to supporting documents by requiring a file location characteristic.

Access Rights

You can take advantage of the system's security to restrict cases of a given type to certain users. For example, you can restrict high-bill complaints to specific user groups.

The following points describe how to implement this type of security:

- Create an application service for each type of case you need to secure
- Define the access modes Add, Inquire and Change for each application service
- Define the applicable application service on each case type
- Link the appropriate user groups to each application service
  - For user groups that are allowed to add cases of a given type, define Add as a valid access mode.
  - For user groups that are allowed to view cases of a given type, define Inquire as a valid access mode
  - For user groups that are allowed to change cases of a given type, define Change as a valid access mode

If you restrict access to a case type's cases, you can further restrict which users can work on cases given the status of the case. Refer to Which Users Can Transition A Case for more information.

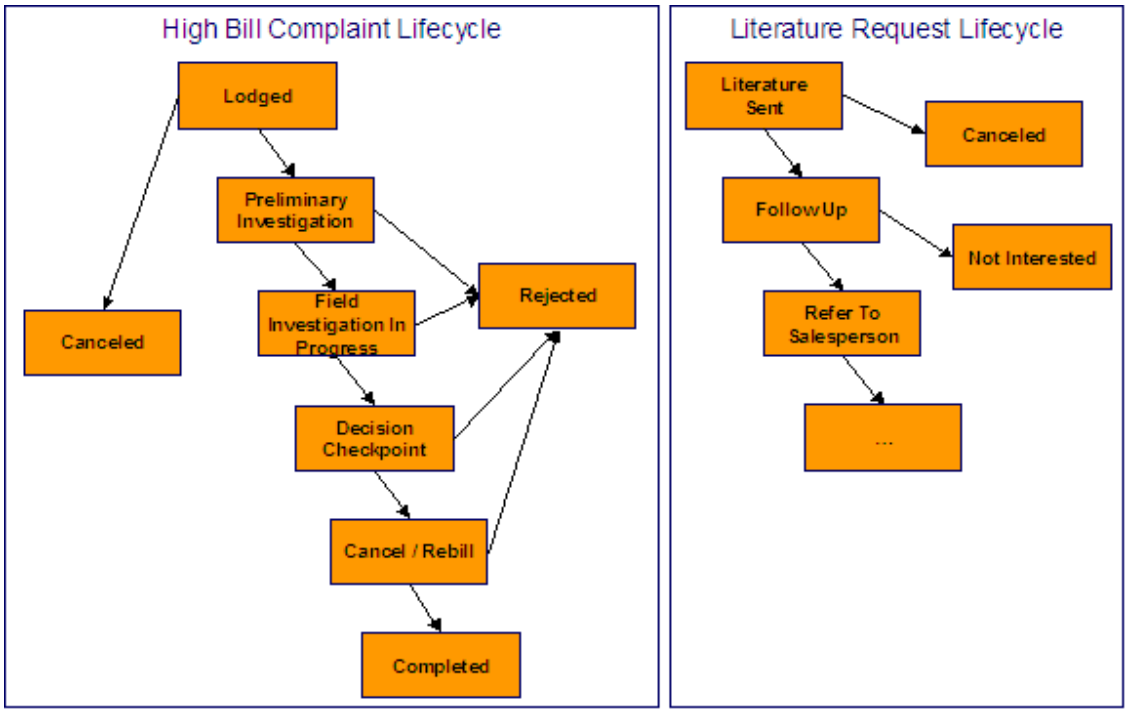
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Restricting access to cases is optional.** If you don't specify an application service on a case type, all users (who have access to the case transaction) may access its cases.

Lifecycle

Many objects in the system have predefined lifecycles whose rules are governed by the base-package and cannot be changed. For example, a service agreement starts out in the Pending Start state and eventually becomes Closed when it's been final billed (and paid). You can't change the system to allow a service agreement to starts its life in the Closed state.

The lifecycle of cases is not governed by the base package. Rather, you define the lifecycle of your cases when you set up their case types. Examine the following diagrams; the one on the left shows the potential lifecycle of a case that manages a high-bill complaint, the one on the right shows the potential lifecycle of a case that manages a customer's literature request.



Potential Lifecycles Of Two Types Of Cases

> **Note**
>
> **Just examples.** The above lifecycles are just examples. When you set up your case types, you must define the valid states for your case type.

The topics that follow describe important concepts that are illustrated in the above diagrams.

- Valid States versus State Transition Rules

Copyright © 2026, Oracle and/or its affiliates.

- Transitory States
- One Initial State and Multiple Final States
- Allowing A Case To Be Reopened
- Make Sure To Have A Canceled State
- Buttons Are Used To Transition A Case From Status To Status
- State Transitions Are Audited
- Reports and Analytics Highlight Productivity

Valid States versus State Transition Rules

The orange boxes in the above diagram show the potential valid states a given case can have. The lines between the boxes indicate the state transition rules. These rules govern the states a case can move to while it's in a given state. For example, the above diagram indicates a high bill complaint that's in the Lodged state can be either Canceled or moved into the Preliminary Investigation state.

When you set up a case type, you define both its valid states and the state transition rules.

Transitory States

You can define a state in a case type as **Transitory** if you do not wish the case to exist in a particular state. For example, let's assume that an algorithm is associated with the Decision Checkpoint state (Enter Processing) that would automatically determine the next state for the case (i.e. Cancel/Rebill or Reject) and also contains logic to transition the case accordingly. In this scenario, you may not ever want the case to exist in the Decision Checkpoint state, so that a user won't ever see a high bill complaint in that state. If the other states were marked as non-transitory, and an error were to occur during the transition from Decision Checkpoint to Cancel/Rebill, the case would roll back any changes to data made in the Cancel/Rebill (Enter Processing) state along with the changes made in the Decision Checkpoint state, and would end up in the Field Investigation In Progress state - the last non-transitory state prior to Decision Checkpoint.

One Initial State and Multiple Final States

When you set up a case type's states, you must pick one as the initial state. The initial state is the state assigned to new cases of a given type. For example, high-bill complaint cases have an initial state of Lodged, whereas literature request cases have an initial state of Literature Sent.

You must also define which statuses are considered to be "final". When a case enters a "final" state, it is complete and no further action is necessary. You might want to think of the "final" states as the potential outcomes of a case. For example, a high-bill complaint has potential outcomes of Completed, Rejected, and Canceled.

The "final" states are used by the system to differentiate between open and closed cases. For example, an alert highlights when the person / account / premise in context has open cases (this alert only exists if you've plugged-in the appropriate installation alert).

Copyright © 2026, Oracle and/or its affiliates.

Allowing A Case To Be Reopened

You can set up your state transition rules to allow a case to be reopened (i.e., to be moved from a final state to a non-final state). Neither of the above examples allows this, but it is possible if you configure a case type accordingly.

Make Sure To Have A Canceled State

The system does not allow you to delete a case. Therefore, if you want to support logical deletion, you should have a status of Canceled early in a case type's lifecycle. Doing this allows a user to cancel (i.e., logically delete) a case.

> **Note**
>
> **Cancel reason.** You might want to consider setting up your case types to require a cancel reason (in the form of a predefined value characteristic) when a user cancels a case. Refer to Required Fields Before A Case Enters A State for more information.

Buttons Are Used To Transition A Case From Status To Status

When a case is displayed on Case - Main, a separate button is shown for each state into which the case can be transitioned. For example, a high-bill complaint case that is in the Lodged state would show two buttons: **Start Investigation** and **Cancel**. If the user presses the **Start Investigation** button, the case is transitioned to the Preliminary Investigation state. If the user presses the **Cancel** button, the case is moved to the Canceled state.

You may define the text displayed on the button differently for each state transition. This allows the action description to be varied according to the previous status. For example, the button to transition from New to Active may be labeled **Activate**, but the button to change from Closed to Active may be labeled **Reactivate**.

Refer to Which Users Can Transition A Case for instructions describing how to restrict users to specific actions.

State Transitions Are Audited

The system maintains an audit trail whenever a case transitions from one state to another. This audit is shown in the case's log.

Reports and Analytics Highlight Productivity

When you set up a case type's lifecycle, keep in mind that several reports and analytics highlight how long it took cases to transition into a state. For example, you can use a report to see how long it took high-bill complaints to be completed (or initially actioned or ...). Refer to the Reports chapter for the details of case reports.

Copyright © 2026, Oracle and/or its affiliates.

Status-Specific Business Rules

As described in Lifecycle, when you set up a case type, you define the possible states its cases can pass through. The following topics describe business rules that can be configured for each state.

- A Script That Helps A User Work Through A Case
- Required Fields Before A Case Enters A State
- Validation Before A Case Enters A State
- Additional Processing When Entering A State
- Validation Before A Case Exits A State
- Additional Processing When Exiting A State
- Automatic Transition Rules
- Script Launching Option
- Which Users Can Transition A Case Into A State

A Script That Helps A User Work Through A Case

You can define a Business Process Assistant script that helps a user work a case while it's in a given state. For example, when you set up the Preliminary Investigation state for the high-bill complaint case type, you can define a script. A user can then easily launch this script to help them work through a case in this state.

Please keep the following in mind when you're designing how to integrate BPA scripts with your cases:

- You can have a different script for each state. For example, you could develop a script to help a user work on a case while it's in the Preliminary Investigation state and a different script to help them work in a case while it's in the Decision Checkpoint state.
- Rather than make a user launch a script by pressing a hyperlink on the case page, you can have the system automatically launch the script while the case is in a given state. Refer to Script Launching Option for more information.
- You can also have the system automatically launch a script when a user selects a To Do entry. Refer to Launching Scripts When To Do Entries Are Selected for more information.

NOT_SUPPORTED

Refer to Scripts and Cases for more information about how to streamline your case processing with scripts.

Required Fields Before A Case Enters A State

Copyright © 2026, Oracle and/or its affiliates.

You can define additional fields (i.e., characteristics) that are required before a case can enter a given state. For example,

- You can indicate a high-bill complaint must reference at least one bill before it enters the Preliminary Investigation state.
- You can indicate a case must reference a cancel reason before it enters the Canceled state.

You do this by indicating that characteristics (that were optional when the case was added) are required when a case enters a given state.

Validation Before A Case Enters A State

You can define validation that executes before a case can enter a given state. For example, you can indicate the case must have been assigned a responsible user before it can enter the Preliminary Investigation state. This validation logic is held in algorithms that are plugged in on the case type and therefore you can define any type of validation.

Additional Processing When Entering A State

You can define additional processing that should happen when a case enters a given state. For example, you can have a letter created when a high-bill complaint case is Rejected. Similarly, you can have a To Do entry created when a high bill complaint enters the Preliminary Investigation state. This additional processing is held in algorithms that are plugged in on the case type and therefore you can define any type of additional processing.

You can also incorporate state transitioning logic within routines that are executed when a case enters a state, so that you do not need to rely upon CASETRAN to transition your cases. For example, when the state entry routines of the Preliminary Investigation status for a high-bill complaint are executed, they may be designed to transition the case into either the Rejected or Field Investigation In Progress state without waiting. Note that your Exit Validation and Exit Processing logic, if configured for the case state, will still be executed as part of the state transition. Auto-Transition logic for this state will be ignored during this transition.

Validation Before A Case Exits A State

You can define validation that executes before a case can exit a given state. For example, you might want to check the account's balance is less than a given value before a case can exit a given state. This validation logic is held in algorithms that are plugged in on the case type and therefore you can define any type of validation.

Additional Processing When Exiting A State

You can define additional processing that should happen when a case exits a given state. For example, you can have a To Do entry automatically completed when a high bill complaint leaves the Decision Checkpoint state. This additional processing is held in algorithms that are plugged in on the case type and therefore you can define any type of additional processing.

Copyright © 2026, Oracle and/or its affiliates.

Automatic Transition Rules

You can define rules that automatically transition a case into a different state. For example, you can indicate a literature request should be transitioned to the Follow Up state 1 week after the literature is sent. Similarly, you can indicate a high-bill complaint should transition to the Decision Checkpoint state after the fieldwork is complete. These rules are held in algorithms that are plugged in on the case type and therefore you can define any type of automatic transition rules.

Cases in a state with automatic transition rules are monitored by the CASETRAN background process. Each time this program runs, the respective automatic transition plug-in is called for each such case and it transitions the case if the condition applies.

> **Note**
>
> **When to execute CASETRAN.** Because your automatic transition rules will be dependent on your business requirements, you need to think carefully about when you run the CASETRAN background process. For example, if you have automatic transition rules that transition a case to a new state when a related field activity is completed, you would want to schedule this job to run after field activities are uploaded. If you have rules to transition a case after a customer pays a deposit, you'd want to schedule this job to run after payments are uploaded. Bottom line - your business rules will dictate the frequency of execution.

When the user adds a new case or changes the state of a case manually the system attempts to auto-transition the case to subsequent statuses as necessary. If auto-transition rules apply to the new state (and to subsequent ones) they would be executed right away. In other words, you don't need to wait for the auto-transition background process to be executed. An indication that the case was auto-transitioned online is displayed right below the action buttons section.

> **Note**
>
> **Auto-Transition Errors.** Online auto-transition is performed recursively committing each successful state transition to the database. It is performed up to 100 times or until an error is encountered during the process. If this happens, auto-transition stops at the last **non-transitory** state into which a successful transition had occurred. Two case log entries will be generated automatically - one containing the message that a transition error has occurred, and a second containing the actual error message. A To Do entry will also be generated automatically upon rollback. The type of this To Do entry will be taken from 1) the Case Transition Exception To Do Type **option** for the **Business Object** associated with the case type, and if this is not populated, 2) the Exception To Do Type indicated on the Case Options Feature Configuration. All of the above error handling is true for both batch and online processing of cases.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Triggering Auto-Transition.** If you have a customized process that affects the state of a case and you want the case to be auto-transitioned right away, i.e. not wait for the next scheduled CASETRAN background process to execute, you can customize that process to trigger auto-transition for the specific case, or you can put the state transition logic into the routines that execute at state entry time.

Script Launching Option

You can define whether the script associated with a given state is to be automatically launched while the case is in that state. The system supports the following options:

- Launch the script only if no script is currently active.
- Always launch the script unless this specific script is currently active.

> **Warning**
>
> With this option, if a script is currently open in the page's BPA script area then it will be automatically closed and the case script will open.

- Do not automatically launch the script.

You do this by plugging-in a Script Launching algorithm for the given state. If no such plug-in is provided the script is not automatically launched.

Which Users Can Transition A Case Into A State

If you have restricted access to a case type, you can further restrict which user groups are allowed to transition a case into specific states. For example, you can control which user group can transition a high bill complaint into the Preliminary Investigation state. The following points describe how this is done:

- Define actions on the application service defined on the case type. You must define an action for each status that you need to secure.
- Define each status's corresponding action. Note, you only need to link a status to an action if it's secured. Any user with access to the case type can perform statuses that aren't linked to actions.
- Define the transition role for each status's valid next status. You can assign valid next statuses to be reachable via system (only), or system and user.
- Define which user groups have access to the actions (i.e., statuses). In addition, these user groups should have access to the Change action.

Responsible User Applicability

Copyright © 2026, Oracle and/or its affiliates.

Some of your cases may require a "responsible user". This is the user who has overall responsibility for the case. When you set up a case type, you define if a responsible user is required, optional or not allowed on its cases.

The following points describe how to set up the system if a responsible user is not required when a case is first created, but is later in its lifecycle:

- Indicate that a responsible user is optional on the case type
- Plug-in either an exit validation or entry validation algorithm on one of the case type's states to require a responsible user at some point in a case's lifecycle

> **Note**
>
> **Address To Do entries to the responsible user.** If you use the base-package algorithm to create a To Do entry when a case enters a given state, you can indicate that the To Do entry should be addressed to the responsible user on the case.

Scripts and Cases

There are three ways Business Process Assistant scripts can be used to manage cases:

- You can create a BPA script to help users create a case. For example, a script can help a user create a new high-bill complaint.
- Using a script to create a case can save a user a lot of time (and training efforts). This is because the script can automatically populate many fields on the case based on answers to questions.

Refer to Initiating Scripts for a description of how end-users initiate scripts.

- You can create a script to help users work on a case when it's in a given state. Refer to A Script That Helps A User Work A Case for more information.
- You can set up your case types to create To Do entries to notify users when cases exist that require their attention. Users can complete many of these ToDo entries without assistance. However, you can set up the system to automatically launch a script when a user selects a ToDo entry. For example, consider a ToDo entry that highlights a high-bill complaint that requires investigation. You can set up the system to execute a specific script when a user selects this ToDo entry. This script might guide the user through the investigation process (and help them update the case). Refer to Executing A Script When A To Do Entry Is Selected for more information.

To Do's and Cases

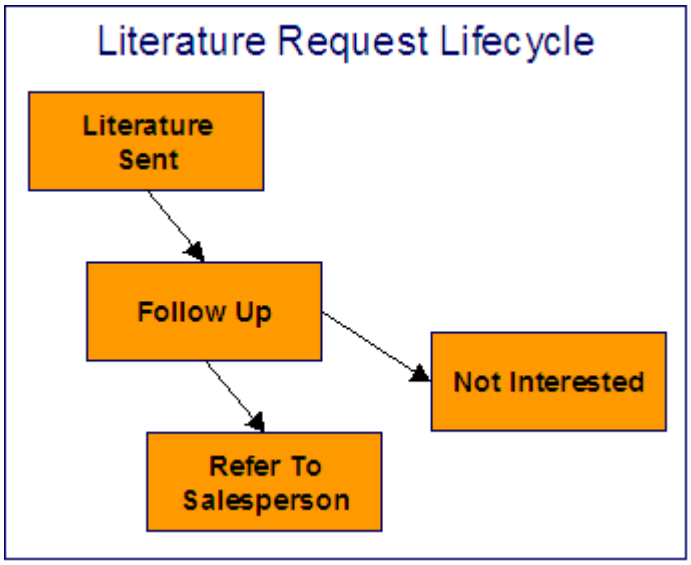
The topics in this section provide background information about how to facilitate case management with To Do entries.

- Creating and Completing To Do Entries
- Launching Scripts When To Do Entries Are Selected
- All To Do Entries Are Visible

Copyright © 2026, Oracle and/or its affiliates.

Creating and Completing To Do Entries

You can configure your case types to create and complete To Do entries when a case enters or exits a state. Let's use the following lifecycle diagram to illustrate a potential use of To Do's:



Let's assume the following:

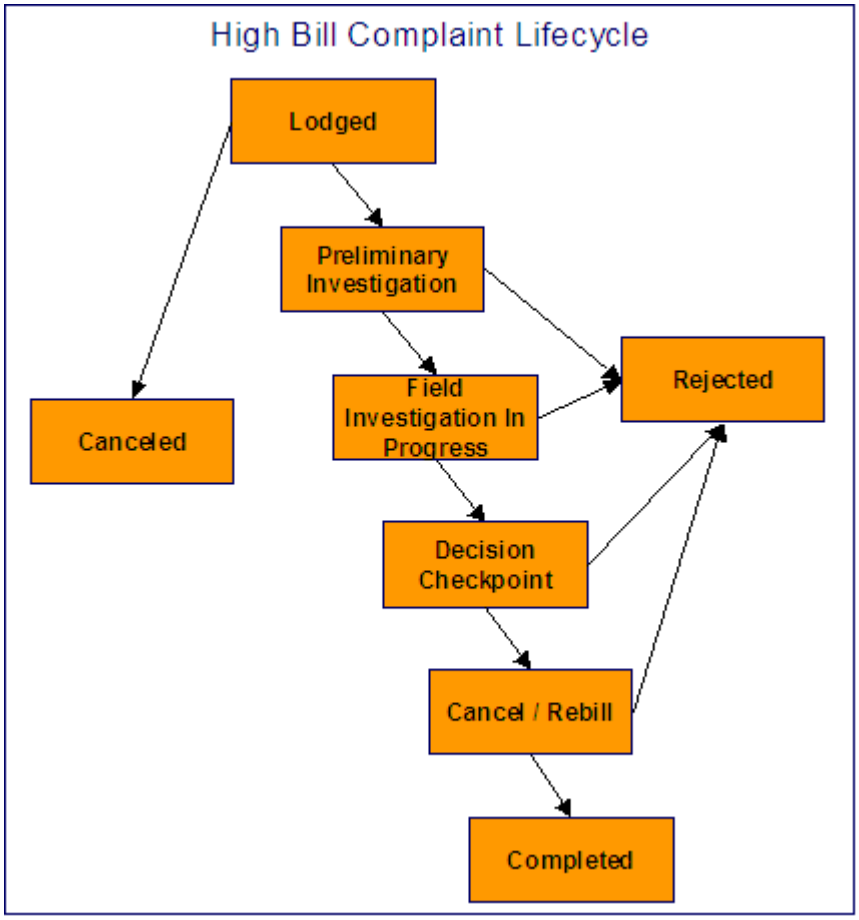
- You want a To Do entry created when a literature request case enters the Follow Up state. You want this To Do automatically completed when the case enters either the Refer To Salesperson or Not Interested states. Note, we refer to this as the "first" To Do entry below.
- You want a different To Do entry created when a case enters the Refer To Salesperson state. You do not want the system to automatically complete this entry (the sales person must manually do this). Note, we refer to this as the "second" To Do entry below.

To implement the above, you would set up a case type as follows:

- Plug-in an entry processing algorithm on the Follow Up status to create the first To Do entry.
- Plug-in an exit processing algorithm on the Follow Up status to complete the first To Do entry.
- Plug-in an entry processing algorithm on the Refer To Salesperson status to create the second To Do entry.

While the case type illustrated above had a single To Do entry "active" at any point in time, you can easily configure a case type to have multiple To Do entries active at any point in time. Let's use the following lifecycle diagram to illustrate this point:

Copyright © 2026, Oracle and/or its affiliates.



Let's assume the following:

- You want a To Do entry created when a high bill complaint is created and you want it completed when the case reaches the Canceled, Rejected or Approved states. This To Do entry could be used by a supervisor to monitor the number of high-bill complaints logged. Note, we refer to this as the "first" To Do entry below.
- You want a different To Do entry created when the case enters the Preliminary Investigation state and you want this entry automatically completed when the case leaves this state. Note, we refer to this as the "second" To Do entry below.
- You want a different To Do entry created when the case enters the Decision Checkpoint state and you want this entry automatically completed when the case leaves this state. Note, we refer to this as the "third" To Do entry below.

To implement the above, you would set up the case type as follows:

- Plug-in an entry processing algorithm on the Lodged status to create the first To Do entry. Plug-in an entry processing algorithm on the Canceled, Rejected and Completed statuses to complete this entry.
- Plug-in an entry processing algorithm on the Preliminary Investigation status to create the second To Do entry. Plug-in an exit processing algorithm on the Preliminary Investigation status to complete this entry. We elected to use an exit processing algorithm because we only have to plug it in on one status. If we'd used an entry processing algorithm, we would need to plug it in on the 2 statuses into which a Preliminary Investigation status can transition.

Copyright © 2026, Oracle and/or its affiliates.

- Plug-in an entry processing algorithm on the Decision Checkpoint status to create the third To Do entry. Plug-in an exit processing algorithm on the Decision Checkpoint status to complete this entry.

Launching Scripts When To Do Entries Are Selected

You can set up your case types to create To Do entries to notify users when cases exist that require their attention. Users can complete many of these To Do entries without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a high-bill complaint that requires investigation. You can set up the system to execute a specific script when a user selects this type of To Do entry. This script might guide the user through the investigation process. Refer to Executing A Script When A To Do Entry Is Selected for more information.

All To Do Entries Are Visible

When a case is displayed on Case Maintenance, the system summarizes the number of To Do entries associated with the case (if you've set up your To Do types appropriately).

Examples of Case Types

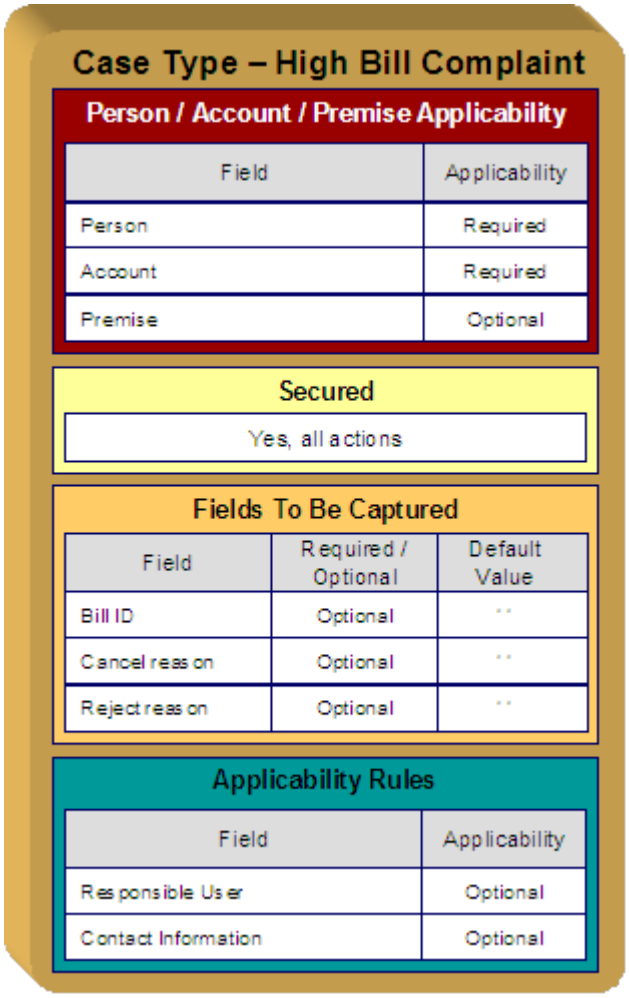
The topics that follow provide examples of case types related to several business processes. Use the information in this section to form an intuitive understanding of case types. After attaining this understanding, you'll be ready to design your own case types.

- High Bill Complaint

High Bill Complaint

Some organizations will set up a case to manage a high-bill complaint. The following diagram illustrates how such a case type might look:

Copyright © 2026, Oracle and/or its affiliates.



Case Type – High Bill Complaint

Person / Account / Premise Applicability

Field	Applicability
Person	Required
Account	Required
Premise	Optional

Secured

Yes, all actions

Fields To Be Captured

Field	Required / Optional	Default Value
Bill ID	Optional	''
Cancel reason	Optional	''
Reject reason	Optional	''

Applicability Rules

Field	Applicability
Responsible User	Optional
Contact Information	Optional

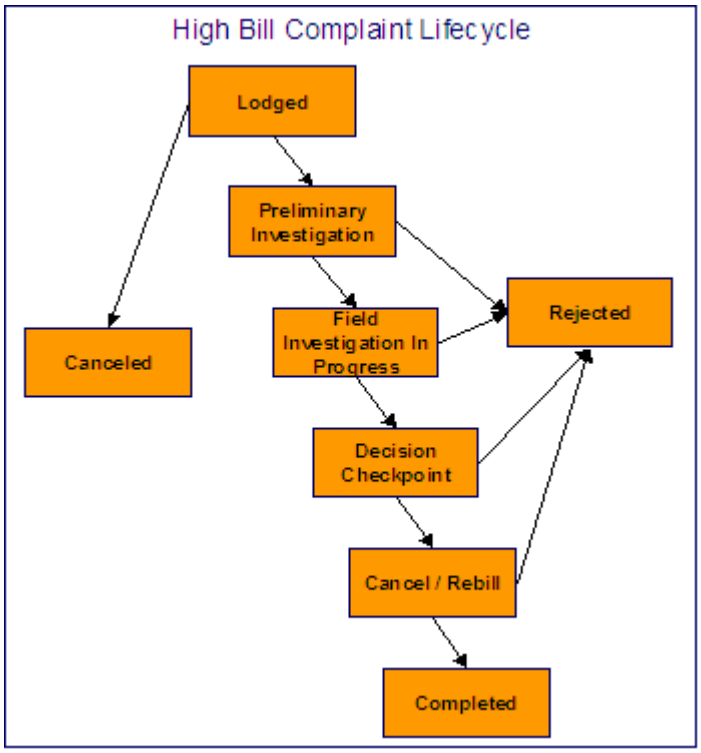
Note the following about the High Bill Complaint case type:

- Notice that we set up the case type to require a person and an account, but premise is optional. This is because a bill can span multiple premises and knowing the premise isn't so important on cases of this type.
- We need to restrict high-bill complaints to specific user groups. This means we need to set up a specific application service for this case type (that has a separate "action" for each status).
- Cases of this type have 3 additional fields over their lifetime. Notice the following:
  - The **Bill ID** characteristic is set up to be optional. This is because we've assumed that sometimes a high-bill complaint case can be lodged when you can't find the bill in question and you still want to log the case.
  - Later in this section, you'll see that we've configured the Preliminary Investigation status to require a **Bill ID** before a case can enter this state.
  - Both the **High Bill Complaint Cancel Reason** and **Reject Reason** are optional. Later in this section, you'll see that we've configured the Canceled and Rejected statuses to require these fields, respectively.
- Cases of this type do not need a **Responsible User** when first created. Later in this section, you'll see how we've configured the Preliminary Investigation status to require a **Responsible User** before a case can enter this state.

Copyright © 2026, Oracle and/or its affiliates.

- Cases of this type do not need **Contact Information**. This was a subjective decision and depends on your organization's philosophy.

The topics that follow describe each of the statuses in a high-bill complaint's lifecycle. We have assumed the following state transition rules:

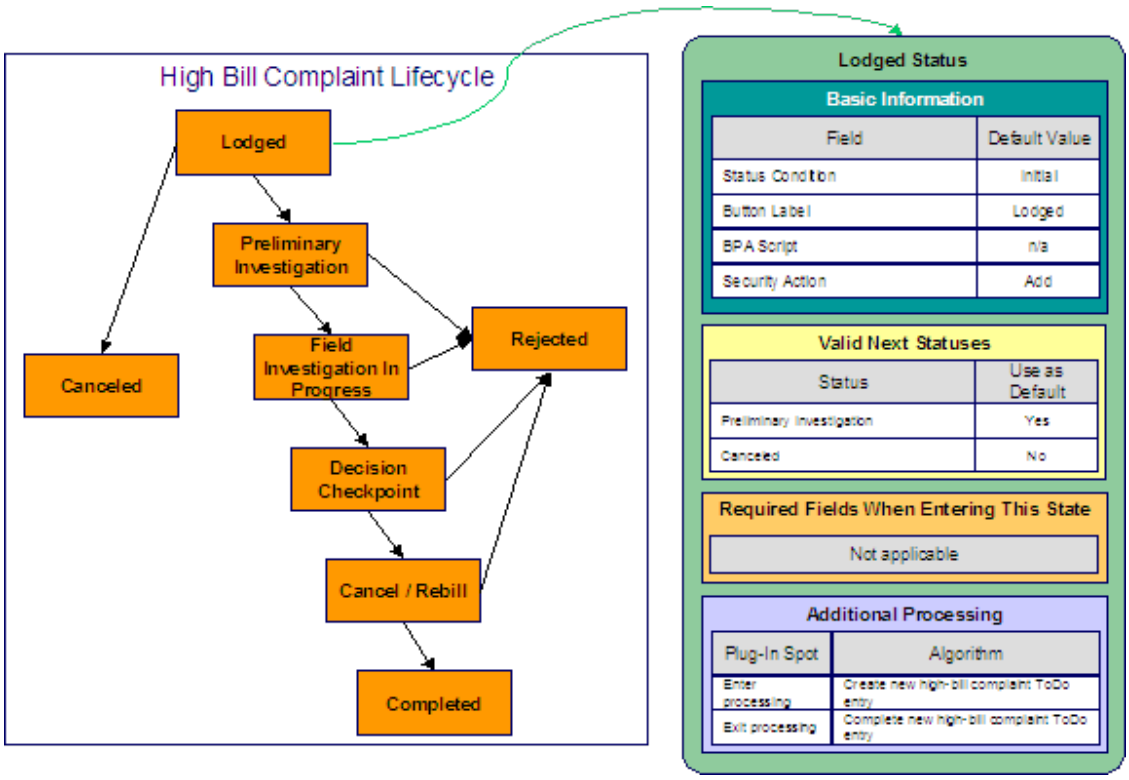


- Lodged High Bill Complaint
- Preliminary Investigation High Bill Complaint
- Field Investigation High Bill Complaint
- Decision Checkpoint High Bill Complaint
- Cancel Rebill High Bill Complaint
- Completed High Bill Complaint
- Rejected High Bill Complaint
- Canceled High Bill Complaint

Lodged High Bill Complaint

The following is an example of the configuration of the Lodged status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



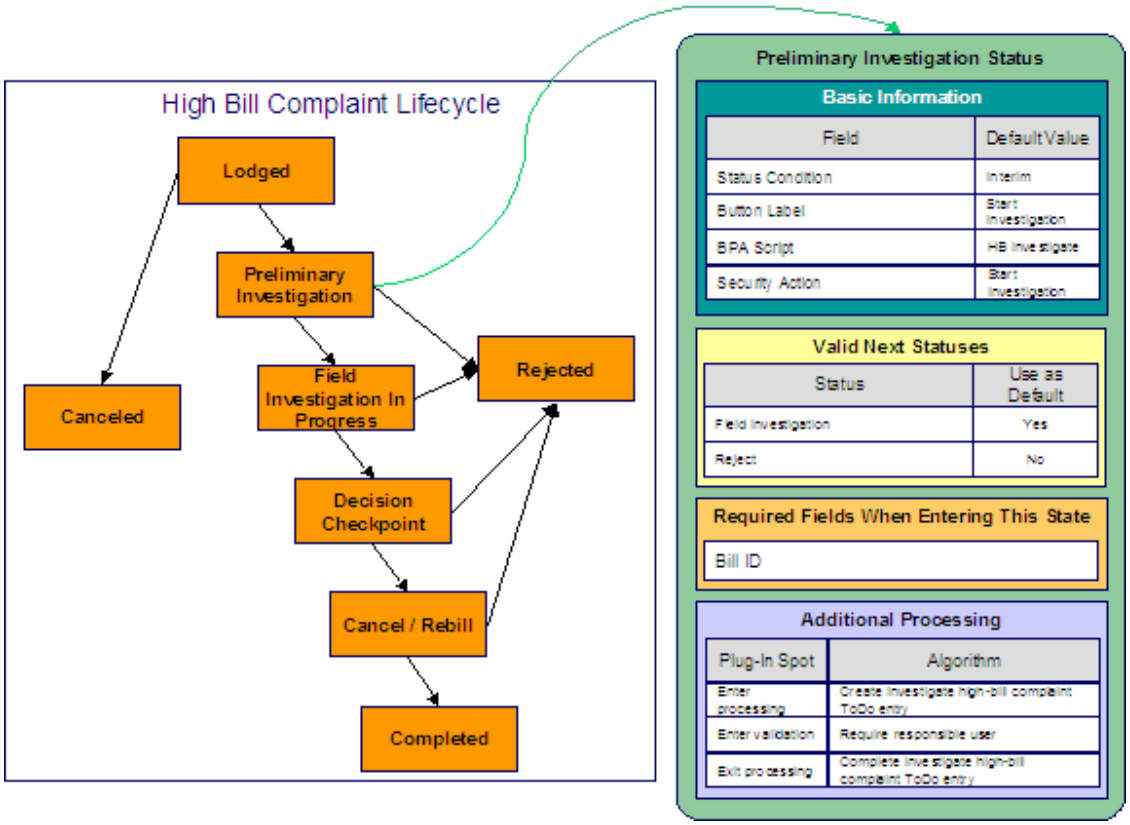
Note the following about this status:

- Lodged is the initial state. The initial state is the state populated on new cases of a given type. Remember that only one status can be defined as the initial state.
- It has a button label even though it's the initial state. The above diagram doesn't allow a user to ever transition a case into this state and therefore there will never be a button with such a label. However, it's a required field just in case you change your business rules.
- We have decided not to use a BPA script to help a user work on a high-bill complaint when it's in this state (this is probably not the best decision as BPA scripts can be quite useful).
- We have associated the Add action with this status. This means that only users with rights to the add action for the application service defined on the case type can add cases of this type.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Canceled and Preliminary Investigation states.
- Notice that the Additional Processing plug-ins create and complete a To Do entry when a case enters and exits this state, respectively.

Preliminary Investigation High Bill Complaint

The following is an example of the configuration of the Preliminary Investigation status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



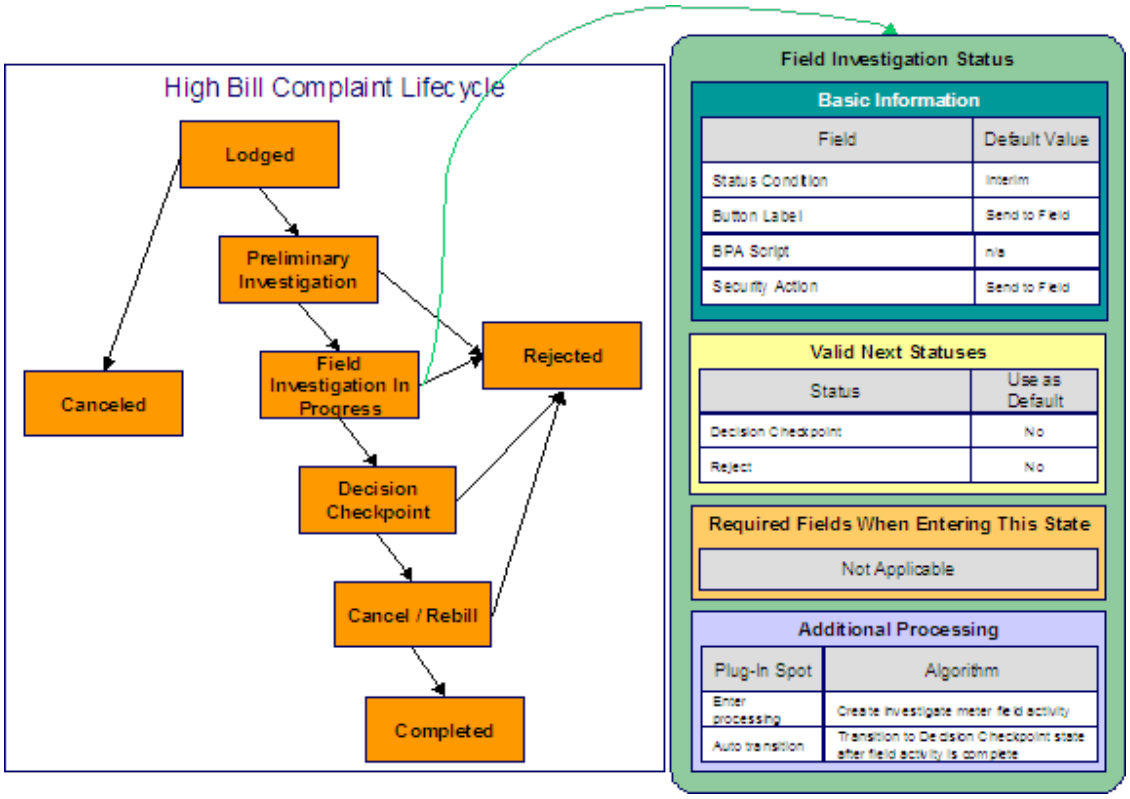
Note the following about this status:

- Preliminary Investigation is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Start Investigation**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Lodged state as this is the only state that can transition into the Preliminary Investigation state.
- We have decided not to specify a BPA script on this status. Rather, we're going to set up the To Do type used to highlight cases in this state to automatically launch an appropriate BPA script when a user selects the To Do entry.
- We have associated the Start Investigation action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Field Investigation and Rejected states.
- Notice that a **Bill ID** must be specified on the case before it can be moved into this state.
- Notice that the Additional Processing plug-ins do the following:
  - Create and complete a To Do entry when a case enters and exits this state, respectively
  - Require a responsible user before a case can enter this state

Field Investigation High Bill Complaint

Copyright © 2026, Oracle and/or its affiliates.

The following is an example of the configuration of the Field Investigation In Progress status for high bill complaint cases.



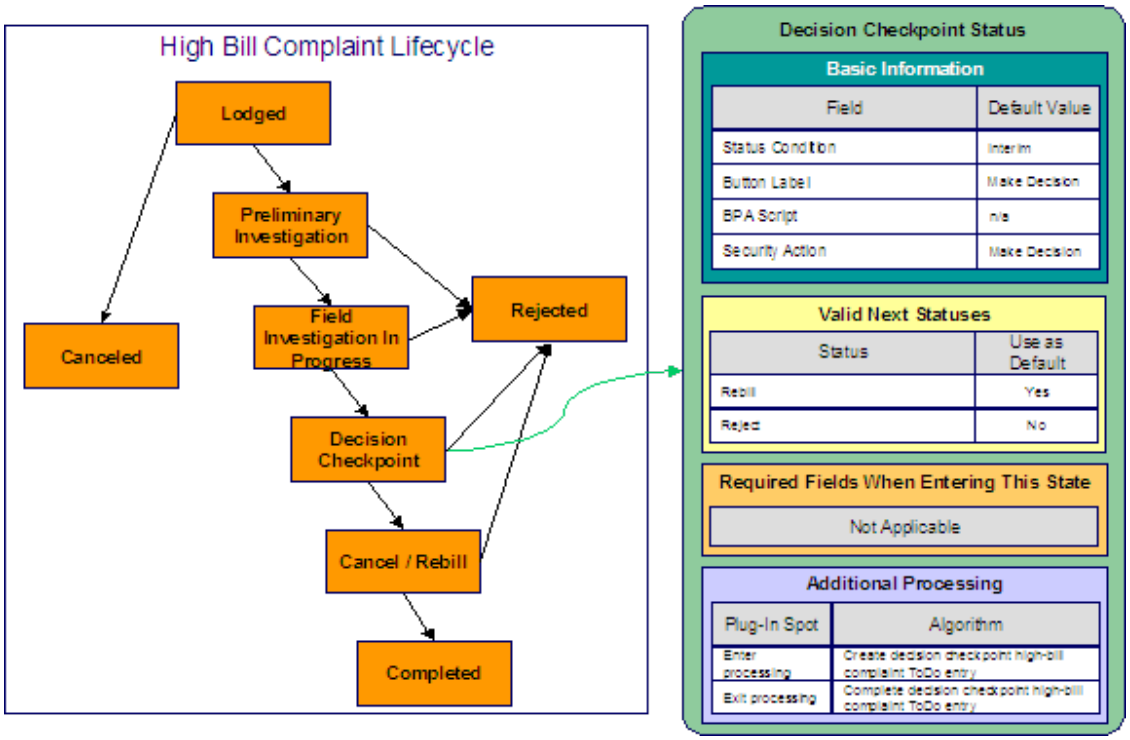
Note the following about this status:

- Field Investigation In Progress is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Send to Field**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Preliminary Investigation state as this is the only state that can transition into the Field Investigation in Progress state.
- We have decided not to specify a BPA script on this status because users don't work cases in this state (see the Additional Processing notes below for why this is the case).
- We have associated the Send to Field action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Decision Checkpoint and Rejected states.
- Notice that the Additional Processing plug-ins do the following:
  - Create a field activity when a case enters this state
  - Cause the case to automatically transition to the Decision Checkpoint state when the field activity is completed

Decision Checkpoint High Bill Complaint

Copyright © 2026, Oracle and/or its affiliates.

The following is an example of the configuration of the Decision Checkpoint status for high bill complaint cases.



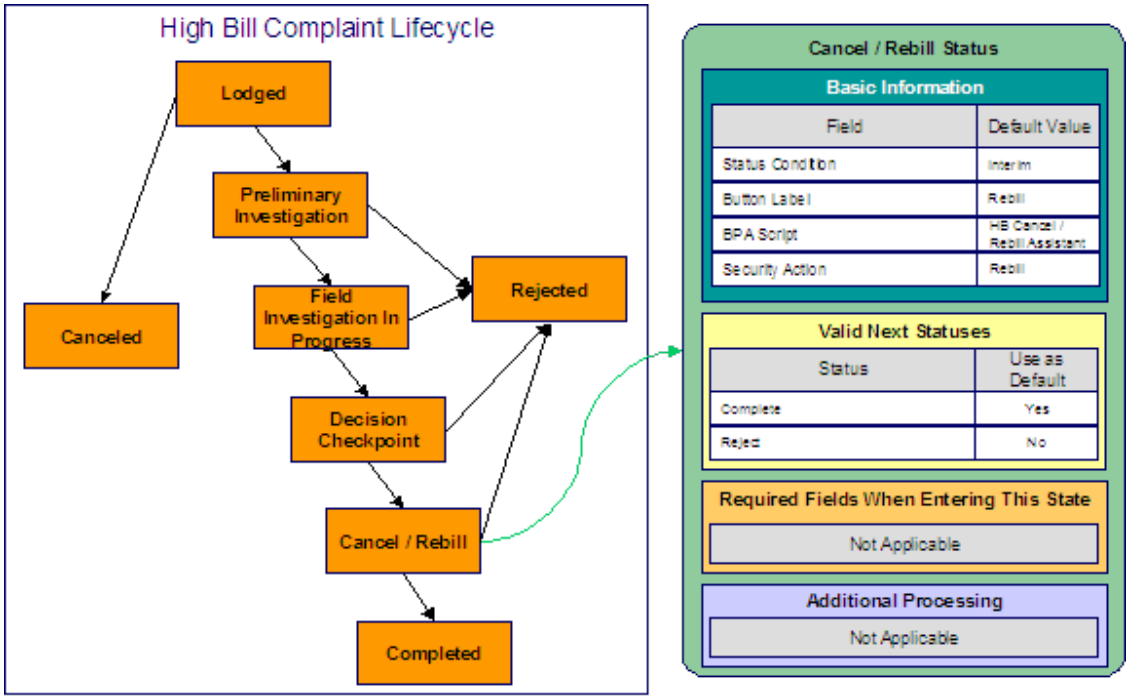
Note the following about this status:

- Decision Checkpoint is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Make Decision**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Field Investigation in Progress state as this is the only state that can transition into the Decision Checkpoint state.
- We have decided not to specify a BPA script on this status. Rather, we're going to set up the To Do type used to highlight cases in this state to automatically launch an appropriate BPA script when a user selects the To Do entry.
- We have associated the Make Decision action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state. Because the system will automatically transition cases into this state when the related field activity is complete, users will probably never press this button (and you may wish to prevent users from pressing this button be restricting security rights to the related action).
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Cancel / Rebill and Rejected states.
- Notice that the Additional Processing plug-ins create and complete a To Do entry when a case enters and exits this state, respectively.

Cancel Rebill High Bill Complaint

The following is an example of the configuration of the Cancel / Rebill status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



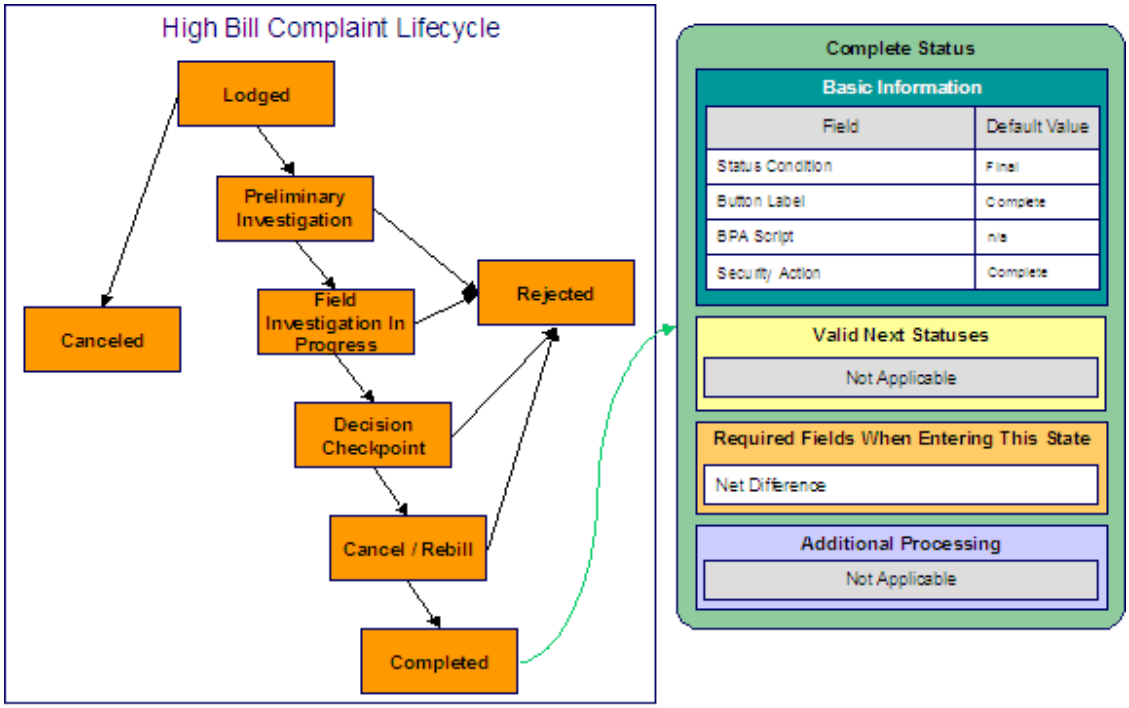
Note the following about this status:

- Cancel / Rebill is an interim state (meaning that it's not an initial or final state).
- It has a button label of **Cancel / Rebill**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Decision Checkpoint state as this is the only state that can transition into the Cancel / Rebill state.
- We have referenced a BPA script that can assist a user in the cancel / rebill efforts.
- We have associated the Rebill action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that Valid Next Statuses are what restricts a case in this state to be transitioned to the Completed and Rejected states.

Completed High Bill Complaint

The following is an example of the configuration of the Completed status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



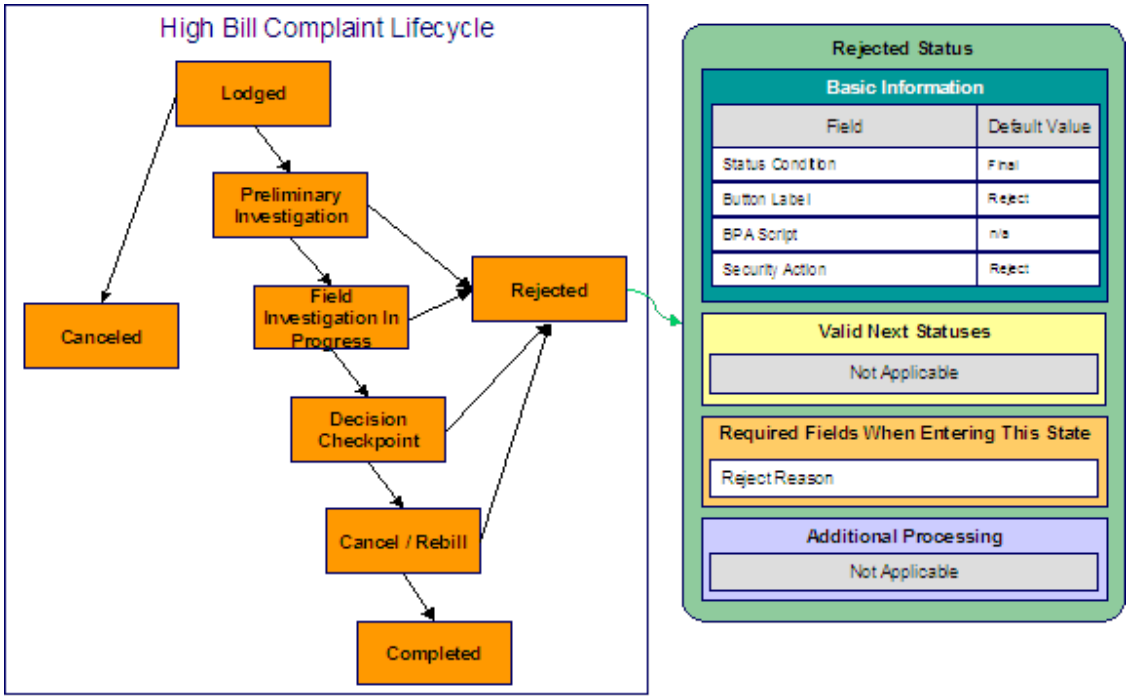
Note the following about this status:

- Completed is a final state (meaning that no further action is necessary).
- It has a button label of **Complete**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Cancel / Rebill state as this is the only state that can transition into the Completed state.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Complete action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Completed case to be reopened, you would need to define the state into which a Completed case could be transitioned.
- Notice that the **Net Difference** must be specified on the case before it can be moved into this state. This would be the difference to the customer's balance after the cancel rebill took place.

Rejected High Bill Complaint

The following is an example of the configuration of the Rejected status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



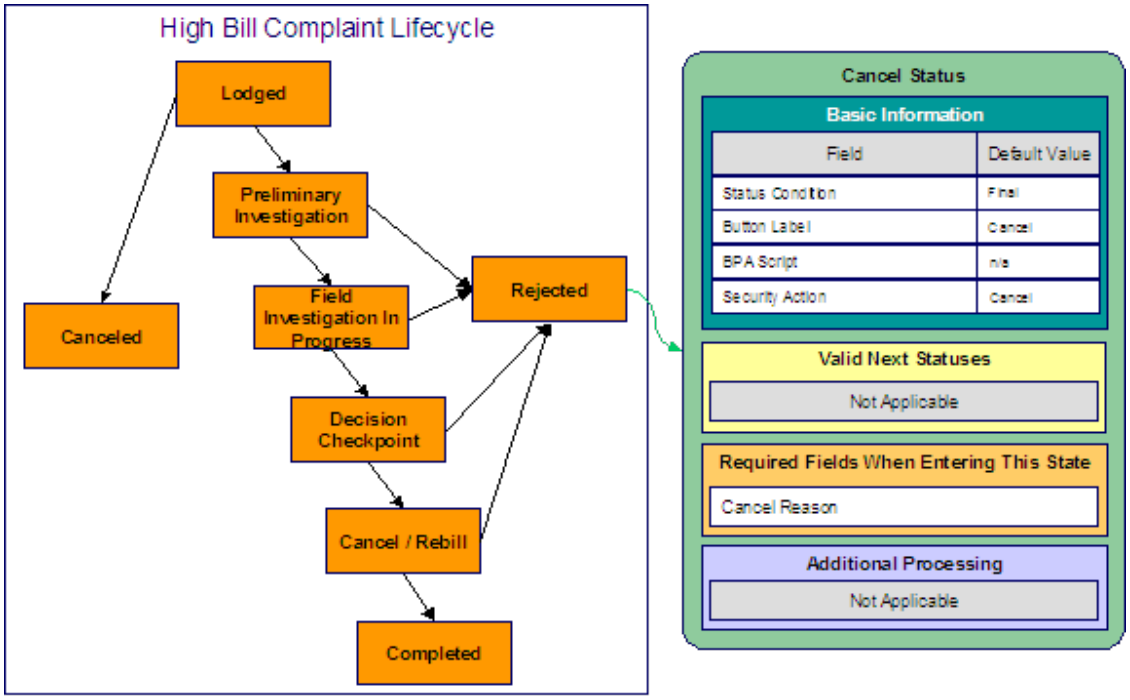
Note the following about this status:

- Rejected is a final state (meaning that no further action is necessary).
- It has a button label of **Reject**. This is the label on the button that a user presses to move a case into this state. This button will appear on cases that are in the Preliminary Investigation, Field Investigation in Progress , Decision Checkpoint and Cancel / Rebill states.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Reject action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Rejected case to be reopened, you would need to define the state into which a Rejected case could be transitioned.
- Notice that a **Reject Reason** must be specified on the case before it can be moved into this state.

Canceled High Bill Complaint

The following is an example of the configuration of the Canceled status for high bill complaint cases.

Copyright © 2026, Oracle and/or its affiliates.



Note the following about this status:

- Canceled is a final state (meaning that no further action is necessary).
- It has a button label of **Cancel**. This is the label on the button that a user presses to move a case into this state. This button will only appear on cases that are in the Lodged state as this is the only state that can transition into the Completed state.
- We have not referenced a BPA script because this is a final state (and no additional user action is necessary).
- We have associated the Cancel action with this status. This means that only users with rights to this action for the application service defined on the case type can move a case into this state.
- Notice that there are no Valid Next Statuses (because this is a final state). If you wanted to allow a Canceled case to be reopened, you would need to define the state into which a Canceled case could be transitioned.
- Notice that a **Cancel Reason** must be specified on the case before it can be moved into this state.

Setting Up Case Management Options

The topics in this section describe how to set up the system to enable case management.

> **Warning**
>
> The following topics assume you thoroughly understand the concepts described under The Big Picture Of Cases.

- Installation Options

Copyright © 2026, Oracle and/or its affiliates.

- Setting Up Application Services
- Setting Up Scripts
- Setting Up To Do Types
- Setting Up Characteristic Types
- Setting Up Case Types

Installation Options

- Case Info May Be Formatted By An Algorithm
- Alert Info Is Controlled By An Installation Algorithm

Case Info May Be Formatted By An Algorithm

An algorithm may be plugged in on the installation record to format the standard case information that appears throughout the system. Refer to CSIN-DFLT for an example of this algorithm.

This algorithm may be further overridden by a "Case information" plug-in on the Case Type. Refer to C1-CT-INFO for an example of this algorithm.

Alert Info Is Controlled By An Installation Algorithm

An algorithm that is plugged in on the installation record is responsible for formatting the alerts that highlight if the person / account / premise in context has open cases. Refer to CCAL-CASE for an example of this algorithm.

Setting Up Application Services

As described under Access Rights, you can prevent unauthorized users from accessing cases. The following points describe how to implement this type of security:

- Create an application service for each case type that needs to be secured
- Create an action on the application service for each status you need to secure
- Link the valid user groups to the application service and define which actions they can perform
- Define the application service on the case type
- Define the related action for each status on the case type / status

Setting Up Scripts

Copyright © 2026, Oracle and/or its affiliates.

As described under Scripts and Cases, BPA scripts can facilitate the creation and working of cases. Refer to the Defining Script Options for instructions describing how to set up scripts.

Setting Up To Do Types

As described under To Do's and Cases, To Do entries can be used to highlight cases that require user attention.

The following points provide a high-level description of how to create (and complete) To Do entries for a case type:

- Create a To Do type for each different type of To Do entry used during a case's lifecycle
  - On the To Do type, think carefully about the roles whose users can work on the entries
  - Also consider if you would like a BPA script launched when a user selects the entry
- Specify the To Do type on the appropriate entry processing or exit processing algorithm
- If you want the system to automatically complete To Do entries, specify the To Do type on the appropriate entry processing or exit processing algorithm

Please be aware that the case maintenance transaction highlights the number of open and assigned To Do entries linked to the case being displayed on the page. However, the system can only do this if the To Do entries reference a foreign-key characteristic whose foreign key references the case table. If you use the CSEN-TD algorithm to create To Do entries when a case enters a given state, this algorithm will do this for you if:

- You have set up a foreign-key characteristic type whose foreign key references the case table
- In addition, the characteristic type must reference a characteristic entity of To Do Entry

Setting Up Characteristic Types

As described under Additional Information, some of your cases may require additional information (in the form of characteristics). If this is true, you must set up the characteristic types before setting up the case types.

Refer to Setting Up To Do Types for instructions regarding a characteristic type that must be set up in order for the system to know the To Do entries that are associated with a case.

If you use the CSEN-CC algorithm to create customer contacts when a case enters a given state, you should set up a foreign-key characteristic type as follows:

- Its foreign key must reference the case table
- In addition, the characteristic type must reference a characteristic entity of Customer Contact

Setting Up Case Types

The case type maintenance transaction is used to maintain your case types. The topics in this section describe how to use this transaction.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to The Big Picture Of Cases for more information about how a case type encapsulates the business rules that govern a case.

- Case Type - Main
- Case Type - Case Characteristics
- Case Type - Lifecycle

Case Type - Main

Use this page to define basic information about a case type.

Open the case type page by selecting **Admin**, and then **Customer**, and then **Case Type**, and then **Case Type**, and then **Add.**

Note

Recommendation. Before using this transaction, we strongly recommend that you review the Examples of Case Types.

Main Information

Enter a unique **Case Type** code and **Description** for the case type.

Use **Long Description** to provide a more detailed explanation of the purpose of the case type.

If your implementation requires restricting case types by CIS Division, specify a **CIS Division** on the case type.

Person Usage controls the applicability of a person on cases of this type. Select Required if a person must be defined on this type of case. Select Optional if a person can optionally be defined on this type of case. Select Not Allowed if a person is not allowed on this type of case.

Account Usage controls the applicability of an account on cases of this type. Select Required if an account must be defined on this type of case. Select Optional if an account can optionally be defined on this type of case. Select Not Allowed if an account is not allowed on this type of case.

Premise Usage controls the applicability of a premise on cases of this type. Select Required if a premise must be defined on this type of case. Select Optional if a premise can optionally be defined on this type of case. Select Not Allowed if a premise is not allowed on this type of case.

Entity Relationship controls the validation of relationships on cases of this type. ***All*** ensures that the account belongs to the person and that the account and premise are related. ***Person and Account Only*** disables the account and premise validation. ***Premise and Account Only*** disables the person and account validation. ***None*** disables all validation and allows any values. This setting only impacts the validation of values entered and works in conjunction with the three usage fields described above.

Copyright © 2026, Oracle and/or its affiliates.

If you need to restrict access to cases of this type to specific user groups, reference the appropriate **Application Service**. Refer to Setting Up Application Services for the details of how to secure access to your cases.

If you are configuring a case type to handle the processing of data defined via a **Business Object**, associating the case type with a business object serves to link the properties of the business object (e.g. BO options) with cases of that type. Refer to Business Objects for further information. In addition, refer to Automatic Transition Rules for information on the role of BO options in case auto-transition errors.

Responsible User Usage controls the applicability of a responsible user on cases of this type. Select Required if a responsible user must be defined on this type of case. Select Optional if a responsible user can optionally be defined on this type of case. Select Not Allowed if a responsible user is not allowed on this type of case. Refer to Responsible User Applicability for more information.

Contact Information Fields

There are three contact information fields: **Contact Person & Method Usage**, **Contact Instructions Usage**, and **Case Phone Usage**. These fields are used to determine whether or not each type of contact information must be entered on case records with this case type. Select Required if the contact information must be entered, select Optional if the user can choose whether or not to include the contact information on this type of case, or select Not Allowed if the contact information cannot be entered on this type of case.

Algorithms

The **Algorithms** grid contains algorithms that control functions for cases of this type. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Case Information	Optional	We use the term "Case information" to describe the basic information that appears throughout the system to describe a case. The data that appears in "case information" is constructed using this algorithm. Plug an algorithm into this spot to override the "Case information" algorithm on installation options or the system default "Case information" if no such algorithm is defined on installation options.

Case Type Tree

Copyright © 2026, Oracle and/or its affiliates.

The tree summarizes the case type's lifecycle. You can use the hyperlinks to transfer you to the **Lifecycle** tab with the corresponding status displayed.

Case Type - Case Characteristics

To define characteristics that can be defined for cases of this type, open **Admin**, and then **Customer**, and then **Case Type**, and then **Search** and navigate to the **Case Characteristics** page.

Description of Page

Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on cases of this type. Turn on the **Default** switch to default the **Characteristic Type** when cases of this type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** box is checked. Refer to Required Fields Before A Case Enters A State for a description of how you can make option characteristics required at later stages in a case's lifecycle.

Case Type - Lifecycle

Case types that involve multiple users and multiple potential outcomes have complex lifecycles. Before you can design a case type's lifecycle, it's important that you thoroughly understand the concepts described under Lifecycle and Status-Specific Business Rules. After thoroughly understanding these concepts, we recommend you perform the following design steps:

- Draw a "state transition diagram" as illustrated above under Lifecycle. Keep in mind that if your state transition diagram is complex, your cases will be complex. While some cases warrant complexity, you should always ask yourself if there aren't better ways to achieve the desired results if your first effort results in complexity.
- Determine which characteristics (if any) are required during each stage of a case's lifecycle
- Determine when To Do entries (if any) should be created (and completed) during a case's lifecycle
- Determine additional validation (if any) that should be executed before a case enters and exits each state
- Determine additional processing (if any) that should transpire when a case enters or exits each state
- Determine if scripts are warranted to help users work the cases and, if so, design the scripts for each applicable state

When the above tasks are complete, you will be ready to set up a case type's lifecycle.

Open the Lifecycle page by selecting **Admin**, and then **Case Type**, and then **Case Type**, and then **Search** and navigate to the **Lifecycle** page.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can navigate to a status by clicking on the respective node in the tree on the Main tab. You can also use the hyperlinks in the Next Statuses grid to display a specific status in the accordion.

Main Information

The **Status** accordion contains an entry for every status in the case type's lifecycle.

Use **Status** to define the unique identifier of the status. This is NOT the status's description, it is simply the unique identifier used by the system.

Use **Description** to define the label that appears on the lifecycle accordion as well as the status displayed on the case.

Use **Script** to reference a BPA script that can assist a user work on a case while it's in this status. Refer to A Script That Helps A User Work Through A Case for the details.

Use **Access Mode** to define the action associated with this status. This field is disabled if an application service is not specified on the Main page. Refer to Access Rights for the details of how to use this field to restrict which users can transition a case into this state.

Use **Batch** to specify a batch control that will auto-transition the case. Any case in a status configured with a batch control will be transitioned when the batch job runs (rather than when CASETRAN is executed). For this purpose, batch process C1-CSTRS (**Case Scheduled Transition**) is supplied with base package, which will execute all Exit Status logic for the current status, and Enter Status logic for the destination status. You may choose to create a batch process with your own transition logic.

> **Note**
>
> If you wish to defer transitioning a case in a particular status until the batch process on your case type status is executed, you should not populate an Auto-Transition algorithm on that status. Otherwise, CASETRAN will transition the case according to your Auto-Transition logic.

Use **Comment** to describe the status. This is for your internal documentation requirements.

Use **Sequence** to define the relative order of this status in the tree on the Main page.

Use **Status Condition** to define if this status is an Initial, Interim or Final state. Refer to One Initial State and Multiple Final States for more information about how this field is used.

Use **Transitory State** to indicate whether a case should ever exist in this state. Only Initial or Interim states can have a transitory state value of No.

The **Alert Flag** is used to indicate whether or not an alert should be displayed for customers with cases in the state. (The alert is shown via the base package Installation - Alert algorithm.)

Algorithms

The **Algorithms** grid contains algorithms that control important functions for cases of this type. You must define the following for each algorithm:

Copyright © 2026, Oracle and/or its affiliates.

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Description
Auto Transition	This algorithm is executed to determine if a case that's in this state should be transitioned into another state. Refer to Automatic Transition Rules for the details.
Enter Processing	This algorithm holds additional processing that is executed when a case is transitioned into this state. You can also specify state transition logic within Enter Processing routines. Refer to Additional Processing When Entering A State for the details.
Enter Validation	This algorithm holds validation logic that executes before a case can enter a given state. Refer to Validation Before A Case Enters A State for the details.
Exit Processing	This algorithm holds additional processing that is executed when a case is transitioned out of this state. Refer to Additional Processing When Exiting A State for the details.
Exit Validation	This algorithm holds validation logic that executes before a case can by transitioned out of a given state. Refer to Validation Before A Case Exits A State for the details.
Script Launching	This algorithm sets the script launching option for the script associated with a given state, if any. Refer to Script Launching Option for the details.

Next Statuses

Use the **Next Statuses** grid to define the statuses a user can transition a case into while it's in this state. Refer to Valid States versus State Transition Rules for more information. Please note the following about this grid:

- Use **Action Label** to indicate the verbiage to display on the action button used to transition to this status.
- **Sequence** controls the order of the buttons that appear on Case - Main. Refer to Buttons Are Used To Transition A Case Into A State for more information.
- **Use as Default** controls which button (if any) is the default button.
- **Transition Condition** may be configured to identify a common transition path for cases of this type in the current state. This transition condition may then be referenced across multiple case types. You'll need to add values to Look Up table field **TR_COND_FLG** that fit the typical transitions for your case types (e.g. Ok, Error, etc.).

By assigning the transition condition value to a given "next status", you can design your Enter State transition or Auto-Transition logic to utilize those flag values *without specifying a status*

Copyright © 2026, Oracle and/or its affiliates.

particular to a given case type. Thus, similar logic may be used across a range of case types to transition a case into, for example, the next Ok state for the case's current status.

- **Transition Role** controls whether only the System or both System and User have the ability to transition a case into a given "next status".
- You can use the status description hyperlink to open the Status accordion to the respective status.
- When you initially set up a case type, none of the statuses will reside on the database and therefore you can't use the search to define a "next status". We recommend working as follows to facilitate the definition of this information:
  - Leave the Next Statuses grid blank when you initially define a case type's statuses
  - After all statuses have been saved on the database, update each status to define its Next Statuses (this way, you can use the search to select the status).

Required Characteristics

Use the **Required Characteristics** grid to define characteristics that are required when a case enters this state. Only Optional characteristics defined on the main page appear in this grid. Refer to Required Fields Before A Case Enters A State for more information.

Defining Umbrella Agreement Options

Your organization may use umbrella agreement functionality to manage many situations, including (but not limited to) the following:

- Creating a "negotiated contract" with a large customer with many sites.
- Creating a "negotiated contract" with all customers in a common geographic area, for example a specific city or an apartment complex
- Grouping and managing the proposal service agreements created during the quoting process
- Managing the renewal process for special contracts
- Applying override rate terms to one or more service agreements for a given effective period
- more...

> **Note**
>
> **Separately module.** The umbrella agreement functionality is part of the Contract Management module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information.

- The Big Picture Of Umbrella Agreements
- Setting Up Umbrella Agreement Options
- Umbrella Agreements Configuration Examples

Copyright © 2026, Oracle and/or its affiliates.

The Big Picture Of Umbrella Agreements

The topics in this section provide background information about how to configure the system to support your umbrella agreement requirements.

- Renewable Umbrella Agreements
- Terms Of Service Covered Entities
- Overriding Rate Terms

Renewable Umbrella Agreements

When defining your umbrella agreement types, you must determine if umbrella agreements of this type are renewable. If so, you must set the renewal flag on the UA type to Renewable and determine the number of days prior to the end date of the umbrella agreement that the renewal process should start.

In addition, you must decide on your renewal procedure and design an appropriate renewal algorithm. You may use one of the sample renewal algorithms provided by the system.

- The sample algorithm URNW-TD creates a To Do entry. Using this algorithm you can direct the To Do entry to an appropriate role to determine if the umbrella agreement should be renewed. If the umbrella agreement is associated with an account management group, the To Do entry could be assigned to a role defined for that AMG.
- The sample algorithm URNW-CA creates a case. Using this algorithm you can create an appropriate case to help track the steps required for renewing an umbrella agreement.

If the base algorithms do not provide the logic your organization needs, you may create a new algorithm.

> **Note**
>
> It is the responsibility of the renewal algorithm to change the renewal date as per the business requirements. If renewal requires manual review, the algorithm should reset the date to blank. If the algorithm can determine a new renewal date, the date should be changed to the appropriate value.

Terms Of Service Covered Entities

For each terms of service type you design, consider whether or not a TOS created for this type should reference one or more entities in the system that are "covered" by the terms of service. Following are some examples of covered entities:

- A special contract is created for all the McDonalds franchises in your service area. When creating the terms of service record for the umbrella agreement used for this contract, you want the user to link the person record representing the McDonalds corporate entity as a covered entity for the TOS.

Copyright © 2026, Oracle and/or its affiliates.

- An umbrella agreement is created for a large customer with several types of services. Perhaps you want to group the services in the contract based on type of service. The service type may be set up as a covered entity.

The covered entities could also be used to denote eligibility. Refer to Linking To A UA For A Group Of Premises for an example of using the covered entity collection for eligibility.

The covered entities are actually characteristics. You must define the desired characteristic types if they do not already exist. In this case, they would be foreign key characteristic types. The characteristic types that you want to use must reference a characteristic entity of TOS covered entity.

A terms of service record may only reference covered entity types that you designate on the Terms of Service type record. When designing the covered entities that are allowed for TOS records of a given type, you may also configure the Terms of Service type to indicate that one or more covered entity types are required.

Refer to Terms Of Service Type - Main for more information.

Overriding Rate Terms

If your organization has business rules that require a service agreement's rate terms to be overridden, you can accomplish this using umbrella agreements and terms of service records. There are two levels of overriding that are possible:

- Override rate schedule
- Override other terms on the rate, including contract riders, contract terms and tax exemptions

To override any rate terms, the service agreement must be linked to a terms of service record that references a template service agreement. In addition, you must configure settings in the system to indicate whether or not overriding of terms is allowed.

- The SA type for the customer's service agreement indicates whether or not the rate schedule may be overridden using the rate source flag.
  - If the rate source value is Check TOS First, then SA, when applying the rate, the system determines if the SA is linked to a TOS effective at the time of the bill with a template SA. If so, the rate schedule on the template SA is used.
  - If the rate source value is Check SA Only, when applying the rate, the always uses the rate schedule on the customer's SA (regardless of whether or not the SA is linked to a TOS effective at the time of the bill with a template SA).
- If any bill factor on one of calculation rule indicates that it is eligible for tax exemptions, contract terms or contract riders, the terms of service usage flag on the bill factor indicates where the system should look for the appropriate value. The flag tells the system to find applicable information either on the TOS's template SA only, on the customer's SA only or it should first check the TOS, then check the customer's SA.

The following table illustrates the different scenarios possible when configuring your bill factors and the resulting behavior. The term "standard behavior" in the tables below indicate that the standard behavior as described in Defining Bill Factors is followed.

	TOS Usage	Template SA	Customer SA	BF Char Source

Copyright © 2026, Oracle and/or its affiliates.

Contract Rider Applicability is checked	Check TOS First, then SA	Found	Not checked	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed.
		Not Found	Found	If source = SA, the customer SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
		Not Found	Not Found	N/A
	Check TOS Only	Found	Not checked	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
		Not Found	Not checked	N/A
	Check SA Only	Not checked	Standard behavior	

	TOS Usage	Template SA	Customer SA	BF Value Source	BF Char Source
Value in Contract Term is checked	Check TOS First, then SA	Found	Not checked	Template SA	N/A
		Not Found	Found	Customer SA	N/A
		Not Found	Not Found	Bill Factor Value (based on Char Source)	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
	Check TOS Only	Found	Not checked	Template SA	N/A

Copyright © 2026, Oracle and/or its affiliates.

		Not Found	Not checked	Bill Factor Value (based on Char Source)	If source = SA, the template SA's characteristic collection is checked. If source <> SA, the standard behavior is followed
	Check SA Only	Not checked	Standard behavior		

	TOS Usage	Template SA	Customer SA	BF Value Source
Tax Exemption is checked	Check TOS First, then SA	Found	Not checked	Template SA
		Not Found	Found	Customer SA
		Not Found	Not Found	No value
	Check TOS Only	Found	Not checked	Template SA
		Not Found	Not checked	No value
	Check SA Only	Not checked	Standard behavior	

Note that proration logic is followed for the above functionality as expected. For example, imagine that a bill factor with Value in Contract Terms is proratable and the TOS usage value is Check TOS First, then SA. If the template SA has a contract value that is only in effect for the first 10 days of a bill, that value is used for the first 10 days and the remaining days in the bill take the value from the SA (if applicable) or the bill factor value as per the table rules above.

> **Note**
>
> When designing your terms of service type, you must indicate whether a template service agreement is optional, required or not allowed. When the template SA is required, the user setting up the TOS record for this TOS type must link the correct service agreement as the template SA. If a new template service agreement is required, the new SA must be created first.

Setting Up Umbrella Agreement Options

The topics in this section describe how to set up umbrella agreement options.

- Configure SA Types for Umbrella Agreements
- Configure Bill Factors for Umbrella Agreements
- Configure A Campaign To Link An SA To A Terms Of Service
- Setting Up Umbrella Agreement Types
- Setting Up Terms Of Service Types
- Setting Up Terms of Service Cancel Reasons

Copyright © 2026, Oracle and/or its affiliates.

Configure SA Types for Umbrella Agreements

For each type of service agreement that may be linked to a terms of service record, you must configure the Rate Source for the SA type to indicate if the rate should always be taken from the service agreement (Check SA Only) or if the rate should be taken from the terms of service record for the service agreement, if any (Check TOS First, then SA).

NOT_SUPPORTED

Refer to Overriding Rate Terms for more information.

Note

The rate source is only visible on SA type if the Contract Management module is not turned off.

Configure Bill Factors for Umbrella Agreements

For each bill factor that has a value in contract term, contract rider applicability or tax exemption applicability, you must indicate where the system should look for a value when a service agreement is linked to a terms of service with a template SA and where this bill factor is referenced by the rate being applied. Using the terms of service usage flag, tell the system to Check TOS Only, Check SA Only or Check TOS First, then SA.

NOT_SUPPORTED

Refer to Overriding Rate Terms for more information.

Note

The terms of service usage flag is only visible on bill factor if the Contract Management module is not turned off.

Configure A Campaign To Link An SA To A Terms Of Service

The system provides a pair of column reference algorithm types that support linking a new service agreement to an existing terms of service record via an order. The pair includes a validation algorithm type CRVL-SATOS and a posting algorithm type CRPS-SATOS.

In order to take advantage of this functionality, you must:

Copyright © 2026, Oracle and/or its affiliates.

- Create an algorithm for each of the above algorithm types. Only one algorithm is needed for each algorithm type because there are no parameters for either one.
- Setup a unique column reference for the terms of service ID column reference code and plug in the new validation and posting algorithms on the record.
- For campaigns whose orders link a new service agreement to an existing umbrella agreement, make sure to define a miscellaneous field for the terms of service ID. Note, the miscellaneous field must reference the column reference that you setup above.

Based on your business practice, you may want to further configure your campaign / package for linking a service agreement to an existing terms of service. Refer to Linking A Service Agreement To A TOS Through Orders for more information.

Setting Up Umbrella Agreement Types

The umbrella agreement type transaction is used to maintain your UA types. The topics in this section describe how to use this transaction.

- Umbrella Agreement Type - Main
- Umbrella Agreement Type - Algorithms

Umbrella Agreement Type - Main

Open **Admin**, and then **Umbrella Agreement**, and then **Umbrella Agreement Type**, and then **Add** to define basic information about an umbrella agreement type.

Description of Page

Enter a unique **Umbrella Agreement Type** code and **Description** for the UA type.

Use **Renewal** to indicate whether an umbrella agreement of this type is Renewable or Not Renewable. For renewable UA types, indicate the **Renewal Days Before Expiration**. The renewal date for umbrella agreements of this type are calculated based on this number of days before the end date.

Use the characteristics collection to define characteristics that can be defined for umbrella agreements of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on umbrella agreements of a given type. Turn on the **Default** switch to default the **Characteristic Type** when umbrella agreements of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined as permissible on an umbrella agreement record. Refer to *Setting Up Characteristic Types & Their Values* for more information.

Enter the valid **Terms of Service Type**s that may be referenced for umbrella agreements of this type.

Copyright © 2026, Oracle and/or its affiliates.

Umbrella Agreement Type - Algorithms

Open **Admin**, and then **Umbrella Agreement**, and then **Umbrella Agreement Type**, and then **Search** and navigate to the **Algorithms** page.

Description of Page

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (descriptions of all possible events are provided below).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** for which you can define algorithms.

System Event	Description
UA Renewal	Algorithms of this type are executed by the UARENEW background process when an umbrella agreement has reached its renewal date. Only one renewal algorithm is allowed for a given UA type.

Setting Up Terms Of Service Types

The terms of service type transaction is used to maintain your Terms of Service types. The topics in this section describe how to use this transaction.

- Terms Of Service Type - Main
- Terms Of Service Type - UA Types

Terms Of Service Type - Main

Open **Admin**, and then **Umbrella Agreement**, and then **Terms of Service Type**, and then **Add** to define basic information about terms of service types.

Description of Page

Enter a unique **Terms of Service Type** code and **Description** for the TOS type.

Use **Template SA Usage** to indicate whether a template SA is Optional, Required or Not Allowed. Refer to Overriding Rate Terms for more information about using template SAs.

Use the characteristics collection to define characteristics that can be defined for terms of service records of a given type. Use **Sequence** to control the order in which characteristics are defaulted. Turn on the **Required** switch if the **Characteristic Type** must be defined on terms of service records of a given type. Turn on the **Default** switch to default the **Characteristic Type**

Copyright © 2026, Oracle and/or its affiliates.

when terms of service records of the given type are created. Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on.

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined as permissible on a terms of service record. Refer to *Setting` Up Characteristic Types & Their Values* for more information.

Use the **Covered Entities** collection to define the types of entities that are "covered" by terms of service records of this type. Refer to Terms Of Service Covered Entities for more information

> **Note**
>
> **Characteristic Types.** You can only choose characteristic types defined as permissible for TOS covered entity. Refer to *Setting Up Characteristic Types & Their Values* for more information.

Terms Of Service Type - UA Types

Open **Admin**, and then **Umbrella Agreement**, and then **Terms of Service Type**, and then **Add** and navigate to the **UA Types** page.

Description of Page

Enter the valid **Umbrella Agreement Type**s that may reference this terms of service type.

Setting Up Terms of Service Cancel Reasons

Whenever a terms of service record is canceled, a cancel reason must be specified. Open **Admin**, and then **Terms of Service Cancel Reason** to specify such cancellation reasons.

Description of Page

Enter an easily recognizable **Terms of Service Cancel Reason** and **Description** for the terms of service cancellation reason.

Umbrella Agreements Configuration Examples

This section provides examples of how you would configure the system to support various business scenarios.

- Linking A Service Agreement To A TOS Through Orders

Copyright © 2026, Oracle and/or its affiliates.

Linking A Service Agreement To A TOS Through Orders

As described in Configure A Campaign To Link An SA To A Terms Of Service, the system has provided a pair of column reference algorithms to enable your users to link a new service agreement to an existing terms of service record (and its umbrella agreement) when starting service via the order transaction.

When designing your campaign and packages to use this option, consider the scenarios where you may use this to determine whether or not you require additional configuration. The following topics describe some possible scenarios.

- Linking To A UA For A Given Account
- Linking To A UA For A Group Of Premises

Linking To A UA For A Given Account

Imagine that you would use this logic when an important customer is starting additional service and has an existing umbrella agreement for other services that the new service should be added to. The user creating the order is probably the account manager for that account and would know the correct terms of service record to link the new service to.

For this scenario, you do not need to provide any "help" to the user to indicate that linking to a TOS is an option or to help select the correct TOS. You simply need to ensure that the campaign that would be used by this account manager to start service for the important customer simply includes a question / miscellaneous field to link the SA to the terms of service.

Linking To A UA For A Group Of Premises

Imagine you have created an umbrella agreement for a group of premises that are not actively managed by a specific user. For example, perhaps there are several apartment complexes that have negotiated a special rate with your company. When new customers sign up for service at a premise that belongs to one of these apartments, the user should be informed that there is a special contract for this premise. The user may also need help selecting the correct terms of service record. To accomplish this logic, you must define additional algorithms to help your users.

For example, perhaps your standard campaign for residential electric customers includes a package to be used by premises linked to one of these apartment buildings and should only be eligible to one of those premises. You must create an eligibility algorithm for this package. Its logic could be designed based on the following assumptions:

- The qualifying premises are linked to a parent premise
- The parent premise is linked to the appropriate terms of service record in its covered entity collection

The eligibility algorithm checks to see if the order's premise is linked to a parent premise that is referenced on a terms of service record for an umbrella agreement that is currently in effect. It could also be restricted to search for a specific umbrella agreement type or terms of service type if appropriate.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The campaign for this scenario must include a question / miscellaneous field to link the SA to the terms of service. It should be configured with an applicability of only applicable on package. The package that is used to include the SA into the terms of service for the apartment complex would include this question / miscellaneous field.

Once the user chooses this package, user is prompted to provide the correct terms of service record via the question / miscellaneous field. To help the user, you should create a column reference retrieval algorithm to default the correct terms of service record for this premise. It would use the same logic as the eligibility algorithm described above. In other words, it would find the parent premise for this premise and then find the terms of service record covered entity collection includes this parent premise and whose umbrella agreement is currently in effect.

Reports Addendum

This chapter is an addendum to the general Defining and Designing Reports chapter.

- Configuring the System to Invoke Oracle Analytics Publisher Enterprise Real-Time
- Configuring the System for Batch Bill Print
- Description of Sample Reports

Configuring the System to Invoke Oracle Analytics Publisher Enterprise Real-Time

In order to invoke Oracle Analytics Publisher Enterprise from within the system, it must be configured. This topic describes the steps at a high level that are required to configure your system to enable BI Publisher.

- Specify Reporting Tool Options
- Reporting Tool Algorithm
- Interface with Oracle Analytics Publisher Enterprise Batch Scheduler

Specify Reporting Tool Options

The following information should be defined on Reporting Tool Options table:

- **Reporting Server** is the web server URL where the reporting tool is installed.
- **Reporting Folder** defines the name of the directory on the Oracle BI Publisher server where reports are located. The system's reporting tool algorithm constructs the URL to open Oracle Analytics Publisher Enterprise based on the information provided on Reporting Options. The same folder name should be used as a report folder in Oracle Analytics Publisher Enterprise where reports are published.

Copyright © 2026, Oracle and/or its affiliates.

Reporting Tool Algorithm

The base product provides an installation algorithm plug-in spot called Reporting Tool. This plug- in spot should contain an algorithm that invokes the reporting tool real-time. The corresponding algorithm type, called algorithm reference: F1-BIPR-INV, invokes Oracle Analytics Publisher Enterprise.

This algorithm relies on information defined in the Reporting Options table: the reporting server and folder names for accessing the reporting tool. The algorithm constructs several strings, for example, the URL of the Analytics Publisher, information about the report being requested and user info in the format expected by Oracle Analytics Publisher Enterprise. The Java program returns the correct URL to the browser. The browser then opens an Analytics Publisher Enterprise window with the appropriate URL.

To use the algorithm reference: F1-BIPR-INV algorithm type to invoke Oracle Analytics Publisher Enterprise, perform the following steps:

1. Create an algorithm for this algorithm type.
2. In the installation options, add an entry to the algorithm collection with an algorithm entity of Reporting Tool and indicate the algorithm created in the previous step.

Interface with Oracle Analytics Publisher Enterprise Batch Scheduler

Presently the base product does not provide support for batch jobs with Oracle Analytics Publisher. Please use Oracle Analytics Publisher directly in order to schedule reports to run in batch.

Configuring the System for Batch Bill Print

This section describes the steps for configuring the applications to print bills in batch.

- Configure Oracle BI Publisher for Batch Bill Print
- Configure Oracle Utilities Customer Care and Billing for Batch Bill Print

Configure Oracle BI Publisher for Batch Bill Print

> **Note**
>
> Using BI Publisher for bill print is only supported with on-premises implementations. Cloud service implementations must use a 3rd party bill print system.

1. Configure the Oracle BI Publisher Scheduler to allow report job submission.

> **Note**
>
> Refer to the Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher for more information.

Copyright © 2026, Oracle and/or its affiliates.

2. Ensure that the CI_BILLPR report artifacts (CI_BILLPR.rtf, CI_BILLPR.xdo, CI_BILLPR.xdm) are deployed to corresponding BI Publisher catalog folder:

Configure Oracle Utilities Customer Care and Billing for Batch Bill Print

1. Go to **Admin**, and then **Reporting**, and then **Reporting Options** and configure the following values:
   - **Reporting Server from Browser** should point to the correct BI Publisher Server URL; for example, `http://<hostname>:port`
   - **Reporting Folder** is the name as the Catalog Folder in BI Publisher where CI_BILLPR is located; for example, CCB
   - **Reporting Engine User ID** is the BI Publisher User ID.
   - **Reporting Engine Password** is the BI Publisher Password.
2. Create a custom Bill Route Type Extract Algorithm using the C1-BLEX-BIP Algorithm Type. Ensure that the following algorithm soft parameters are set:
   - **Output Directory** is the directory in BI Publisher server where the generated output files will be placed, for example. `/tmp/`
   - **Report Absolute Path** is the path where CI_BILLPR.xdo is located in the BI Publisher Catalog, for example, `/CCB/CI_BILLPR/CI_BILLPR.xdo`
   - **Report Format** is the format of the generated output files; for example, `pdf`
   - **Report Code** is the CI_BILLPR report code; for example, CI_BILLPR
3. **Go to Admin Menu, B, Bill Route Type** and configure the corresponding Bill Route Type (such as, `POSTAL`) to use the RTTYPOST Batch Control and the custom Bill Route Type Extract Algorithm.

Description of Sample Reports

This section provides an overview of each sample report supplied with Oracle Utilities Customer Care and Billing that may be found in the demonstration database. They may be used by your organization as they are or as a starting point for creating a new report.

> **Note**
>
> **Account Security**. The sample reports provided with the product do NOT incorporate account security. If a user has been given security to view the report, then all the data in the report is available for viewing.

- Active Severance Processes - CI_ACSVPR
- Bill Print in BI Publisher - CI_BILLPR
- Billed Revenues by Rate - CI_BILREV
- Case Statistics By Case Type - CI_CSESTS
- Case Statistics for a Given Status - CI_CSESGS

Copyright © 2026, Oracle and/or its affiliates.

- Collection Summary - CI_CLLSUM
- Customer Contacts by Type - CI_CUSTCN
- Customers with Life-Support / Sensitive-Load - CI_PMLSSL
- GL Accounting Summary - CI_GLACSM
- Letter Print - BI Publisher Sample Welcome Letters - CI_LTRGN_ENG REPORT
- Open Cases By Type - CI_CSEOPN
- Payments Balance - CI_PMTBAL
- Receivables Aging - CI_RCVAGA
- Tax Payables Analysis - CI_TXPYBL
- To Do Entries - CI_TDENTR
- Umbrella Agreement Summary - CI_UASUMM
- Unbilled Revenues - CI_UBLREV

Active Severance Processes - CI_ACSVPR

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Start date to use for reporting severance processes. If not defined, start date is set to 1900-01-01.
End Date	P_TO_DT	End date to use for reporting active severance processes. If not defined to date is set to the current date.

Report Description

This report is used to aid in the monitoring of active severance processes. The report details are used to review how the company is doing as far as the collection of outstanding debt, as well as to monitor the progress of currently active severance processes.

The report selects active severance processes whose creation date is between the input start and end dates.

Bill Print in BI Publisher - CI_BILLPR

> **Note**
>
> Using BI Publisher for bill print is only supported with on-premises implementations. Cloud service implementations must use a 3rd party bill print system.

Parameters

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Parameter Code	Description
CC&B user	P_USER_ID	No default value. Required. Returns an error message if user ID is null, empty or does not exist in database.
Batch Switch	P_BATCH_SW	For a single bill print, Batch Switch is 'No' and for multiple bill print, Batch Switch is 'Yes'.
First Bill	**P_FROM_BILL_ID**	First Bill Id to print for the Bill Range.
Last Bill	**P_TO_BILL_ID**	Last Bill Id to print for the Bill Range.
Batch Code	**P_BATCH_CD**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing that references this batch code are selected.
Batch Number	**P_BATCH_NBR**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing that references this batch number are selected.
Extract Algorithm	**P_EXTRACT_ALG**	Passed in by the system the report is submitted in batch. Only bills with a Bill Routing whose bill route type references this extract algorithm are selected.

Report Description

This report is used with BI Publisher to display all the information that appears on the customer's printed bill for the bill ID entered in the input parameters. Refer to the description of the parameters above. The bill is only displayed if the bill is complete.

See below for details of the information extracted for the report. In addition, the sample report layout provided for the base product includes:

- Payment coupon.
- Form for change of address.
- A graph of consumption for current charges.

The following sections highlight the data that is extracted for this report.

- Bill Information
- Bill Segment Information
- Account Summary Information
- Current Charge Information
- Premise Information
- Auto Pay Information
- Bill Segment Readings
- Bill Segment Consumption History

Copyright © 2026, Oracle and/or its affiliates.

- Payment Information
- Adjustment Information
- Bill Message Information

Bill Information

Field Name	Format	Source/Value/Description
SORT_KEY	A12	Primary sort for the report - currently sorted by Bill Id
BILL_ID	A12	CI_BILL
SEQNO	N5	CI_BILL_ROUTING
PER_ID	A10	CI_BILL_ROUTING
NBR_BILL_COPIES	N1	CI_BILL_ROUTING
ENTITY_NAME1	A64	CI_BILL_ROUTING. Mailing Entity Name 1.
ENTITY_NAME2	A64	CI_BILL_ROUTING. Mailing Entity Name 2.
ENTITY_NAME3	A64	CI_BILL_ROUTING. Mailing Entity Name 3.
COUNTRY	A3	CI_BILL_ROUTING
ADDRESS1	A254	CI_BILL_ROUTING
ADDRESS2	A254	CI_BILL_ROUTING
ADDRESS3	A254	CI_BILL_ROUTING
ADDRESS4	A254	CI_BILL_ROUTING
CITY	A30	CI_BILL_ROUTING
COUNTY	A30	CI_BILL_ROUTING
STATE	A6	CI_BILL_ROUTING
POSTAL	A12	CI_BILL_ROUTING
IN_CITY_LIMIT	A1	CI_BILL_ROUTING
BATCH_CD	A8	CI_BILL_ROUTING
BATCH_NBR	N10	CI_BILL_ROUTING
NO_BATCH_PRT_SW	A1	CI_BILL_ROUTING
ACCT_ID	A10	CI_BILL
BILL_STAT_FLG	A2	CI_BILL
BILL_DT	Date	CI_BILL
DUE_DT	Date	CI_BILL
ACCT_ENTITY_NAME	A64	Get Main Person from CI_PER_NAME. Account Entity Name
LANGUAGE_CD	A3	Person's Language from CI_PER
CURRENCY_CD	A3	CI_ACCT
CUR_SYMBOL	A4	CI_CURRENCY_CD
DECIMAL_POSITIONS	N1	CI_CURRENCY_CD

Copyright © 2026, Oracle and/or its affiliates.

CUR_POS_FLG	A2	CI_CURRENCY_CD
OPEN_ITEM_SW	A1	CI_CUST_CL
BILL_COPY	A1	Bill copy number

Bill Segment Information

Field Name	Format	Source/Value/Description
BSEG_ID	A12	CI_BSEG
SA_ID	A10	CI_BSEG
START_DT	Date	CI_BSEG
END_DT	Date	CI_BSEG
BILL_CYC_CD	A4	CI_BSEG
WIN_START_DT	Date	CI_BSEG
EST_SW	A1	CI_BSEG
CLOSING_BSEG_SW	A1	CI_BSEG
PREM_ID	A10	CI_BSEG
CIS_DIVISION	A5	CI_SA
SA_TYPE_CD	A8	CI_SA
SAT_DESCR	A60	CI_SA_TYPE_L
BILL_PRT_PRIO_FLG	A2	CI_SA_TYPE
GRAPH_UOM_CD	A4	CI_SA_TYPE
HEADER_SEQ	N3	CI_BSEG_CALC
BCALC_START_DT	Date	CI_BSEG_CALC
BCALC_END_DT	Date	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
EFFDT	Date	CI_BSEG_CALC
BILLABLE_CHG_ID	A12	CI_BSEG_CALC
CALC_AMT	N15.2	CI_BSEG_CALC
BCALC_DESC_BILL	A80	CI_BSEG_CALC
SEQNO	N5	CI_BSEG_CALC_LN
PRT_SW	A1	CI_BSEG_CALC_LN
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_LN_AMT	N15.2	CI_BSEG_CALC_LN
DST_ID	A10	CI_BSEG_CALC_LN
MSR_PEAK_QTY_SW	A1	CI_BSEG_CALC_LN
DESCR_LN	A80	CI_BSEG_CALC_LN
STATE	A6	CI_PREM
CITY	A30	CI_PREM
POSTAL	A12	CI_PREM
SIBLING_ID	A12	CI_FT

Copyright © 2026, Oracle and/or its affiliates.

Account Summary Information

Field Name	Format	Source/Value/Description
CURRENCY_CD	A3	Derived from CI_BILL_SA or CI_FT.
SUM_CUR_AMT	S15.2	Get Bill's Current Balance from CI_BILL_SA Bill's Current Charge from CI_FT Bill's Current Correction Charge from CI_FT Bill's Current Adjustment from CI_FT Bill's Current Payment from CI_FT
SUM_TOT_AMT	S15.2	Get Bill's Total Balance from CI_BILL_SA Bill's Total Charge from CI_FT Bill's Total Correction Charge from CI_FT Bill's Total Adjustment from CI_FT Bill's Total Payment from CI_FT
BILL_PRT_FLG	A4	Set as follows: ASBL - Indicator for Bill's Balance from CI_BILL_SA ASBS - Indicator for Bill's Current Charge from CI_FT ASBC - Indicator for Bill's Correction from CI_FT ASAD - Indicator for Bill's Adjustment from CI_FT ASPS - Indicator for Bill's Payment from CI_FT

Current Charge Information

Field Name	Format	Source/Value/Description
SUM_CUR_AMT	S15.2	Derived from CI_FT (Only the current charge bill segments.)
SUM_TOT_AMT	S15.2	Derived from CI_FT (Only the current charge bill segments.)
CURRENCY_CD	A3	CI_BILL_SA or CI_FT
DEBT_CL_CD	A4	Debt Class of SA Type
DESCR	A60	Debt Class Description

Copyright © 2026, Oracle and/or its affiliates.

Premise Information

One record for every premise linked to the bill.

Field Name	Format	Source/Value/Description
PREM_ID	A10	Premise Id obtained from Bill Segment
PREM_TYPE_CD	A8	CI_PREM
LL_ID	A10	CI_PREM
MAIL_ADDR_SW	A1	CI_PREM
TREND_AREA_CD,	A8	CI_PREM
COUNTRY	A3	CI_PREM
ADDRESS1	A254	CI_PREM
ADDRESS2	A254	CI_PREM
ADDRESS3	A254	CI_PREM
ADDRESS4	A254	CI_PREM
CITY	A30	CI_PREM
NUM1	A6	CI_PREM
NUM2	A4	CI_PREM
HOUSE_TYPE	A2	CI_PREM
COUNTY	A30	CI_PREM
STATE	A6	CI_PREM
POSTAL	A12	CI_PREM
GEO_CODE	A11	CI_PREM
IN_CITY_LIMIT	A1	CI_PREM

Auto Pay Information

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_APAY_CLR_STG
APAY_SRC_CD	A12	CI_APAY_CLR_STG
EXT_ACCT_ID	A50	CI_APAY_CLR_STG
SCHED_EXTRACT_DT	Date	CI_APAY_CLR_STG. The date that the automatic payment will be downloaded.
APAY_SRC_NAME	A30	CI_APAY_SRC_L
APAY_SRC_DESCR	A60	CI_APAY_SRC_L
TNDR_TYPE_DESCR	A60	CI_TENDER_TYPE_L
TENDER_TYPE_CD	A4	CI_TENDER_TYPE

Copyright © 2026, Oracle and/or its affiliates.

Bill Segment Readings

Field Name	Format	Source/Value/Description
BSEG_ID	A12	CI_BSEG_READ
SP_ID	A10	CI_BSEG_READ
SEQNO	N5	CI_BSEG_READ
REG_CONST	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	S3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
UOM_CD	A30	CI_BSEG_READ
TOU_CD	A30	CI_BSEG_READ
SQI_CD	A30	CI_BSEG_READ
START_REG_READ_ID	A12	CI_BSEG_READ
START_READ_DTTM	DTTM	CI_BSEG_READ
START_REG_READING	S15.6	CI_BSEG_READ
END_REG_READ_ID	A12	CI_BSEG_READ
END_READ_DTTM	DTTM	CI_BSEG_READ
END_REG_READING	S15.6	CI_BSEG_READ
MSR_QTY	S18.6	CI_BSEG_READ
FINAL_UOM_CD	A4	CI_BSEG_READ
FINAL_TOU_CD	A8	CI_BSEG_READ
FINAL_SQI_CD	A30	CI_BSEG_READ
FINAL_REG_QTY	S18.6	CI_BSEG_READ
BADGE_NBR	A30	CI_MTR
NBR_OF_DGTS_RGT	S2	CI_REG

Bill Segment Consumption History

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_BSEG
START_DT	Date	CI_BSEG
BSEG_ID	A12	CI_BSEG
END_DT	Date	CI_BSEG
UOM_CD	A30	CI_BSEG_SQ
TOU_CD	A30	CI_BSEG_SQ
SQI_CD	A30	CI_BSEG_SQ

Copyright © 2026, Oracle and/or its affiliates.

QTY	S18.6	Service Quantity from CI_BSEG_SQ

Payment Information

One record for every payment linked to the bill.

Field Name	Format	Source/Value/Description
PAY_DT	Date	CI_PAY_EVENT
FT_TYPE_FLG	A2	Pay or Pay Cancel ('PS' or 'PX')
CAN_RSN_CD	A4	CI_PAY
CURRENCY_CD	A3	CI_FT.
CUR_AMT	S15.2	Derived from CI_FT. Sum of CUR_AMT for the bill with FT Type Flag = (PS or PX) and Show on Bill Switch = 'Yes'
TOT_AMT	S15.2	Derived from CI_FT. Sum of CUR_AMT for the bill with FT Type Flag = (PS or PX) and Show on Bill Switch = 'Yes'
TOT_PREV_BAL	S15.2	Calculated as Previous Balance = Ending Balance - (current charges + payments + adjustments + corrections)
PAY_CAN_RSN_DESCR	A60	Description of cancel reason code. (Note, that this is retrieved in a special stored procedure that retrieves all the cancel reason codes and descriptions.)

Adjustment Information

One record for every adjustment linked to the bill.

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_FT
ACCOUNTING_DT	Date	CI_FT
ARS_DT	Date	CI_FT
PARENT_ID	A12	CI_FT
SIBLING_ID	A12	CI_FT
FT_ID	A12	CI_FT
CUR_AMT	S15.2	CI_FT
TOT_AMT	S15.2	CI_FT
FT_TYPE_FLG	A2	Adjustment or Adjustment Cancel ('AD' or 'AX')

Copyright © 2026, Oracle and/or its affiliates.

SHOW_ON_BILL_SW	A1	CI_FT
CURRENCY_CD	A3	CI_FT
XFERRED_OUT_SW	A1	CI_FT
ADJ_TYPE_CD	A8	CI_ADJ
CAN_RSN_CD	A4	CI_ADJ
SA_ID	A10	CI_ADJ
CHAR_PREM_ID	A10	CI_SA
DESCR_ON_BILL	A254	CI_ADJ_TYPE_L
ADJ_CAN_RSN_DESCR	A60	Description of cancel reason code. (Note that this is retrieved in a special stored procedure that retrieves all the cancel reason codes and descriptions.)

Bill Message Information

One record for every bill message linked to the bill.

Field Name	Format	Source/Value/Description
MSG_PRIORITY_FLG	A12	CI_BILL_SA or CI_FT
INSERT_CD	S15.2	CI_BILL_SA or CI_FT
MSG_ON_BILL	S15.2	CI_BILL_SA or CI_FT

Billed Revenues by Rate - CI_BILREV

Parameters

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_REV_ACCTY_CHAR	Account Type Char Value for Revenue related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #2. The parameter value indicated for this parameter should be one that represents revenue accounts.

Copyright © 2026, Oracle and/or its affiliates.

Report Description

This is an analysis report for the billed revenues for an accounting period according to the various rates, which were in effect in the system. The information in this report helps to adjust rates in order to achieve better financial results and comply with regulations and market trends.

This report selects all records in the financial transaction GL collection that satisfy the following criteria:

- The financial transaction is frozen.
- The Accounting Date on the financial transaction within input accounting period (parameter 1)
- The distribution code associated with the GL entry has a characteristic type and value that matches the input Account Type Characteristic and Account Type (parameters 2 & 3)

> **Note**
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Case Statistics By Case Type - CI_CSESTS

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used. If end date is not specified, it is defaulted to the current processing date.
Case Condition (Open, Closed)	P_COND_FLG	If specified, only cases in this condition are included in the report. If left blank, the reports produces statistics for both open and closed cases.

Report Description

This report provides two types of statistics:

1. Open cases whose creation date falls between the input Start Date and End Date (inclusive)

Copyright © 2026, Oracle and/or its affiliates.

2. Closed cases whose closing date falls between the input Start Date and End Date (inclusive)

The third parameter is only used if you want to restrict the statistics to only open or closed cases. If you leave this parameter blank, both open and closed statistics will be produced.

The following information is provided in graphical format:

- Number of cases by case type
- Percentage of cases by case type

Case Statistics for a Given Status - CI_CSESGS

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used. If end date is not specified, it is defaulted to the current processing date.
Case Type/Status	P_CASE_STATUS_CD	This is the desired Case Type and Status that will be reported on.
Responsible User	P_CASE_OWNER	If specified, only cases with this responsible user are included in the report.
First Bucket High Limit	P_B1_LIMIT	Cases that took <= this number of days to reach the given status will be grouped together for statistical reporting.
Second Bucket High Limit	P_B2_LIMIT	Cases that took <= this number of days but more than the first bucket high limit to reach the given status will be grouped together for statistics reporting.
Third Bucket High Limit	P_B3_LIMIT	Cases that took <= this number of days but more than the second bucket high limit to reach the given status will be grouped together for statistics reporting. Cases that took more than this number of days are included in another group.

Report Description

Copyright © 2026, Oracle and/or its affiliates.

This report shows cases of a given case type that transitioned to a given status during a given date range.

Graphs are printed to show the number and percentage of cases grouped by the time it took to reach the status. These statistics are grouped into age buckets whose size is controlled by the last 3 parameters.

Summary statistics are also printed showing the minimum, maximum, average and median times for these cases.

Collection Summary - CI_CLLSUM

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	If from date is not set, the default is one month prior to the current date.
End Date	P_TO_DT	End date of the date range. If not defined by user, it is set to one month after the current date

Report Description

This report provides detailed monthly summary information of all collection activities. The report is typically used by a collection department for resource planning and performance review purposes.

This report selects pending and completed collection events whose event trigger date falls between the input start and end dates.

Customer Contacts by Type - CI_CUSTCN

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Start date to use for reporting customer contacts. If not defined, the start date is set to the current date minus 7 days.
End Date	P_TO_DT	End date to use for reporting customer contacts. If not defined, End Date is set to the current date.
Customer Contact Class / Type	P_CC_TYPE_CD	Specify a Customer Contact Class / Type to restrict the report output to a specific class / type.

Report Description

Copyright © 2026, Oracle and/or its affiliates.

This report lists all customer contacts in the system created within the input date range. You may optionally restrict the report to customer contacts for a given Customer Contact Class / Type (parameter 3).

> **Note**
>
> **Graphs.** The information on this report is shown in both textual and graphical formats.
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_CC table on CC_DTTM to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Customers with Life-Support / Sensitive-Load - CI_PMLSSL

Parameters

Parameter	Parameter Code	Description
Service Type	P_SERVICE_TYPE	Service type used to restrict the report to premises with services of this type.

Report Description

This reports displays a detailed list of premises that are coded with Life-Support (LS) or Sensitive-Load (SL) information. It may optionally restrict the output to premises with service points for input service type.

This information is used by a company to make sure that customers at these premises are dealt with appropriately in the case of an outage (planned and unplanned) or service cut due to non-payment.

The report provides detailed information about the premise and the related facility elements providing service to that premise (e.g. substation, feeder and node for electric service).

> **Note**
>
> If ANY person connected to an account has an LS/SL indication in the Person record, then all the premises connected to this account will be treated as LS/SL. This means, for example, that a premise connected to an account that has a 3rd party guarantor with LS/SL is treated as a premise with LS/SL.

GL Accounting Summary - CI_GLACSM

Copyright © 2026, Oracle and/or its affiliates.

Parameters

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type. The account types for the GL accounts are used for grouping the output to the report.

Report Description

This is a financial audit report used to check the financial details in Oracle Utilities Customer Care and Billing for an accounting period against the GL system. The report summarizes all financial transaction (FT) information for a given accounting period according to the different operating and GL divisions and according to various levels of the account GL information.

> **Note**
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Letter Print - BI Publisher Sample Welcome Letters - CI_LTRGN_ENG REPORT

> **Note**
>
> Generating letters in batch is not supported yet.

Parameters

Parameter	Parameter Code	Description
Batch Switch	P_BATCH_SW	For reports accessed online, set the switch to 'No'. Otherwise, it should be set to 'Yes'.

Copyright © 2026, Oracle and/or its affiliates.

Customer Contact	P_CC_ID	References the customer contact associated with the main customer linked to the account.

Report Description

This sample report template for BI Publisher produces letters that are not associated with any other object (i.e., the template does not have to extract information from another object to merge into a letter). You can use this template as a welcome letter for a new customer.

By default, the English versions of the report templates are provided with the base product. If multilingual report templates are required, your implementation should provide reports for each language. When a letter is generated, the system uses the report template based on the customer's language.

The address and name for the company are extracted from the installation options. The text for the letter is defined in the report layout and not provided by Oracle Utilities Customer Care and Billing. Reports are printed according to the customer's language definition and not based on the user's language definition.

This sample report uses the following text:

Welcome to %1. You have been filed with ID Number %2.

We hope to provide you with our best possible service. If you experience any problems or have any questions, please contact one of our customer service representatives at (800)1234567.

%1 is the company name stored as a message on the installation options.

%2 is Person Id stored on the Customer Contact.

Open Cases By Type - CI_CSEOPN

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	See the report's description for how this field is used. If start date is not specified, it is defaulted to 7 days prior to the end date.
End Date	P_TO_DT	See the report's description for how this field is used. If end date is not specified, it is defaulted to the current processing date.
Case Type	P_CASE_TYPE_CD	If specified, only cases of this type are included in the report.
Responsible User	P_CASE_OWNER	If specified, only cases with this responsible user are included in the report.

Copyright © 2026, Oracle and/or its affiliates.

First Bucket High Limit	P_B1_LIMIT	Cases that are open for less than or equal to this number of days will be grouped together for statistical reporting.
Second Bucket High Limit	P_B2_LIMIT	Cases that are open less than or equal to this number of days but more than the first bucket high limit will be grouped together for statistics reporting.
Third Bucket High Limit	P_B3_LIMIT	Cases that are open less than or equal to this number of days but more than the second bucket high limit will be grouped together for statistics reporting. Cases that took more than this number of days are included in another group.

Report Description

This is a report on open cases that were created between a given date range.

The report can be limited to a specific type and/or responsible user.

For each case type, the report shows the following:

- Number of open cases by age bucket (the last 3 parameters control the size (in days) of each bucket)
- Percentage of open cases by age bucket
- Details of the open cases

Payments Balance - CI_PMTBAL

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Report gets all the payments that have been received during a given date range (from and to date parameters). If start date is not defined by the user, it is set to 7 days prior to the current date.
End Date	P_TO_DT	End date of the date range. If not defined by user it is set to the current date

Report Description

This report provides an overall view of all payments created within the input date range. It is typically used for financial control and audit purposes. The report provides summary information about valid payments received and about canceled payment. Data is summarized by the tender source and the type of payment.

Copyright © 2026, Oracle and/or its affiliates.

Receivables Aging - CI_RCVAGA

Parameters

Parameter	Parameter Code	Description
Cutoff Date	P_CUTOFF_DATE	The date from which the arrears buckets are calculated. If no value is entered, the default is the current date minus 7 days.
First Bucket High Limit	P_B1_LIMIT	High limit of first bucket.
Second Bucket High Limit	P_B2_LIMIT	High limit of second bucket.
Third Bucket High Limit	P_B3_LIMIT	High limit of third bucket.

Report Description

This report lists all accounts and their arrears information as of the input cutoff date using a balance forward accounting method.

Outstanding debt is placed into the buckets provided as input using the age of the debt as of the cutoff date. Credits are applied to the oldest debt first. For each account a separate bucket is used to display new charges. In addition, the total accounts receivable balance is displayed for each account.

> **Note**
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ARS_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Tax Payables Analysis - CI_TXPYBL

Parameters

Parameter	Parameter Code	Description
Start Date	P_FROM_DT	Show summary of the tax amounts starting from this date. If not specified, the system will default this value to the current date minus 7 days.

Copyright © 2026, Oracle and/or its affiliates.

End Date	P_TO_DT	Show summary of the tax amounts up to this date. If not specified, the system will default this value to the current date.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_TAX_ACCTY_CHAR	Account Type Char Value for tax related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #3. The parameter value indicated for this parameter should be one that represents tax liability accounts.

Report Description

This report displays a summary of the tax amounts that were levied by the company to customers within the input date range. It also includes the tax exemption information for that period.

This report select all records in the financial transaction GL collection that satisfy the following criteria:

- The financial transaction is frozen.
- The Accounting Date on the financial transaction within the input date range
- The distribution code associated with the GL entry has a characteristic type and value that matches the input Account Type Characteristic and Account Type (parameters 3 & 4)

The report also provides tax exemption information for bill segments whose financial transactions satisfy the above criteria. The tax exemption information is retrieved by looking at the bill calculation lines associated with the FT's bill segment.

> **Note**
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_FT table on ACCOUNTING_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

To Do Entries - CI_TDENTR

Parameters

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Parameter Code	Description
ToDo Entry Status	P_ENTRY_STATUS_FLG	Defines if the To Do entries on the report should be limited to those with a given status value. If this parameter is left blank, the report will show all open and assigned To Do entries.
ToDo Type	P_TD_TYPE_CD	Defines if the To Do entries on the report should be limited to those of a given ToDo type. If this parameter is left blank, the report will show all ToDo type that have at least one open or assigned entry.

Report Description

The report shows open and assigned To Do entries.

You can limit the report to entries in a given status by specifying the desired **To Do Status** (open or assigned). If you don't specify a status, all open and assigned To Do entries will appear on this report.

You can also limit the report to entries of a given To Do Type by specifying the desired To Do Type. If you don't specify a To Do Type, all To Do Types with at least one entry in the open / assigned state will appear on this report.

> **Note**
>
> **Graphs.** The information on this report is shown in both textual and graphical formats.

Umbrella Agreement Summary - CI_UASUMM

Parameters

At least one parameter is required. If the umbrella agreement is entered, no other parameter value is allowed. The report definition uses the validation algorithm RPTV-UASUMM to check these conditions.

Parameter	Parameter Code	Description
Umbrella Agreement	P_UA_ID	Specify the Umbrella Agreement to restrict the report to this umbrella agreement (regardless of its status). When this parameter is specified, all other parameters are not allowed.

Copyright © 2026, Oracle and/or its affiliates.

Account Management Group	P_AMG	Specify a value for this parameter to restrict the report to umbrella agreements related to this Account Management Group.
Umbrella Agreement Characteristic Type	P_CHAR_TYPE	Specify a value for this parameter to restrict the report to completed umbrella agreements related to this Characteristic type and value (parameter 4).
Umbrella Agreement Characteristic Value	P_CHAR_VALUE	Specify a value for this parameter to restrict the report to completed umbrella agreements related to this Characteristic type (parameter 3) and value.
Number of Days Before Expiration	P_EXPIRE_IN_X_DAYS	Specify a value for this parameter to restrict the report to completed umbrella agreements whose End Date is on or before the current date less the Number of Days before Expiration.

Report Description

This report provides summary information about one or more umbrella agreements.

If the user does not supply an umbrella agreement ID, but enters one or more of the other parameters, the report only selects umbrella agreements that match the input criteria AND the match the following criteria:

- The status of the umbrella agreement is complete
- The start and end dates of the umbrella agreement include the current date

The report does not include canceled terms of service records or canceled service agreements linked to the above umbrella agreements.

For each umbrella agreement that is included in the report, the output includes information about the umbrella agreement, its collection of terms of service records and for each TOS record, its collection of service agreements.

In addition, the report includes the following graphs:

- Umbrella agreement summary graph showing billing history by accounting period for the effective dates of the umbrella agreement for all service agreements linked to the umbrella agreement's terms of service records.
- Terms of service summary graph showing billing history by accounting period for the effective dates of the umbrella agreement for all its service agreements

Unbilled Revenues - CI_UBLREV

Parameters

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Parameter Code	Description
Accounting Period	P_ACCT_PERIOD	Defines the accounting period used for the report. A valid fiscal year and accounting period for a valid accounting calendar must be provided.
Account Type Characteristic	P_CHAR_TYPE	Defines a Characteristic Type for a characteristic linked to the Distribution Code to define an Account Type.
Account Type	P_REV_ACCTY_CHAR	Account Type Char Value for Revenue related GL Accounts. The char type defined for this parameter should match the Char Type code defined as parameter #2. The parameter value indicated for this parameter should be one that represents revenue accounts.

Report Description

This report provides a simplified calculation of estimated unbilled revenue for a given month.

It processes frozen financial transactions associated with bill segments and canceled bill segments whose bill segment end date is within the **Accounting Period** (parameter 1). It determines the billed revenue for the FT and then uses this information to calculate the unbilled revenue.

To determine "revenue", the report summarizes amounts posted to any distribution code with a characteristic entry that matches the input **Account Type Characteristic** type and **Account Type** characteristic value (parameters 2 and 3).

The estimate for the unbilled portion is calculated as (Bill Amount / Number of Days in the Bill Period) * Number of Unbilled Days for the Accounting Period.

For example, consider a bill segment on a monthly-billed cycle for the period of 3/10/03 - 4/9/03, and for $150 (revenue part of the bill). Assume we are in the April/03 accounting period, which covers 4/1/03 through 4/30/03. This means 21 days are unbilled for April. The unbilled revenue is calculated as 150/31*21 = $101.61.

The following are some points to note about this report.

- If the report runs for an historical date, it still estimates the unbilled revenue portion based on the above formula even if actual data exists. In other words, it will not try to find actual bills for subsequent months. As a result, this report always shows what would have been the estimated unbilled revenue for a particular month if that month is the most recent one.
- Services can be billed monthly bi-monthly, quarterly or in any desired frequency. This report only performs the revenue recognition estimation using actual bills that were created in the report's accounting period. As a result, it does not estimate unbilled revenue for accounts that were not billed in any portion of the input accounting period.
- Revenue recognition practices are unique and may vary from customer to customer. In a given month we can produce current bills (e.g. for electric service) or future bills (e.g. cable services). In addition in some cases we can have bills that started before the accounting period and ending after it (for example in a cancel-rebill situation). This report will only

Copyright © 2026, Oracle and/or its affiliates.

estimate unbilled revenues for bills with the accounting period equals to the report's accounting period.

> **Note**
>
> **Performance Consideration.** If your implementation chooses to use this report, you may consider adding an index to the CI_BSEG table on START_DT, END_DT to aid in performance. When making this decision, carefully weigh the benefit of improving report performance against the possible degradation to the performance of day-to-day processing as a result of defining a new index. In addition, many companies opt to create a reporting database that is a shadow of production to ensure that indexes defined to benefit reports may be created without any affect on the production environment.

Defining Overdue Processing Options

The system periodically monitors how much your customers owe to ensure they haven't violated your collection rules. When a violation is detected, the system initiates the appropriate activities (e.g., letters, disconnect notices, collection agency referrals, and eventually write off). The topics in this section describe how to configure the system to manage your overdue processing requirements.

> **Note**
>
> The overdue processing module has been designed to collect on virtually anything from an unpaid bill to an unmatched financial transaction. You tell the system what you collect on by configuring the various overdue processing control tables. In this release, the base-package is delivered with algorithms that support collecting on overdue bills. If your organization practices balance-forward accounting (i.e., collection is based on overdue service agreement balances), you will not use this functionality. Rather, you will use the functionality described under Defining Credit & Collections Options.

> **Warning**
>
> Setting up this module is as challenging as your organization's collection rules. If you have simple rules, the set up process will be straightforward. If your rules are complicated (e.g., they differ based on the type of customer, the age of debt, the type of service, etc.), your setup process will be more challenging.

- Case Study - Collecting On Overdue Bills
- How Does The Overdue Monitor Work?
- The Big Picture Of Overdue Processes
- Bill-Oriented Collection - Advanced Topics

Copyright © 2026, Oracle and/or its affiliates.

- Creating Overdue and Cut Procedures

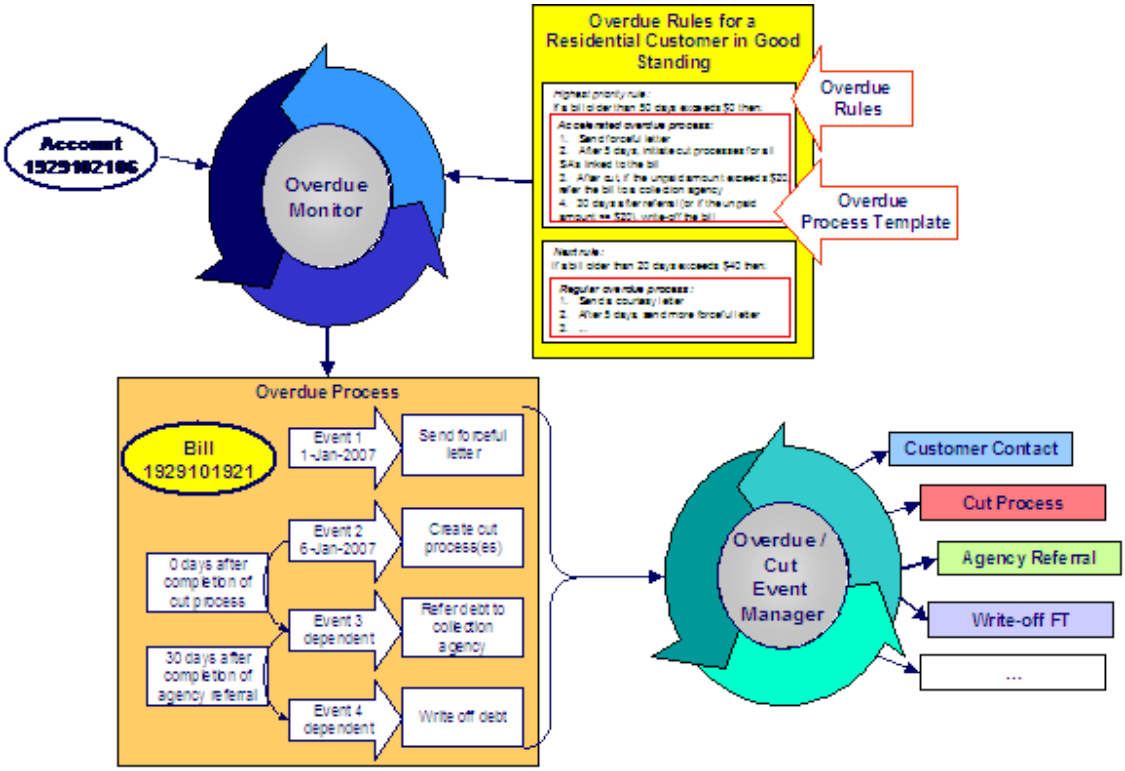
Case Study - Collecting On Overdue Bills

The following topics introduce a case study that describes how overdue processing can be used to collect on overdue bills. This is just an example as virtually every aspect of overdue processing is configurable. Use this case study to familiarize yourself with the overdue processing core concepts.

- Monitoring Overdue Bills
- Cutting Service Agreements

Monitoring Overdue Bills

The following diagram illustrates the objects and processes involved with collecting overdue bills.



There are many important concepts illustrated above:

Copyright © 2026, Oracle and/or its affiliates.

The Overdue Monitor checks if your accounts have bills that violate your overdue rules	The Overdue Monitor is a background process that periodically reviews your account's financial obligations. Note well: every account belongs to a collection class. There are two types of collection classes: those whose accounts are monitored by the Account Debt Monitor, and those that are monitored by the Overdue Monitor. This chapter describes the Overdue Monitor.
Overdue rules define when and how unpaid bills are collected	An account's collection class overdue rules have algorithms that monitor an account's financial obligations. These algorithms are invoked by the Overdue Monitor when it's time to review an account's obligations. These algorithms can contain any type of criteria. However, most are defined using a combination of a threshold age and monetary amount. For example, a classic algorithm would check if a bill has unpaid financial transactions more than 20 days old that exceed $50. In the case of bill-oriented collection, the monitoring algorithms look at each of the account's bills to determine if they are paid. Note, a bill is considered paid if its financial transactions (FTs) are linked to a balanced match event. If a monitoring algorithm finds an unpaid bill, it can check if it old enough (and large enough) to be considered a violation. When you set up a monitoring algorithm, you define the type of overdue process that should be created when an overdue bill is detected. You do this by defining the appropriate "overdue process template".
An overdue process template defines how to handle an overdue bill	An overdue process template contains one or more overdue event types. These define the number and type of events that are created to prod the customer to pay. For example, you might set up an overdue process template with event types to send a series of letters followed up by a call. The overdue process template has contains the rules defining when events are activated. The specific action that's performed by an overdue event is controlled by the Activation algorithm defined on its event type. Refer to Overdue Event Type - Main for a list of the various Activation algorithms delivered with the base package.

Copyright © 2026, Oracle and/or its affiliates.

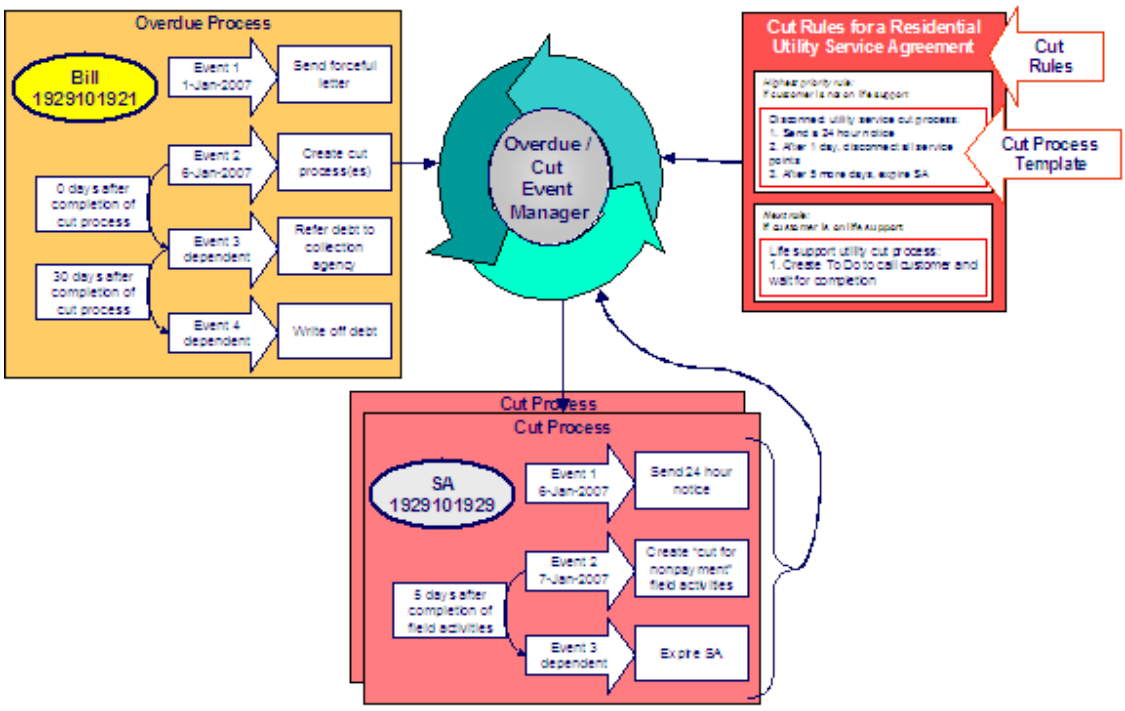
Multiple objects can be associated with a single process	The above diagram shows a single bill linked to an overdue process. It should be noted that an overdue process is capable of referencing multiple bills (or other objects). Note well: while a single overdue process can reference many overdue objects, all such objects must be of the same type. For example, you cannot commingle bills and service agreements under a single overdue process. The type of object managed by an overdue process is defined on its overdue process template.
If a customer pays the bill, the overdue process is cancelled	If an overdue bill is paid, the overdue process is canceled real-time. You control if and how an overdue process is cancelled by setting up the appropriate rules on the overdue process template.
The Overdue / Cut Event Manager activates and triggers overdue events	The Overdue / Cut Event Manager is a background process that activates overdue events on the appropriate date. On this date, the event's Activation algorithm(s) are called. This Overdue / Cut Event Manager also has the responsibility of recursively activating later events that are dependent on the completion of earlier events.
Events can be activated real-time	Overdue Process - Main has a button that allows users to activate (and recursively trigger) overdue events online / real-time. This means you don't have to wait for a batch job to activate events.
Overdue events can wait for related activities to complete	As described above, an overdue event's Activation algorithm can create virtually any object. What wasn't explained is that the event can be set up to wait for the ancillary object to finish before it completes. For example, an event can create a To Do entry and wait for it to complete before the next event is triggered. You can introduce plug-ins to create and wait on virtually any object. While an overdue event is in the Wait state, the Overdue / Cut Event Manager monitors the state of the related object(s). When the related object completes, the event is transitioned to the Complete sate (thus triggering dependent overdue events). Please see Some Events Wait For Something Before Completing for more information.

Cutting Service Agreements

An overdue process may contain an overdue event that creates a cut process(es). A cut process is used to cut (i.e., stop) a service agreement. The following diagram illustrates the objects and processes involved with cutting a service agreement.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Cut processes aren't required.** It's quite conceivable to design an overdue process that doesn't cut service. For example, the overdue process may just contain an event that creates a case and the case manages the collection activities.



An overdue process may contain an overdue event that cuts (i.e., stops) service	If an overdue process's bill remains unpaid, one of the latter overdue events typically creates one or more cut processes. A cut process contains the activities to stop a service agreement (in the hopes that lack of service will inspire the customer to pay). It should be noted that a separate cut process is created for each service agreement.
Cut rules define how to sever service agreements	The system allows you to define rules to control the type of cut process created. For example, you may have a different cut process if the customer has life support equipment, or if it's winter, or ... The cut rules are defined on the service agreement's SA type. This allows different rules for different types of service agreements.

Copyright © 2026, Oracle and/or its affiliates.

Overdue events can "wait" for related activities to complete	When an overdue event creates cut process(es), it typically enters the Wait state. It enters this state as it is waiting for the cut process(es) to complete before it can itself complete. While in the Wait state, the Overdue / Cut Event Manager monitors the state of the related cut process(es). When the cut process(es) complete, the system transitions the originating overdue event to the Complete state (thus triggering dependent events).
A cut process template defines how to cut a service agreement	Cut processes and events are created using a cut process template. A cut process template defines the actions involved in cutting a given type of service agreement. A cut process template usually contains several cut events. These events are a series of letters and / or disconnection field activities that eventually result in the expiration of a service agreement if payment is not received. An Activation algorithm controls the specific action associated with a cut event and therefore an event can do practically anything. Refer to Cut Event Type - Main for a list of the various Activation algorithms delivered with the base package.
Cut events are also managed by the Overdue / Cut Event Manager	The same background process that manages the activation of overdue events manages cut events. This means you only have to submit one batch job to activate and trigger both overdue and cut events.
Events can be activated real-time	Cut Process - Main has a button that allows users to activate (and recursively trigger) cut events online / real-time. This means you don't have to wait for a batch job to activate events.
Cut events can "wait" just like overdue events	Cut events support the ability to wait for something to complete just like overdue events. For example, if a cut event creates a field activity, it enters the Wait state. While in the Wait state, the Overdue / Cut Event Manager monitors the state of the related field activities. When the field activities complete, the system transitions the originating cut event to the Complete state. And, just like for overdue events, the notion of a cut event creating something and then waiting for it to complete is not limited to field activities. For example, you could have a cut event create a To Do entry and wait for it to complete before the next event is triggered. Please see Some Events Wait For Something Before Completing for more information.

Copyright © 2026, Oracle and/or its affiliates.

After a service agreement is stopped, it will be final billed	Many cut processes are configured so their last cut event expires the service agreement. What happens at expiration depends on the type of service agreement. However, eventually the service agreement is transitioned into the Stopped state. When the last active service agreement linked to an account is stopped, the system changes the account's bill cycle to bill that evening. If only one of many SAs is stopped, the SA will only be final billed as per the account's original bill cycle schedule.
If a customer doesn't pay their final bill, the final bill will be processed using a different type of overdue process	Final bills only differ from ongoing bills in that the service agreements associated with the bill's FTs are not active. This means that there is nothing to cut. This means that the type of overdue process used to handle a "final" bill should differ from that used to handle "ongoing" bills. As described above, overdue rules define how unpaid bills are handled (both ongoing and final). We recommend configuring your overdue rules to use different overdue process templates for final versus ongoing bills.
The last overdue event typically causes the debt to be written off	Ultimately, if the overdue and cut events fail to inspire payment, the debt will be written off. The method used to write-off a bill is controlled by an overdue event Activation algorithm.

How Does The Overdue Monitor Work?

This section describes how the Overdue Monitor background process C1-ADMOV) uses your overdue rules to collect overdue debt.

This process has an optional input parameter for CIS Division. It can be used if you implementation requires processing of data by CIS Division. You will need to create a copy of this process's batch control for each CIS Division.

> **Note**
>
> **Recommendation.** We recommend that you familiarize yourself with the concepts described in the case studies before reading this section.
>
> **Collecting overdue bills.** The examples in the following section frequently refer to how the Overdue Monitor is configured for an organization that collects on unpaid bills. It should be noted that these are just examples as the Overdue Monitor can be used to collect on virtually anything (if you create the appropriate plug-in algorithms).

- Different Overdue Rules For Different Customers
- Overdue Rules Are Embodied In Algorithms
- When Is An Account Monitored?

Copyright © 2026, Oracle and/or its affiliates.

- Collection Class Defines If And How Accounts Are Monitored

Different Overdue Rules For Different Customers

The Overdue Monitor uses rules to control how it monitors an account's debt. The system allows you to define different rules for different combinations of collection class, division and currency code. For example,

- You probably have different collection rules for different jurisdictions (i.e., CIS Divisions). For example, if you have customers in different states / provinces, you may have different rules applied to each jurisdiction's debt. The **CIS division on each customer's account** defines the jurisdictional rules applied to the account's debt.
- You probably have different collection rules for different customer segments. For example, bills for large customers might be overdue if they are more than 10 days late, whereas small customers might have 24 days. You differentiate your customers in respect of the overdue via the **collection class on the customers' accounts**. An account's initial collection class is defaulted from its customer class. You may override an account's collection class at will.
- You will have different criteria for every currency in which you work because the monitoring process always compares a customer's debt against some value and this value must be denominated in the customer's currency. A customer's currency is defined using a **currency code on the account**.

The above means that every combination of CIS division, collection class, and currency code can have different rules. The following matrix is used to illustrate a sample organization's rules (note, we assume a single currency and therefore avoid the third dimension):

Account's Division	Account's Collection Class: Commercial Customer	Account's Collection Class: Residential Customer
North	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 45 days, create the commercial 45 days late overdue process. Next Priority: If a bill exists with unpaid FT's > $100 that is older than 30 days, create the commercial 30 days late overdue process.	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 50 days, create the accelerated overdue process for residential customers. Lower Priority: If a bill exists with unpaid FT's > $25 that is older than 25 days, create the courtesy reminder overdue process for residential customers.
South	...	...

Notice that there can be multiple criteria for each cell in the matrix. What differentiates one criteria from another is its priority. The higher priority criteria will be compared first. If the debt violates the criteria, the overdue process is initiated and no further comparisons are performed.

Overdue Rules Are Embodied In Algorithms

Your organization's overdue rules are defined in algorithms plugged in on Collection Class Overdue Rules (in the Overdue Monitor Rule system event). When the Overdue Monitor analyzes an account, it uses the rules associated with the account's collection class, CIS

Copyright © 2026, Oracle and/or its affiliates.

division and currency code. To analyze an account, it simply invokes these algorithms in sequence order, i.e., the lower the sequence, the higher its priority.

An Overdue Monitor Rule algorithm has two responsibilities:

- it determines if an account violates its overdue rules,
- if so, it creates one or more overdue processes using an overdue process template

When Is An Account Monitored?

The Overdue Monitor determines if an account violates your overdue rules at least every X days, where X is defined on the Customer Class - Controls associated with the account's customer class and division (in the field Min Credit Review Freq (Days)).

In addition, the Overdue Monitor examines an account's debt as follows:

- X days after an account's bill due date (X is defined in the field Credit Review Grace Days on customer class control).
- If a payment is canceled with a cancellation reason that indicates non-sufficient funds.
- If a payment arrangement is broken (assuming you use the base package's break payment arrangement plug-in). Refer to Breaking A Bill Oriented Payment Arrangement for more information.
- If a pay plan is broken. Refer to The Pay Plan Monitor for more information.
- If a match event is added, changed or deleted.
- When a written off bill is reversed.

> **Note**
>
> **Additional events.** Your implementation can have other events trigger the analysis of an account by the Overdue Monitor. To do this, add logic to insert a row on the CI_ADM_RVW_SCH table when the event occurs. This row simply references the account ID to be reviewed and the desired review date.

Collection Class Defines If And How Accounts Are Monitored

As described above, every account references a collection class. The collection class defines if and how its accounts are monitored. There are three options:

- The accounts are monitored by the Account Debt Monitor
- The accounts are monitored by the Overdue Monitor (this is described in this chapter).
- The accounts are not monitored for overdue debt.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Migration.** If you plan to migrate from the Account Debt Monitor (ADM) method of collection to the Overdue Monitor method, a special Overdue Monitor algorithm (C1-ACT-CSW) has been supplied that will skip accounts that are eligible for review by the Overdue Monitor if they have an active collection, severance or write-off process. This algorithm should be plugged in the applicable Collection Class Overdue Rules so that it will be invoked first. This allows accounts with active ADM-oriented collection activities to work themselves through the system. When an account no longer has any active ADM-oriented activities, monitoring responsibilities will be assumed by the Overdue Monitor.

The Big Picture Of Overdue Processes

As described above, the Overdue Monitor subjects your accounts to overdue rules. If a rule is violated, an overdue process is created. The topics in this section provide background information about overdue processes.

- How Are Overdue Processes Created?
- The Components Of An Overdue Process
- Experimenting With Alternative Overdue Process Templates
- Overdue Process Information Is Overridable
- Original and Unpaid Amounts
- Overdue Processes Are Highlighted Elsewhere
- How Are Overdue Processes Cancelled?
- Overdue Processes Are Created From Templates
- The Big Picture Of Overdue Events
- The Big Picture Of Cut Processes
- Cut Events Are Like Overdue Events
- Write Offs Are Implemented Using Overdue Events
- Calendar vs Work Days

How Are Overdue Processes Created?

As described above, the system creates an overdue process when an account violates your overdue rules. In addition, a user can manually create an overdue process at their discretion.

The Components Of An Overdue Process

The following topics describe the major components of an overdue process.

- Overdue Objects

Copyright © 2026, Oracle and/or its affiliates.

- Overdue Events
- Overdue Log

Overdue Objects

When an overdue process is created, the system links the overdue object(s) to the process. For example, if an overdue bill is detected, the bill is linked to the overdue process.

When you set up an overdue process template, you define the type of object it collects on by defining the foreign key characteristic type used to reference the object. For example, when you set up an overdue process template to collect on bills, you define a foreign key characteristic type that references the bill object.

Overdue Events

An overdue process has one or more overdue events. These events are the actions designed to encourage the customer to pay. For example, you might set up overdue events that:

- Send letters (via the creation of a customer contact)
- Create To Do entries
- Impact the account's credit rating
- Create a case (e.g., to seize the customer's assets)
- ... (the list is only limited by your imagination as algorithms are used to perform the event's actions)

You define the number and type of events by configuring overdue process templates. When the system creates an overdue process, it copies the events defined on the specified template.

It's important to note that all overdue events are created when the overdue process is created. A separate background process, the Overdue / Cut Event Manager, is responsible for activating, monitoring, and triggering overdue events. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, start a cut process, refer debt to a collection agency, write-off debt, etc.).

Overdue Log

Every overdue process has a log holding its history. Entries are added to the log when meaningful events occur, for example:

- When the process is created, a log entry is created to describe why the process was started.
- When an overdue event is activated, a log entry is created. These entries frequently contain a foreign key to the object that the event created so that users can easily drill down to the object from the log. For example, if an event creates a To Do entry, the To Do entry's foreign key is placed on the log and this allows a user to drill down on the log entry to see the To Do entry.
- When a process is canceled, a log entry is created to describe the circumstances of the cancellation (e.g., manual versus automated).

Copyright © 2026, Oracle and/or its affiliates.

- Users can manually add log entries (you might want to think of these as "diary" entries) as desired.
- ...

Many of the base-package algorithms involved in overdue processing insert log entries so that a thorough audit trail is maintained. These algorithms have been designed to allow you to control the verbiage in each log entry by defining the desired message number using an algorithm parameter.

The log is viewable on the Overdue Process - Log page.

> **Note**
>
> **More than just an audit trail.** The log entries are more than just an audit trail. The system makes use of the log entries to know what it did. For example, when an overdue event needs to monitor the state of the To Do entries that it created, it uses the log to determine the identity of these To Do entries.

Experimenting With Alternative Overdue Process Templates

The system allows you to determine the efficacy of proposed overdue process templates using a small subset of customers before implementing the templates on the entire customer base. We use the term "champion / challenger" to reference this functionality.

We'll use an example to explain. Let's assume your prevailing overdue process template for residential customers starts with a "gentle reminder" letter followed 10 days later by a letter threatening collection agency referral if payment is not received. You may want to experiment with the impact of a change to this template. For example, you may want to change the "gentle reminder" to something more assertive and follow this up 5 days later with an even sterner warning. You can use the "champion / challenger" functionality to perform this experiment.

The following points describe how to implement "champion / challenger" functionality:

- Set up a "challenger" overdue process template for each template that you want to experiment with.
- Insert a new **Champion/Challenger** option on the Overdue Processing Feature Configuration for every champion template. Each option's value defines:
  - the "champion" overdue process template code
  - the "challenger" overdue process template code
  - the percentage of the time the system should use the "challenger" template
- Keep in mind that you can only experiment with one challenger template per champion template. For example, let's assume you have two prevailing overdue process templates - one for residential customers and another for commercial customers. You can experiment with different challenger templates for the residential and commercial templates. However, you cannot experiment with two different challenger templates for the residential champion template (i.e., a champion template can have 0 or 1 challenger template).

After setting up the above, your implementation's Overdue Rule Plug-In may include logic to use the challenger template X% of the time rather than the champion template. The sample plug-in provided in the base product, called C1-CB-CR-RAT, includes this logic.

Copyright © 2026, Oracle and/or its affiliates.

If you are using the Oracle Utilities Analytics product, you can configure analytic zones in innumerable ways to compare the efficacy of the champion versus the challenger. For example,

- You can set up a graph to show the average duration of each type of process.
- You can set up a graph to show the average dollars that were successfully collected.
- You can set up a dimensional scorecard to show how each template performed in different regions (or customer classes or ...).
- Etc (the list is limited by your imagination)

Overdue Process Information Is Overridable

"Overdue process info" is the concatenated string of information that summarizes an overdue process throughout the user interface. The base-package logic constructs this string by concatenating the following information:

- The description of its overdue process template
- Its status
- For active processes, the number of days since it was created. For inactive processes, the number of days since it was inactivated.
- For active processes, the unpaid amount of the objects being collected

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your overdue process templates. This design allows some / all overdue process templates to have an override info string.

Original and Unpaid Amounts

There are two amounts associated with each overdue object linked to an overdue process: its Original Amount and its Unpaid Amount. These amounts are used throughout overdue processing.

You control how these amounts are calculated by defining the appropriate algorithm on your overdue process templates. For example, you can plug in a base-package algorithm (C1-CUAOA) if you collect on overdue bills and the original amount is the amount of financial transactions linked to the bill, and the unpaid amount is the sum of financial transactions that are not linked to balanced match events.

> **Note**
>
> The system captures a history of an overdue process's unpaid amounts. Refer to How to Capture Arrears History for more information.

Overdue Processes Are Highlighted Elsewhere

The topics in this section describe how other parts of the system highlight the existence of overdue and cut processes.

Copyright © 2026, Oracle and/or its affiliates.

- Alert Zone
- Credit and Collections Zone
- Account Activity History Zone

Alert Zone

If you plug-in the appropriate alert algorithm (C1-OD-PROC) on the Installation record, alert(s) will be shown for active overdue processes in the Alert Zone that appears in the Sidebar and on Control Central - Account Information.

Credit and Collections Zone

The Credit and Collections zone on Control Central - Account Information shows active overdue and cut processes.

Account Activity History Zone

The Account Activity History Zone on Control Central - Account Information shows pending and waiting events and inactive processes.

How Are Overdue Processes Cancelled?

A user may cancel an overdue process at their discretion, online / real-time using Overdue Process - Main.

The system will automatically cancel an overdue process when the object(s) associated with the overdue process are sufficiently paid. Exactly when the system checks if an overdue process should be cancelled is dependent on your organization's billing and accounting rules. For example, if you practice open-item accounting, you'd want to analyze an account's active overdue processes whenever a match event is added, changed or deleted (as match events are the only objects that impact if debt is considered paid in an open-item world). The base-package supports this specific example. If you need additional events to check if an overdue process should be canceled (e.g., the creation of a pay plan), a base-package change MAY be necessary. Please check with customer support if you have questions.

Two algorithms plugged-in on the overdue process template handle the cancellation:

- The Cancel Criteria algorithm is responsible for determining if an overdue process should be canceled. Algorithms of this type analyze the outstanding debt on the objects linked to the overdue process and indicate whether a process can be cancelled.
- The Cancel Logic algorithm is responsible for actually canceling the process. The logic involved in cancellation can be quite sophisticated as canceling an overdue process can result in the cancellation of its pending events and cut processes.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Why two algorithms?** The reason two algorithms are involved in cancellation is because we want the cancellation logic to be encapsulated in one place so it can be called during both manual and automated cancellation.
>
> **Different logic for different templates.** Because both the Cancel Criteria and Cancel Logic algorithms are plugged-in on the overdue process's template, you can have different cancellation criteria and logic for different templates.

Overdue Processes Are Created From Templates

As described above, you set up overdue process templates to define the types of events and when they are executed. When an overdue process is created, its events are created by copying the event types from an overdue process template. The remaining topics in this section provide background information to assist you in setting up your templates.

The Big Picture Of Overdue Events

This section describes the various types of overdue events and their lifecycle.

- How Are Overdue Events Created?
- Overdue Events Can Do Many Things
- Overdue Event Information Is Overridable
- Overdue Event Lifecycle

How Are Overdue Events Created?

Overdue events are created as follows:

- The Overdue Monitor invokes Overdue Monitor Rules to periodically check your accounts (refer to Overdue Rules Are Embodied In Algorithms for how this works). An Overdue Monitor Rule creates an overdue process when an account violates your overdue rules. The overdue process has one or more overdue event(s). The number and type of events is controlled by the overdue process template specified on the Overdue Monitor Rule.
- Users can create an overdue process manually on Overdue Process - Main. To do this, they specify an overdue process template. The number and type of overdue events is defaulted from the template.
- An overdue event may be manually added to an existing overdue process by a user on Overdue Process - Events.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bottom line.** Most overdue events are created by the system when it creates an overdue process for delinquent obligations. If you need to create an ad hoc overdue event, you can either create an overdue process whose template contains the desired event OR link the desired event to an existing process.

Overdue Events Can Do Many Things

An overdue event can perform a wide number of activities as the logic is embodied in an algorithm. The following points describe how this works:

- Every overdue event references an overdue event type.
- The overdue event type, in turn, references an Event Activation algorithm.
- The Event Activation algorithm is invoked when the event is triggered.

Overdue Event Information Is Overridable

"Overdue event info" is the concatenated string of information that summarizes an overdue event throughout the system. The base-package logic constructs this string by concatenating the following information:

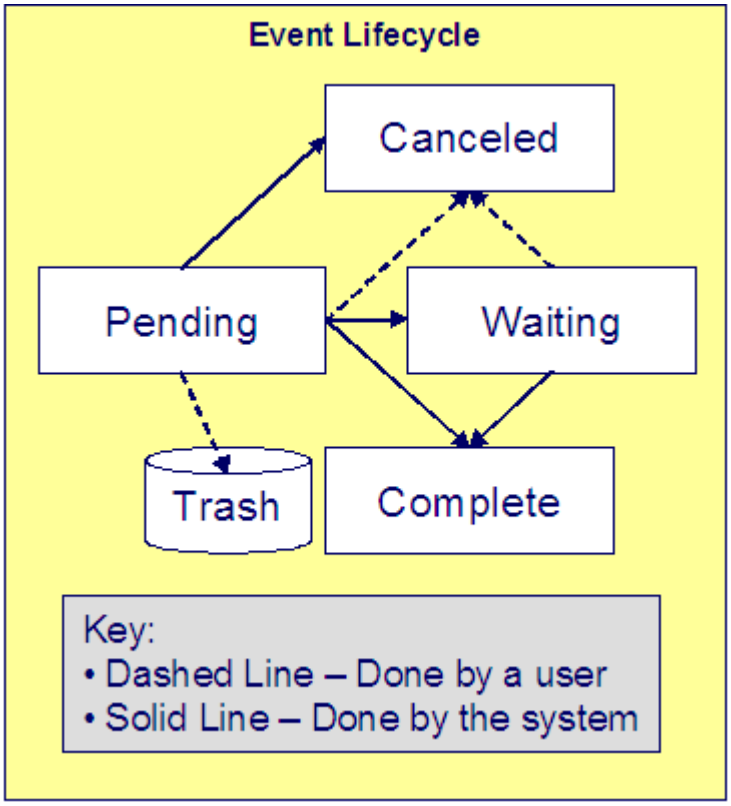
- The event type's description
- The event's status
- If it's pending:
  - If the event has a trigger date, the number of days until it's triggered plus the verbiage day(s) from today
  - Otherwise, the verbiage dependent on other events
- If it's waiting, the number of days, hours and minutes that it's been waiting
- If it's canceled, the cancel reason code's description
- If it's complete, the number of days, hours and minutes that it's been complete

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your event types. This design allows some / all event types to have an override info string.

Overdue Event Lifecycle

The following diagram shows the possible lifecycle of an overdue event:

Copyright © 2026, Oracle and/or its affiliates.



Overdue events are initially created in the Pending state. An event can take myriad paths after it's created; it all depends on how you've configured the system. The following topics describe an event's lifecycle:

- How and When Events Are Activated
- Activating Events Should Add a Log Entry
- Holding Events
- Some Events Wait For Something Before They Activate
- How Are Events Canceled

How and When Events Are Activated

An overdue event contains the date is should be activated; this is referred to as its trigger date. On this date, the Overdue / Cut Event Manager (a background process (C1-ODET)) invokes the Event Activation algorithm plugged-in on the event's event type. The Event Activation algorithm, in turn, will decide on the state in which to leave the overdue event (e.g., it may transition it to the Complete state or the Waiting state).

If a user can't wait for the Overdue / Cut Event Manager real-time, they can click a button on Overdue Process - Main to activate (and recursively trigger) events online / real-time.

You control how an event's trigger date is populated by configuring the overdue process template. You are given two choices when you link an event type to an overdue process template:

- You can indicate the event should be assigned a trigger date when it is first created. You'd use this approach on the first event and events with no dependencies on earlier events. The following points describe how to configure the overdue process template to do this:

Copyright © 2026, Oracle and/or its affiliates.

- Indicate the event type is NOT dependent on other events, and
- Define the number of days after the process's creation to use when calculating the trigger date.

- You can indicate the event should be assigned a trigger date only after earlier events are Complete. This technique should be used whenever you have an event that is only executed after other events are Complete. The following points describe how to configure the overdue process template to do this:
  - Indicate the event is dependent on other events, and
  - Define the number of days after the completion / cancellation of all dependent event(s) that the trigger date should be set to. The Overdue / Cut Event Manager sets the trigger date on such an event when it detects that all of its dependent events are complete / canceled.

> **Note**
>
> **Calendar vs Workdays.** When an overdue event is created by the system, its trigger date is set in accordance with your date arithmetic preferences. Refer to Calendar vs. Work Days for more information.

Activating Events Should Add a Log Entry

As described above, an overdue process has a log holding a history of meaningful events in the process's life. Most Event Activation algorithms will add an entry to the process's log.

These log entries are more than just an audit trail as they also reference the objects that are created during activation. For example, if an activation algorithm creates a customer contact, the ID of the customer contact will be referenced on the log (and end-users will be able to drill down on the log entry to see the customer contact).

Holding Events

You can prevent a pending event with a trigger date on / before the current date from activating by plugging-in a Hold Event Activation plug-in on the overdue process template. This might prove useful, for example, if you want to suspend an overdue process while a case used to investigate the assertion of a customer is outstanding. Then, when the case closes, the overdue process can start up where it left off.

This technique would prove useful, for example, if your users can grant ad hoc suspend periods (e.g., if a customer wants a few extra days to pay before a cutoff). To do this, create a case type that has two states: Open and Close. Make sure to set up the Open state to have an automatic transition algorithm to close the case after X days. Then, all a user has to do is create a case of this type when they want to provide a suspend period. When the suspend period is over and payment isn't received, the case will close and the overdue process will start up where it left off.

Some Events Wait For Something Before They Activate

Copyright © 2026, Oracle and/or its affiliates.

Consider this scenario - you want an overdue event to create a To Do entry so a user can authorize the next phase of an overdue process. When this event activates, the event's activation algorithm will create a To Do entry, but it will NOT transition the event to complete. Rather, the overdue event will exist in the waiting state. While in the waiting state, the Overdue / Cut Event Manager will monitor the state of the To Do entry. When the To Do entry completes, the original overdue event can transition to the complete state and then latter dependent events can be triggered. The following points describe how to configure the system to support this type of event:

- The event type's Event Activation algorithm should behave as follows:
  - It creates the object on which the overdue event waits.
  - It must link this object to the overdue process by creating a log entry where the prime-key of the related object is referenced (in a foreign-key characteristic). This log entry should also reference the event.
  - It should leave the overdue event in the waiting state.
- The event type must have a Monitor Waiting Event algorithm. This algorithm is invoked each time the Overdue / Cut Event Manager executes. If the related object has transitioned to a "final" state, the originating overdue event is transitioned to the complete state (and then latter dependent events are triggered).

> **Note**
>
> **Bottom line.** Two algorithms must be set up on an overdue event type to implement waiting functionality: an Event Activation algorithm that creates the monitored object and a Monitor Waiting Event algorithm to check on the state of the monitored object. The Overdue / Cut Event Manager has the dual responsibility of activating the event and monitoring its related object for completion (and then triggering the dependent events when it completes).

While the above example illustrated how an overdue event could create and then monitor a To Do entry, you can use this functionality to create and monitor any object that has an initial and final state. If the base package does not contain the algorithms you need, simply develop new ones using the base-package algorithms as examples.

How Are Events Canceled

A pending event will be cancelled automatically by the system when the overdue process is canceled. Refer to How Are Overdue Processes Cancelled for more information.

A user may cancel a pending or waiting event at their discretion.

Regardless of what triggers the cancellation, the Cancel Logic algorithm plugged in on the overdue event type handles the cancellation. This allows you to introduce additional cancellation logic should the need arise. Please note that the base package cancel algorithms insert a log entry when a user manually cancels an event.

The Big Picture Of Cut Processes

Copyright © 2026, Oracle and/or its affiliates.

While not required, many overdue processes will contain one or more events that will cut (i.e., stop) service. When such an event activates, it creates a "cut process". A cut process is very similar to an overdue process; the major difference is that a cut process is linked to a specific service agreement (the one being cut), whereas an overdue process is linked to the object(s) in arrears.

The topics in this section provide background information about cut processes.

- How Are Cut Processes Created?
- Overdue Events Wait For The Cut Process To Conclude
- Cut Processes Exist Under An Overdue Process
- The Components Of A Cut Process
- Cut Process Information Is Overridable
- How Are Cut Processes Cancelled?
- Cut Processes and Events Are Created From Templates

How Are Cut Processes Created?

The activation of an overdue event can create one or more cut processes.

In addition, a user can manually create a cut process at their discretion using Cut Process - Main.

Overdue Events Wait For The Cut Process To Conclude

Overdue events that create cut processes are perfect examples of events that wait for the object they create to complete before they, in turn, complete. After the cut process concludes, the originating overdue event will complete thus triggering its dependent events.

Cut Processes Exist Under An Overdue Process

It's important to note that a cut process exists in respect of an overdue process. In other words, you can't create a cut process without referencing a parent overdue process. There are two reasons why:

- A cut process's overdue process defines the objects in arrears. It is these objects that are monitored to determine if the cut process (and overdue process) can be cancelled.
- The log associated with a cut process's overdue process contains the history of the cut process(es) and their events; there is no log specific to a cut process.

The Components Of A Cut Process

Cut processes are simpler than overdue processes as they simply contain one or more "cut events". These events are the service agreement-specific actions designed to encourage the customer to pay. For example, you might set up cut events that:

- Create field activities to cut service

Copyright © 2026, Oracle and/or its affiliates.

- Send letters (via the creation of a customer contact)
- Create To Do entries
- Impact the account's credit rating
- Create a case (e.g., to manage a customer with life support issues)
- ... (the list is only limited by your imagination as algorithms are used to perform the event's actions)

You define the number and type of events by configuring cut process templates. When the system creates a cut process, it copies the events defined on the related template.

It's important to note that all cut events are created when the cut process is created. The Overdue / Cut Event Manager is responsible for activating the cut events at the appropriate time. Activation of an event causes the system to do whatever the event indicates (e.g., send a letter, send a To Do entry to a user, create a field activity, etc.)

Cut processes do not have their own log as their history is maintained on the parent overdue process's log.

Cut Process Information Is Overridable

"Cut process info" is the concatenated string of information that summarizes a cut process throughout the system. The base-package logic constructs this string by concatenating the following information:

- Its cut process template's description
- Its status
- For active processes, the number of days since it was created. For inactive processes, the number of days since it was inactivated.
- For active processes, the unpaid amount of the objects being collected on the cut process's overdue process

If you'd prefer a different info string, you can develop a new algorithm and plug-it in on your cut process templates. This design allows some / all cut process templates to have an override info string.

How Are Cut Processes Cancelled?

A user may cancel a cut process at their discretion, online / real-time using Cut Process - Main.

The system will automatically cancel a cut process when the overdue process is cancelled.

Regardless of what triggers the cancellation, the Cancel Cut Process algorithm plugged in on the cut process's template handles the cancellation.

If a cut process is canceled and the cut process's events created field activities, the cancel logic embodied in the algorithm can be quite sophisticated. Refer to Field Activities To Cut And Reconnect Service for an example.

Cut Processes and Events Are Created From Templates

Copyright © 2026, Oracle and/or its affiliates.

When a cut process is created, its cut events are created by copying a cut process template's event types. Cut events follow the same lifecycle and possess the same behavior as overdue events.

Cut Events Are Like Overdue Events

Cut events almost identical to overdue events. The major difference is that cut events exists under a cut process and therefore are oriented towards the cut process's service agreement. The following topics describe other differences between cut events and overdue events.

- Field Activities To Cut and Reconnect Service
- A Cut Event Expires Service Agreements (and May Trigger Final Billing)

Field Activities To Cut and Reconnect Service

A cut process that's created for a service agreement whose service can be severed will typically contain a cut event that creates field activity(s) to cut service. The base-package cut event activation algorithm will create field activities for the service points linked to the service agreement being cut. The type of activity is defined on the field activity type profile defined on each service point's SP type. Please see C1-CE-CR-FA for the exact details of the base-package algorithm.

If the cut process is canceled after field activity(s) are created and before the cut process completes, different activities can transpire. The specifics of what happens are controlled by a Cancel Logic algorithm plugged in on the cut process template. The following points summarize how a base-package algorithm (C1-CP-DWFA) works:

- It assembles all Pending, Held and Complete cut for nonpayment field activities (FA) created by the cut process (these are defined in the overdue process's log).
- Each FA is processed as follows:
  - If the FA is Pending or Held and it is not linked to a field order (i.e., it hasn't been dispatched):
    - The FA is cancelled
    - An entry is added to the overdue process log to indicate the cancellation
  - If the FA is Pending or Held and it is linked to a field order (i.e., it has been dispatched):
    - A To Do entry is created (the To Do Type is defined on the algorithm)
  - If the FA is Complete and the Cut Process's SA is pending start or active (i.e., it hasn't been expired yet):
    - Reconnect field activity(s) are created for every service point that had a cut for non-payment field activity. The type of activity is defined on the field activity type profile defined on each service point's SP type.

A Cut Event Expires Service Agreements (and May Trigger Final Billing)

The last cut event is typically set up with an activation algorithm that expires the cut process's service agreement. If earlier cut events created "cut for non-payment" field activities, these

Copyright © 2026, Oracle and/or its affiliates.

field activities will be used as the basis for stopping service. Refer to Finalizing Pending Stops for how the system use the meter reads on these field activities as the "stop reads" on the service agreement. Note, you can see the field activities that are used to "cut" and "stop" service by viewing the Field Activities grid on Service Agreement - Service Point.

> **Note**
>
> **Final billing.** When the last service agreement linked to an account is expired, the system will schedule the account for billing (outside of its normal bill cycle schedule).

Write Offs Are Implemented Using Overdue Events

The system has been designed to allow overdue events on the original overdue process to write-off the objects being collected. Refer to Writing Off Bills for a case-study explaining how to do this.

Calendar vs Work Days

When you set up your overdue and cut process templates, you supply information that controls how each event's trigger date is calculated. You have two options:

- You can say that an event's trigger date can only be populated after earlier, dependent events are complete. For example, the 2 nd event (cut service) is triggered 2 days after the 1 st event is complete (48 hour warning letter).
- You can say that an event's trigger date is populated when the process is first created. You simply define the number of days after the start of the process when each such event should be triggered. For example, the 2 nd event (send cutoff warning) can be triggered 7 days after the start of the process.

In addition to the above, an option defined on the Feature Configuration for Overdue Processing plays a part in the calculation of an event's trigger date:

- If you set the option to use calendar days, the trigger date of events will be set to the first workday on / after the calculated date. For example, if you indicate that the 2 nd event is triggered 7 days after the 1 st event, the system will add 7 days to the 1 st event's completion date. It then checks if this is a workday (and not a holiday); if so, this is the trigger date of the event; if not, it assigns the trigger date to the next workday.
- If you set the option to use workdays, the trigger date will be calculated by counting workdays. For example, if you indicate that the 2 nd event is triggered 7 days after the 1 st event, the system will count 7 workdays and set the trigger date accordingly.

> **Note**
>
> **Account's division controls the work calendar.** The system uses the above information in conjunction with the work calendar associated with the account's division to determine workdays.

Copyright © 2026, Oracle and/or its affiliates.

Bill-Oriented Collection - Advanced Topics

The topics in this section provide information on features designed to facilitate the collection of overdue bills.

- Miscellaneous Bill-Oriented Collection Topics
- Writing Off Bills
- Bill-Oriented Payment Arrangements

Miscellaneous Bill-Oriented Collection Topics

The topics in this section provide background information about a variety of bill-oriented collection topics.

- Highlights Of The Sample Overdue Monitor Rule Algorithm
- Holding A Process's Events
- Bill Info Shows Unpaid and Write-off Amounts

Highlights Of The Sample Overdue Monitor Rule Algorithm

A base-package Overdue Monitor Rule algorithm exists to support many different types of overdue rules (see C1-CB-CR-RAT). Keep in mind that multiple such algorithms can be plugged in on Collection Class Overdue Rules. When the Overdue Monitor analyzes an account, it calls these algorithms until there are no more to invoke (or until an algorithm tells it that there is nothing more to check). The following points describe its various options:

- **Different rules based on credit ratings.** You can set up algorithms to only be applied to accounts with a credit rating <= a given value. We anticipate such algorithms will be used in conjunction with other such algorithms to apply different rules based on the customer's credit rating.
- **Different rules based on an account or service agreement characteristic.** You can set up algorithms to only be applied if an account or one of its service agreements has a given characteristic type and value effective within the last X days. For example, you could set up an algorithm that would process an account if it had a broken payment arrangement service agreement within the last 365 days (broken payment arrangement service agreements are marked with a given char type and value). You could set up a different algorithm that treated strategic customers more leniently (given that the definition of "strategic" could be held in an account or service agreement characteristic).
- **Skipping a bill if it has a future postpone date.** You can set up algorithms to ignore an unpaid bill if it has a "postpone date" that's in the future. This date would be defined on the bill using an ad hoc characteristic. The characteristic type is defined as a parameter on the algorithm.
- **Special process for bills with disputed debt.** You can set up an algorithm to create a given type of overdue process if the bill has disputed debt. By "disputed debt", we mean a bill with a financial transaction that's linked to an unbalanced match event that's marked as Disputed.

Copyright © 2026, Oracle and/or its affiliates.

- **Special process for bills with credit balances.** You can set up an algorithm to create a special overdue process if the bill's total charges are negative (i.e., a credit bill). Such bills should be very unusual as we strongly recommend that credit bills should have their credit balance applied to an overpayment service agreement during bill completion.
- **Different rules when all service agreements are inactive.** You can set up an algorithm that should only be applied if all service agreements are inactive. We anticipate this can be used to differ the type of overdue process as if an account does not have active service agreements; there is nothing to stop to encourage them to pay.
- **Different rules based on age and amount of a bill.** You can set up an algorithm to create different types of overdue processes based on the age and amount of the bills.

> **Note**
>
> **You are not limited by the base-package's rules.** If your implementation requires options that are not supported by the base-package algorithm, simply develop your own.

Holding A Process's Events

Refer to Holding Events for a description of how to prevent the activation of an overdue / cut processes events while a given condition is true. The condition is defined in an algorithm so you can set up the system to prevent event activation for practically any condition. For example, you might want to hold event activation while the bill's account has an active case (e.g., a user might set up a case to conduct an ad hoc investigation of billing discrepancies). When the case closes, you'd want the process to start up where it left off.

Bill Info Shows Unpaid and Write-off Amounts

The base-package Bill Information algorithm (plugged in on the Installation record) is responsible for constructing a bill's summary information that appears throughout the system. This algorithm can be configured to show the unpaid and write-off amounts of each bill.

Writing Off Bills

The topics in this section provide background information on how to write off bills.

- After Service Is Cut, Write-Off Oriented Events Can Commence
- Collection Agency Referrals
- Writing Off Bills Will Create Write-Off Adjustments And Possibly A Match Event
- Small Amount Write-Downs
- Bill Info Shows Written Off Amounts
- Online Write Off Of Bills Is Performed On Bill-Main
- Online Reversal Of Written Off Bills Is Performed On Bill-Main
- Write Offs And Overdue Process Cancellation

Copyright © 2026, Oracle and/or its affiliates.

- An Alert Can Show Written Off Bills

After Service Is Cut, Write-Off Oriented Events Can Commence

After the last service agreement has been stopped, overdue events to manage the write-off of the overdue process's bill(s) should be triggered. Please note that a separate overdue process is NOT created to manage write-offs. Rather, events associated with the original overdue process will handle the write-off activities.

Collection Agency Referrals

Before debt is written off, many implementations refer the unpaid bills to a collection agency. The following points describe how to implement this.

- Set up an overdue event type with an activation algorithm that refers an overdue process's bill(s) to a collection agency.
- When such an event is activated, the system creates a Collection Referral record for a collection agency. The specific agency to which the debt is referred is controlled by the event type's activation algorithm. The sample algorithm simply refers debt to the collection agency with the least amount of referred debt. If you prefer different logic, you must develop your own algorithm.
- A collection agency referral history record is linked to an account. It contains the amount of debt referred to the collection agency. It is the creation of this record that triggers the interface of information to the collection agency. The method used to interface the information to the agency is defined on the collection agency's record. Refer to Setting Up Collection Agencies for more information.
- If the collection agency is successful in obtaining the funds, a payment will be added. If the payment satisfies the cancel criteria defined on the overdue process template's cancellation plug-in, the overdue process will cancel. When an overdue process is cancelled, the cancel criteria on the overdue process's template are executed. We strongly recommend plugging in an algorithm that will cancel collection agency referrals when an overdue process is cancelled.
- If the collection agency is not successful in obtaining your funds after a given amount of time, you probably want to cancel the referral and write-off the debt. The cancellation of the referral will happen automatically if you set up your overdue process template to have an event that creates a collection agency cancellation X days after the referral. You can cancel a referral manually by simply creating a new collection agency referral history record (with a type of cancel).
- Collection agencies are notified of the cancellation of a referral by the creation of a new collection agency referral history record (with a type of cancel). This record will be interfaced to the agency in the same manner used to interface a new referral (see above).

> **Note**
>
> **Log entry.** The base-package overdue event activation algorithms that make and cancel collection agency referrals insert rows in the overdue process's log to audit these events.

Copyright © 2026, Oracle and/or its affiliates.

Writing Off Bills Will Create Write-Off Adjustments And Possibly A Match Event

Most overdue process templates will be configured to contain an event that writes-off the unpaid balance of bills.

> **Note**
>
> **Processing is redirected to another algorithm.** The base-package overdue event activation algorithm that writes off bills simply redirects the call to the Write-Off Bill algorithm plugged in on the account's Collection Class Overdue Rules. The reason for this redirection is because users can manually write-off a bill (using Bill - Main) and when they do this, we want to invoke the same logic as when an overdue process writes-off a bill.

The base-package Collection Class Overdue Rules - Write-Off Bill algorithm determines the amount that should be written off for each distribution code on each unpaid financial transaction. It then creates a separate adjustment for each service agreement where the lines on the adjustment contain the amount to be written off for each distribution code. If the unpaid financial transactions are not linked to a match event, the write-off adjustments plus the unpaid financial transactions will be linked to a new match event. If the unpaid financial transactions were linked to an unbalanced match event, the write-off adjustments will be added to the existing match event (thus making it balanced).

If partial payments were made against a bill, the amount written off will be prorated in light of the partial payments. For example, if 20% of a bill had been paid, 80% of each distribution code will be written off.

For example, assume the original bill's FT looked as follows (note, if the bill has multiple service agreements this will need to be done for each SA's FT's):

- Debit A/R $110.00
- Credit Flat Charge Revenue ($50)
- Credit Usage Revenue ($50)
- Credit City Tax Payable ($5)
- Credit State Tax Payable ($5)

Assume the customer had made a partial payment of $11 and it was matched to this FT. At write-off time, the system will create an adjustment whose adjustment lines will cause each distribution code to be written off by 90% (100% - $11 / $110). For example:

- Debit Flat Charge Revenue (or some W/O Expense) $45
- Debit Usage Revenue $45
- Debit City Tax Payable $4.50
- Debit State Tax Payable $4.50
- Credit A/R $99

This adjustment will then be linked to the original match event on which the payment was linked (thus making it Balanced).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Log entry.** The base-package overdue event activation algorithm that writes off bills inserts a row into the overdue process's log for each bill written off.

Small Amount Write-Downs

Many organizations will write-down a bill whose value is small early in an overdue process. The base-package overdue event activation algorithm has parameters to support this requirement (this algorithm allows you to write-off an overdue process's bill(s) if their value is less than a threshold amount).

If your organization writes-down small amounts differently than large amount, simply set up an overdue event type to reference such an activation algorithm and position it in the appropriate place in the overdue process template.

Bill Info Shows Written Off Amounts

The base-package Bill Information algorithm (plugged in on the Installation record) is responsible for constructing a bill's summary information that appears throughout the system. This algorithm can be configured to show the amount written-off for each bill.

Online Write Off Of Bills Is Performed On Bill-Main

If a bill's account is associated with a Collection Class Overdue Rule that has a Write Off Bill algorithm, a button appears on Bill - Main if the bill hasn't been written off. When clicked, the Write Off Bill algorithm is invoked to write-off the bill.

Online Reversal Of Written Off Bills Is Performed On Bill-Main

If a bill's account is associated with a Collection Class Overdue Rule that has a Write Off Bill algorithm, a button appears on Bill - Main if the bill has been written off. When clicked, the Write Off Bill algorithm is invoked (in reversal mode). The algorithm cancels the write-off adjustments (thus making the bill payable again). It also schedules the account for review by the Overdue Monitor the next time it runs.

Write Offs And Overdue Process Cancellation

It should be stressed that writing-off a bill may cause an overdue process to be canceled because the bill's FT will be linked to a balanced match event (note, the specific cancel criteria are in a plug-in so this is not a hard rule).

The following points highlight interesting aspects of bill write-off and overdue process cancellation:

Copyright © 2026, Oracle and/or its affiliates.

- If a user writes off a bill real time and the bill has an Active overdue process, the process's cancel criteria will be invoked. This typically results in the cancellation of the overdue process.
- If an overdue event writes off a bill, the state of the process depends on your cancel criteria and where the overdue event is positioned in the overdue process. For example,
  - Imagine an overdue process that has an overdue event that writes offs small amounts of debt early in the process. If such a process is applied against bill with a small amount of debt, the process will be canceled (when the event activates).
  - Contrast this to an overdue process where the last event writes off the bill. Because there are no other events to activate, the process will complete (i.e., it will not be canceled).

An Alert Can Show Written Off Bills

A base-package Control Central Alert algorithm (C1-WO-BILL) can be set up to highlight if the account in context has written off bills. This algorithm is plugged in on the Installation record.

Bill-Oriented Payment Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in **billed** installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

> **Note**
>
> **Nomenclature.** Some organizations refer to payment arrangements as "current bill plus" agreements because the customer's bills contain charges for both their current debt plus their installment amount. After the customer has paid off their overdue debt, the payment arrangement closes and the customer's bills only contain charges for their current debt.

The topics in this section describe how to set up a payment arrangement and how the system monitors the ongoing arrangements.

- Creating Payment Arrangements
- Installment, Payoff and Current Amounts
- Breaking A Bill-Oriented Payment Arrangement
- Online Breaking
- Online Canceling

Creating Payment Arrangements

When you create a payment arrangement, you are actually creating a service agreement. This service agreement is just like other service agreements in that:

Copyright © 2026, Oracle and/or its affiliates.

- It holds debt.
- It is periodically billed (thus creating unmatched bill segment financial transactions).
- When a payment is received, the payment segment financial transactions are matched to the bill segment financial transactions.

Debt is transferred to a payment arrangement service agreement (PA SA) from the customer's delinquent bills at the inception of the payment arrangement.

When you transfer debt from the overdue bills to a PA SA, transfer adjustments are created to transfer debt from the delinquent SAs to the PA SA. Match events are created to link the "transfer from" adjustments to the original unpaid financial transactions (FT). When all FT's on a bill are linked to balanced match events, the bill is no longer considered overdue and any active overdue process (and its related cut processes) will be cancelled. Refer to How Are Overdue Processes Canceled for more information.

> **Note**
>
> **Use the Payment Arrangement Transaction.** Use the Payment Arrangement - Bill Oriented to set up bill-oriented payment arrangements. This transaction creates a PA SA, transfers overdue bills to it, and sets up the installment amount. This transaction is also used if you need to break or cancel the payment arrangement.

Installment, Payoff and Current Amounts

> **Warning**
>
> If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When you set up a payment arrangement service agreement (PA SA), you transfer delinquent debt to the PA SA using transfer adjustments. After moneys are transferred, the system sets the PA SA's current balance to zero. At this point, there will be no overdue bills. If the customer neglects to pay bills containing charges associated with the payment arrangement, an overdue process will ensue.

PA SAs start their life with a non-zero payoff balance (i.e., they have debt when first started). This debt is transferred from the bills whose outstanding debt necessitated the creation of the PA SA.

The installment amount that the customer is billed is determined by the number of installments used to payoff the debt. For example, if the customer owes $500 and they want to pay this off in 10 installments, you'd set up the installment amount to be $50. The installment amount is saved on the PA SA's recurring charge amount. If the customer again falls into arrears on their bills, you can transfer additional bills to the PA SA. You can also change the installment amount as needed.

A PA SA's payoff balance typically differs from its current balance. The payoff balance is the amount of debt remaining to be paid off under the terms of the payment arrangement. The current balance is the installment amount that has been billed but not paid. For example, a

Copyright © 2026, Oracle and/or its affiliates.

customer who is paying off $500 with 10 installments of $50 would have an initial payoff balance of $500 and a current balance of $0. After the first bill, the PA SA would still have a payoff balance of $500, but its current balance would be $50. When the customer pays, the PA SA's payoff balance would fall to $450 and its current balance would return to $0.

The following table contains a financial example of a customer who sets up a payment arrangement to payoff $1,000 of debt in $10 installments.

Event	Normal SA's GL Accounting	PA SA's GL Accounting	Normal SA's Current Balance	Normal SA's Payoff Balance	PA SA's Current Balance	PA SA's Payoff Balance
Prior to creation of payment arrangement	N/A	N/A	1000	1000	N/A	N/A
Transfer debt from normal SA(s) to PA SA	Xfer 1000 A/R <1000>	PA A/R 1000 Xfer <1000>	0	0	1000	1000
Set current balance to zero on PA SA	N/A	N/A	0	0	0	1000
Customer is billed ($50 for new debt and $10 of payment arrangement debt)	A/R 50 Revenue <50>	N/A	50	50	10	1000
Customer pays $60	Cash 50 A/R <50>	Cash 10 PA A/R <10>	0	0	0	990

When the customer pays off the payment arrangement debt, the system automatically closes the PA SA after it final bills (assuming the PA SA's SA type references a bill segment type that has a bill segment creation algorithm of Recurring Charge With Auto Stop).

Breaking A Bill-Oriented Payment Arrangement

If a customer neglects to pay a bill, an overdue process is created. If the bill contains charges from a payment arrangement, you'll want to "break" the payment arrangement. By "break", we mean cancel the arrangement and reinstate the debt under the originating bills. To do this, you must configure the cut process template used to cut the PA SA to contain a cut event that breaks the payment arrangement.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Processing is redirected to another algorithm.** The base-package cut event activation algorithm that breaks a payment arrangement simply redirects the call to the Bill-Based Payment Arrangement algorithm plugged in on the account's Collection Class Overdue Rules. The reason for this redirection is that users can manually break a payment arrangement (using Payment Arrangement - Bill Oriented). When they do this, we want to invoke the same logic as when a cut event activation algorithm breaks a payment arrangement.

The base-package Collection Class Overdue Rule - Bill-Based Payment Arrangement algorithm does the following:

- Creates "cancel adjustments" for all unpaid financial transactions associated with the payment arrangement (e.g., billed, but not paid installments).
- Cancels ALL adjustments that were used to transfer the debt to the payment arrangement. When these are cancelled, the original bills will become unpaid (and the debt will be rather old by this point).
- Syncs up current balance with payoff balance on the PA SA.
- Makes the PA SA pending stop and expires it (SA activation will stop the SA when it next runs).
- If there is a credit left on the PA SA (because payments were made against the arrangement), the credit will be distributed amongst the bills that contributed debt to it (the oldest bills are paid off first). This relief will occur via transfer adjustments from the PA SA to the original SAs.
- Note, if there is a debit left (e.g., because LPC were issued or some other type of adjustment was created by an operator), an error is issued as we don't handle new debit on a payment arrangement.
- Inserts a characteristic on the PA SA to indicate that it has been broken (you might want to use this to cause more strict overdue rules to be applied to accounts with broken payment arrangements).
- Marks the account so it will be reviewed by the Overdue Monitor the next time it runs.

> **Note**
>
> **The PA SA must final bill before it closes.** It's important to note that the PA SA will only close after the PA SA is final billed. This is OK as it won't have any money left on it.

When the Overdue Monitor next runs, it will analyze the account's reinstated bills. We recommend setting up a Collection Class Overdue Rule - Overdue Monitor Rule algorithm that has more stringent rules if the account has a broken payment arrangement within the last X days. The base-package algorithm (C1-CB-CR-RAT) supports this type of processing.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Log entry.** The base-package overdue event activation algorithm that breaks a payment arrangement SA inserts a row into the overdue process's log.

Online Breaking

A user can click a button on the Payment Arrangement - Bill Oriented page to break a payment arrangement real time. When clicked, the Collection Class Overdue Rule - Bill-Based Payment Arrangement algorithm is called to break the payment arrangement. This is the same algorithm called when a cut event break a payment arrangement.

Online Canceling

A user can click a button on the Payment Arrangement - Bill Oriented page to cancel a payment arrangement real time. Cancellation should be used when you want to "logically delete" a PA SA because it shouldn't have been created.

The logic described above for breaking a payment arrangement is executed when a user cancels a payment arrangement; the only difference is that the PA SA is marked with a different characteristic type / value than when it is broken. The "broken" characteristic type / value can be used to apply stricter rules to the account when it's next reviewed by the Overdue Monitor.

Creating Overdue and Cut Procedures

Your overdue procedures define how your organization collects overdue debt. Your cut procedures define how your organization cuts (i.e., stops) service agreements when collection attempts fail. In this section, we describe how to set up the data that controls these procedures.

> **Warning**
>
> There are numerous ways to design your overdue and cut procedures. Some designs will result in easy long-term maintenance; others will result in maintenance headaches. In this section, we provide information to help you understand the ramifications of the various options. Before you set up your overdue and cut procedures, we encourage you to gain an intuitive understanding of these options by using the system to prototype the alternatives.

- Designing Your Overdue Procedures
- Set Up Tasks
- Setting Up Overdue Processing

Copyright © 2026, Oracle and/or its affiliates.

Designing Your Overdue Procedures

The design of your overdue procedures is an iterative process. Over time, you will develop skills that will allow you to skip some steps. However, when you're starting out, we recommend you use the following matrix as your guide. When the matrix is complete, you're ready to set up the overdue processing control tables.

Account's Division	Account's Collection Class:	Account's Collection Class:

The topics discussed below will gradually complete this matrix using a simple case study.

> **NOT_SUPPORTED**
>
> For more information about how the information in this matrix is used to monitor your customers' debt, refer to How Does The Overdue Monitor Work.

- Your Divisions Are Frequently Preordained
- Designing Your Collection Classes
- Designing Overdue Monitoring Rules
- Designing Overdue Process Templates and Event Types
- Designing Cut Process Templates and Event Types

Your Divisions Are Frequently Preordained

An account's division defines the jurisdiction whose rules govern the account. For example, an account's division controls how its payments are distributed, if / how late payment charges are levied, etc. Divisions have typically been designed in advance of designing your overdue rules.

In our example, we assume you have two divisions: North and South.

Account's Division	Account's Collection Class:	Account's Collection Class:
North		
South		

Designing Your Collection Classes

Multiple collection classes are needed when your organization has different overdue rules and / or procedures based on the type of customer. If all customers are treated the same way, you'll have a single collection class (call it Generic). However, if you're like many organizations, you will have multiple collection classes.

For example, for commercial/industrial customers, you probably don't worry until they owe you more than, say, $100 after 20 days. For residential customers, you probably don't worry until

Copyright © 2026, Oracle and/or its affiliates.

they owe you more than, say, $5 after 20 days. In this situation, you will have at least two collection classes: one for commercial/industrial customers, the other for residential customers.

In our example, we assume you have two collection classes: Residential and Commercial/ Industrial.

Account's Division	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
North		
South		

> **Note**
>
> There are two very different ways to monitor your accounts for overdue debt. This chapter describes the method referred to as Overdue Processing. Refer to Collection / Severance / Write Off for a description of the other method. We anticipate that most organizations will only use a single method. If your organization opts to use both methods, you will need to set up the corresponding collection classes.

Designing Overdue Monitoring Rules

At this point, we have the rows and columns defined in our matrix. Now it's time to work on the individual cells.

> **Note**
>
> **Single currency.** We've assumed that your implementation works in a single currency. If this is not true, you will need to add a 3rd dimension that will have a value for each currency code.

Each cell will contain the rules that the Overdue Monitor uses to determine if an account has overdue debt. These rules will eventually be configured using one or more algorithms on Collection Class Overdue Rules.

Account's Division	Account's Collection Class: Residential	Account's Collection Class: Commercial/Industrial
North		
South		

> **Note**
>
> If the Overdue Monitor encounters an account whose collection class and division does not have overdue rules set up, it will issue an error.

Copyright © 2026, Oracle and/or its affiliates.

Determining the rules in each cell can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For residential debt (regardless of division) we have the following rules:
  - Highest priority. If a bill exists with unpaid FT's > $0 that is older than 50 days, create the "accelerated overdue process" for residential customers. We'll talk more about this process later.
  - Medium priority. If the account's has a broken payment arrangement within the last 60 days with an unpaid bill > $0 that is older than 20 days, create the "broken payment arrangement overdue process".
  - Lower priority. If a bill exists with unpaid FT's > $25 that is older than 25 days, create the "courtesy reminder overdue process" for residential customers. We'll talk more about this overdue process later.
- For commercial-industrial debt (regardless of division) we have the following rules:
  - Highest priority. If a bill exists with unpaid FT's > $0 that is older than 45 days, create the "commercial 45 days late overdue process". We'll talk more about this process later.
  - Medium priority. If the account's credit rating is < 550 and has an unpaid bill > $0 that is older than 20 days, create the "risky commercial customer overdue process".
  - Lower priority. If a bill exists with unpaid FT's > $100 that is older than 30 days, create the "commercial 30 days late overdue process". We'll talk more about this overdue process later.

Given the above, our matrix will look as follows:

Account's Division	**Account's Collection Class:** Residential	**Account's Collection Class:** Commercial/Industrial
North	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 50 days, create the accelerated overdue process for residential customers. Medium Priority: If the account has a broken payment arrangement within the last 60 days with an unpaid bill > $0 that is older than 20 days, create the broken payment arrangement overdue process. Lowest Priority: If a bill exists with unpaid FT's > $25 that is older than 25 days, create the courtesy reminder overdue process for residential customers.	Highest Priority: If a bill exists with unpaid FT's > $0 that is older than 45 days, create the commercial 45 days late overdue process. Medium priority. If the account's credit rating is < 550 and has an unpaid bill > $0 that is older than 20 days, create the risky commercial customer overdue process. Lowest Priority: If a bill exists with unpaid FT's > $100 that is older than 30 days, create the commercial 30 days late overdue process.
South	Same as above	Same as above

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The rules are limited by your imagination (and business requirements).** While the base-package Overdue Monitor Rule algorithm (C1-CB-CR-RAT) supports the above scenarios, we'd like to stress these are just examples. Your implementation can operate using very different rules by either configuring the base-package algorithm (it has many parameters that you can use to tailor your rules) OR by introducing a new algorithm. Refer to Highlights Of The Sample Overdue Monitor Rule Algorithm for the options delivered with the base-package.

Designing Overdue Process Templates and Event Types

The following table shows a sample overdue process template for one of the rules in the Residential / North cell in the previous section's matrix.

Overdue Process Template	Overdue Event Type	When Triggered
Accelerated overdue process for residential customers	Old debt letter	At inception of process
	Cut active service agreements	10 days after inception
	Reduce customer's credit rating	10 days after inception (i.e., at the same time the cut process is created)
	Write down small debt	0 days after completion of the cut process(es)
	Refer debt to collection agent	0 days after attempting the small write down (this means that either the small write-down or the agency referral will take place as if the write-down is successful, the bill's FTs will be matched to balanced match events and the overdue process will stop)
	Cancel collection agent referral	45 days after referral
	Write-off debt	0 days after collection agent cancellation

You should create a similar table for each of the distinct overdue process templates in your matrix.

At this point, you've designed the distinct overdue process templates. Next, you'll need to design the algorithms that control their overdue processes:

- A template's Calculate Unpaid and Original Amount algorithm calculates the original and unpaid amounts of the objects being collected by a process. These values are used throughout the overdue processing module.
- A template's Cancel Criteria algorithm is executed to determine if a process should be cancelled. Refer to How Are Overdue Processes Cancelled for the details.
- A template's Cancel Logic algorithm is executed to cancel a process. Refer to How Are Overdue Processes Cancelled for the details. Please note that the logic embodied in this

Copyright © 2026, Oracle and/or its affiliates.

type of algorithm can be sophisticated because it is responsible for stopping an ongoing process's activities (e.g., this could involve cancelling field activities or cases). Cancellation algorithms are also responsible for inserting log entry(s).

- A template's Hold Event Activation algorithm is invoked to determine if the Overdue / Cut Event Manager should suspend the activation of the process's events.
- A template's Information algorithm is invoked to construct the override "info string".

Next, extract each unique event type from the above table:

Overdue Event Type	Action
Old debt letter	Create a customer contact
Cut active service agreement(s)	Start a cut process for every active SA with an unpaid FT on the bill
Reduce customer's credit rating	Insert an account credit rating history record
Write down small debt	Create write-down adjustments if unpaid debt is less than $x
Refer debt to collection agent	Create a collection agency referral
Cancel collection agent referral	Cancel the collection agency referral
Write off unpaid debt	Create adjustments to write-off unpaid debt

At this point, you know the distinct event types. Next, you'll need to design the algorithms that control the lifecycle of each event type:

- The event type's Event Activation algorithm(s) are executed by the Overdue / Cut Event Manager on its trigger date. The following points describe the logic embodied in such an algorithm:
  - The activity that happens on the trigger date (e.g., creation of a customer contact, To Do, etc.). Refer to Overdue Events Can Do Many Things for the details.
  - Whether the event is transitioned into the Waiting or Complete state when it's triggered. Refer to Some Events Can Wait for the details.
  - How the log entry(s) associated with event activation will be constructed. The base-package algorithms allow you to control the verbiage in the log entry by defining the desired message number on the algorithm. This means that you may have to set up new messages. Refer to Activating Events Should Add A Log Entry for the details.
- The event type's Cancel Logic algorithm(s) are invoked when an event is cancelled. The following points describe the logic embodied in such an algorithm:
  - If the event is allowed to be canceled. This logic may be necessary if some conditions prevent events of this type from canceling. For example, you may want to prevent an event from canceling when there are later dependent events that aren't canceled.
  - Any ancillary actions that take place during cancellation.
  - How the log entry(s) associated with event cancellation will be constructed.
- The event type's Monitor Waiting Event algorithm(s) are invoked to monitor a waiting event. These algorithms are responsible for transitioning a Waiting event to Complete if the object on which it's waiting is complete.
- The event type's Event Information algorithm is invoked to construct the override "info string".

Copyright © 2026, Oracle and/or its affiliates.

Once you've designed each event type's algorithms, you're ready to design your cut processes.

Designing Cut Process Templates and Event Types

While not required, many overdue processes will contain an overdue event that cuts (i.e., stops) service (in the hopes that stopping service will inspire the customer to pay). When such an event activates, it creates one or more cut process(es).

The system allows you to control if and how service is cut by setting up Cut Process Rule algorithms on your SA types. This allows different rules for different types of service agreements. In addition, because these rules are embodied in algorithms, a given SA type can have conditional logic that controls the type of cut process created. For example, you may have a different cut process if the customer has life support equipment, or if it's winter, or ...

> **Note**
>
> **If a service agreements of a given type should never be cut.** If you have certain SA types that should never be cut, do NOT define an algorithm in the SA type's Cut Process Rule .

Determining your Cut Process Rules can be straightforward or complicated; it depends on how your organization works. Our case study assumes the following:

- For service agreements associated with metered service, we have the following rules:
  - Highest priority. If the customer has life support requirements, create "cut process in light of life support" process.
  - Lowest priority. Otherwise, create a "metered service" cut process.
- For service agreements associated with charitable contributions, create an "expiration only" cut process.

Once you've determined if / how each SA type is cut, you need to design your cut process templates (you'll need one for each unique method of cutting a service agreement). The following table shows the cut process templates referenced above. Adjacent to each process are its events and an indication of when they are triggered.

Cut Process Template	Event Number	Cut Event Type	Dependent On Event(s)	Trigger Date Set To X Days After Completion Of Dependent Events
Metered service	10	Letter - 48 hour disconnect for non-payment warning	N/A - first event	0
	20	Field activity - disconnect for non-payment	10	2
	30	'Service has been disconnected' letter	20	0

Copyright © 2026, Oracle and/or its affiliates.

	40	Expire service agreement	20	10
Cut process in light of life support	10	Generate delinquent life support customer To Do entry (seeking approval for the cut)	N/A - first event	0
	20	Letter - 72 hour disconnect for non-payment warning	10	0
	30	Generate impending life support cutoff To Do entry to C&C rep	20	3
	40	Field activity - cut for non-payment	30	0
	50	Service has been disconnected letter	40	0
	60	Expire service agreement	40	10
Expiration only	10	Expire service agreement	N/A - first event	

At this point, you've designed the distinct cut process templates. Next, you'll need to design the algorithms that control the lifecycle of their cut processes:

- A template's Cancel Logic algorithm is executed to cancel a process. In addition to cancelling the event, these algorithms are also responsible for inserting log entry(s).
- A template's Information algorithm is invoked to construct the override "info string".

Next, extract each unique event type from the above table:

Cut Event Type	Action
48-hour warning letter	Create a customer contact - 48-hour warning letter (letters are created via customer contacts)
72-hour warning letter	Create a customer contact - 72-hour warning letter
Disconnect for non payment	Create a cut for non-payment field activity
Permission to start a cut process for a life support customer	Create To Do seeking permission to send 72 hour letter
Permission to create a cut field activity for a life support cutoff	Create To Do seeking permission to issue a cut field activity
Service has been disconnected letter	Create a customer contact - service cut letter
Expire service agreement	Expire service agreement

At this point, you know the distinct event types. Next, you'll need to design the algorithms that control the lifecycle of their events:

Copyright © 2026, Oracle and/or its affiliates.

- The event type's Event Activation algorithm(s) are executed by the Overdue / Cut Event Manager on its trigger date. The following points describe the logic embodied in such an algorithm:
  - The activity that happens on the trigger date (e.g., creation of a customer contact, To Do, etc.). Refer to Cut Events Can Do Many Things for the details.
  - Whether the event is transitioned into the Waiting or Complete state when it's triggered. Refer to Some Events Can Wait for the details.
  - How the log entry(s) associated with event activation will be constructed. The base-package algorithms allow you to control the verbiage in the log entry by defining the desired message number on the algorithm. This means that you may have to set up new messages. Refer to Activating Events Should Add A Log Entry for the details.
- The event type's Cancel Logic algorithm(s) are invoked when an event is cancelled. The following points describe the logic embodied in such an algorithm:
  - If the event is allowed to be canceled. This logic may be necessary if some conditions prevent events of this type from canceling. For example, you may want to prevent an event from canceling when there are later dependent events that aren't canceled.
  - Any ancillary actions that take place during cancellation.
  - How the log entry(s) associated with event cancellation will be constructed.
- The event type's Monitor Waiting Event algorithm(s) are invoked to monitor a waiting event. These algorithms are responsible for transitioning a Waiting event to Complete if the object on which it's waiting is complete.
- The event type's Event Information algorithm is invoked to construct the override "info string".

Set Up Tasks

The above topics provided background information about how overdue processing works. The following discussion summarizes the various set up tasks.

- Cut Event Types
- Cut Process Templates
- SA Type - Cut Process Rules
- Overdue Event Types
- Overdue Process Templates
- Collection Classes
- Collection Class Overdue Monitor Rules
- Feature Configuration
- Overdue and Cut Event Cancellation Reasons
- Collection Agencies
- Alert To Highlight Active Overdue Processes
- Bill-Oriented Collection - Additional Set Up

Copyright © 2026, Oracle and/or its affiliates.

Cut Event Types

You will find that most of the time spent setting up your cut event types is spent setting up the objects that are referenced on the cut event type algorithms. For example, if you use the base-package algorithms, you may need to set up the following:

- The various "types" for the objects created by the plug-ins. For example, if a cut event type creates a To Do entry, you must supply the desired To Do type.
- Foreign key characteristic types that are used to reference the ancillary objects in the log entries (e.g., if an event creates a customer contact, the log references this customer contact using a FK characteristic type). Note, many of these will exist in the base-package.
- Messages that are used to define the verbiage in the log entries. For example, if you use the base-package algorithm that creates a customer contact, you must supply the desired message category and number that contains the verbiage that appears in the log when customer contacts are created. Note, messages have been set up for all base-package algorithms (this means you should not have to set up new messages).
- Etc.

The only way to compile the complete list is to design the parameters for each cut event type algorithm. Refer to Cut Event Type - Main for the supported plug-in spots.

After you've set up the objects referenced on the algorithms, you can then set up the algorithms. Only then can you set up the cut event types.

Cut Process Templates

After your cut event types exist, you can set up your cut process templates. You will find that most of the time spent setting up your cut process templates is spent setting up the objects that are referenced on the cut process template algorithms. Refer to Cut Process Template - Main for the supported system events.

SA Type - Cut Process Rules

After you've created your cut process templates, you can set up the algorithms that hold your cut process rules. These are plugged-in on SA Type - Algorithm in the Cut Process Rule system event.

Overdue Event Types

You will find that most of the time spent setting up your overdue event types is spent setting up the objects that are referenced on the overdue event type algorithms. For example, if you use the base-package algorithms, you will set up the following:

- The various "types" for the objects created by the plug-ins. For example,
  - If an overdue event type creates a To Do entry, you must set up the To Do type.

Copyright © 2026, Oracle and/or its affiliates.

  - If an overdue event type creates a customer contact, you must set up the customer contact type.
  - If an overdue event type writes off debt, you must set up the adjustment types.
  - Etc.
- Foreign key characteristic types that are used to reference the ancillary objects in the log entries (e.g., if an event creates a customer contact, the log references this customer contact using a FK characteristic type). Note, many of these will exist in the base-package.
- Messages that are used to define the verbiage in the log entries. For example, if you use the base-package algorithm that creates a customer contact, you must supply the desired message category and number that contains the verbiage that appears in the log when customer contacts are created. Note, messages have been set up for all base-package algorithms (this means you should not have to set up new messages).
- Etc.

The only way to compile the complete list is to design the parameters for each overdue event type algorithm. Refer to Overdue Event Type - Main for the supported plug-in spots.

After you've set up the objects referenced on the algorithms, you can then set up the algorithms. Only then can you set up the overdue event types.

Overdue Process Templates

After your overdue event types exist, you can set up your overdue process templates. You will find that most of the time spent setting up your overdue process templates is spent setting up the objects that are referenced on the overdue process template algorithms. Refer to Overdue Process Template - Main for the supported system events.

Collection Classes

Set up collection classes as per your overdue procedures. Make sure to indicate that these collection classes use the Overdue collection method (only accounts linked to collection classes designated as using the Overdue collection method or processed by the Overdue Monitor).

Collection Class Overdue Monitor Rules

After your overdue process templates exist, you can set up your Overdue Monitor Rules. These rules are algorithms plugged in on Collection Class Overdue Rules. You will find most of the time spent setting up these algorithms is spent setting up the objects referenced on the base-package algorithm.

Feature Configuration

You must set up a Feature Configuration to define parameters that control various overdue processing options.

The following points describe the various **Option Types** that must be defined:

Copyright © 2026, Oracle and/or its affiliates.

- Trigger Date: Y-Workdays, N-Calendar Days. This option controls how the system computes the trigger dates on overdue and cut events. Enter Y if the system should use workdays. Enter N if the system should use calendar days. Refer to Calendar vs Work Days for the details.
- Payment Arrangement Type (B/S/A). This option indicates whether your implementation uses the Balance-Oriented Payment Arrangements (value **S**), Bill-oriented Payment Arrangements (value **B**) or both (value **A**). The value governs the navigation path for the payment arrangement lines in the Account History and Credit & Collection zones.
- Champion Template$Challenger Template$Percentage(1-100). You need only set up options of this type if your implementation implements Champion / Challenger functionality. Options of this type are entered in the format ABnnn where A is the overdue process template of the champion template, B is the overdue process template of the challenger template, and C is the percent of the time that the system should create the challenger template. The overdue monitor uses this option to override the champion overdue process template X% of the time with the challenger template. You may enter any number of these options (but only one per Champion Template).

Overdue and Cut Event Cancellation Reasons

Overdue events can be cancelled automatically and manually (at the discretion of a user). Regardless of the method of cancellation, a cancellation reason must be supplied. You set up your overdue event cancellation reasons using Overdue Event Cancellation Reason - Main and Cut Event Cancellation Reason - Main.

Collection Agencies

If you refer debt to collection agents, you must set up your collection agencies.

Alert To Highlight Active Overdue Processes

If you want an alert to appear if the account has active overdue processes, you must configure an appropriate Control Central Alert algorithm (C1-OD-PROC). This algorithm is plugged in on the Installation record.

Bill-Oriented Collection - Additional Set Up

The topics in this section provide information on additional set up requirements if you collect on unpaid bills.

- One Bill Per Match Event
- Bill-Based Payment Arrangements
- Bill-Based Write-off
- Alert To Highlight Written Off Bills
- Open-Item Bill Amount Plug-In

Copyright © 2026, Oracle and/or its affiliates.

One Bill Per Match Event

As mentioned earlier, a bill is considered paid if its financial transactions (FTs) are linked to a balanced match event. To determine a bill's outstanding amount, FTs from different bills cannot be commingled on the same match event (but it's OK for a bill's FTs to be on multiple match events). If you stick by the rule of "just one bill per match event" you will then be able to determine the outstanding balance of a partially paid bill. However, if you mix more than one bill under a match event, a particular bill's balance may become indeterminate.

The following Open-Item algorithm types have been provided by the base package to help enforce this rule:

- The Distribute by Bill Due Date Payment Distribution algorithm (C1-PYDS-BDU).
- The match by Bill ID Payment Distribution Override algorithm (C1-PDOV-PYBL).
- The FT cancellation FT Freeze algorithm (C1-CFTZ-COFT).

If any of your customized plug-ins and processes create match events, it is important that these too enforce this rule. You may want to refer to the base package algorithms as an example of how to do this.

Bill-Based Payment Arrangements

If you set up payment arrangements for unpaid bills, you must configure the Bill-Based Payment Arrangement algorithm and plug it in on your Collection Class Overdue Rules.

It's important to set up the adjustment types used during payment arrangement creation to NOT print on bills. This is because the base-package algorithm will match the adjustments used to transfer debt to the payment arrangement with the adjustment used to reduce the payment arrangement's current amount by the amount of the transfer if all adjustment types are set up to not print.

Bill-Based Write-off

If you write-off unpaid bills, you must set up the following:

- Set up the adjustment type that will be used to write-off an unpaid financial transaction. This adjustment type must be configured as follows:
  - **Adjustment Amount Type** must be Calculated Amount
  - Its distribution code is irrelevant as a separate calculation line will be created for each distribution code on the FT's that is written off and these lines will reference the appropriate distribution code.
  - It must reference an adjustment type char type / value that identifies it as one used to write-off a bill's FTs
  - The following algorithms must be defined on the adjustment type:
- The Generate Adjustment system event must reference an algorithm that has the responsibility of determining how to write-off a FT. This algorithm should be determining the FT's GL details and creating a separate adjustment calculation line for each GL detail. The base package is supplied with a sample algorithm that does this (C1-ADJG-WO).

Copyright © 2026, Oracle and/or its affiliates.

- The Adjustment Financial Transaction system event should reference an algorithm that impacts current, payoff and the GL by the amount being written off.
  - The distribution codes referenced on the financial transactions must be set up with a characteristic that holds the distribution code used to write-off the original amounts. For example,
    * Distribution codes used to record tax liabilities will typically reference the same distribution code for write-off (most organizations reverse tax liabilities at write-off time)
    * Distribution codes used to record revenue will typically reference a write-off distribution code used to record a write-off expense.
    * Distribution codes used to record receivables will typically not reference a write-off distribution code because receivables are implicitly written off when revenue and tax liabilities are written off.
  - Set up an adjustment cancellation code used when a users reverses a written-off bill (reversal involves canceling the write off adjustments).
  - Set up a Write Off Bill algorithm and plug it in on your Collection Class Overdue Rules. This algorithm will reference the adjustment type described in the previous point.

Alert To Highlight Written Off Bills

If you want an alert to appear if the account has bills with written-off debt, you must configure an appropriate Control Central Alert algorithm (C1-WO-BILL). This algorithm is plugged in on the Installation record.

Open-Item Bill Amount Plug-In

You must set up the algorithm that computes the original, unpaid, and write-off amounts of your open-item bills. This algorithm is called by other algorithms when these amounts are needed. This algorithm is plugged-in on Installation in the Determine Open Item Bill Amounts spot.

Setting Up Overdue Processing

The topics in this section describe how to set up the control tables to implement your overdue processing.

- Setting Up Overdue Event Types
- Setting Up Overdue Process Templates
- Setting Up Cut Event Types
- Setting Up Cut Process Templates
- Setting Up Collection Class Overdue Rules
- Setting Up Overdue Event Cancellation Reasons
- Setting Up Cut Event Cancellation Reasons

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Overdue Event Types

An overdue event type encapsulates the business rules that govern a given type of overdue event. Open **Admin**, and then **Credit & Collection**, and then **Overdue Event Type**, and then **Add** to set up overdue event types.

> **Note**
>
> **Recommendation.** Before using this transaction, we strongly recommend that you review The Big Picture Of Overdue Events.

Description of Page

Enter a unique **Overdue Event Type** code and **Description** for the overdue event type.

Use **Long Description** to provide a more detailed explanation of the purpose of the overdue event type.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel an overdue event. Refer to How Are Events Canceled for the details.
Event Activation	Required	This algorithm is executed to activate an overdue event on its trigger date. Refer to Overdue Events Can Do Many Things and How and When Events Are Activated for the details.
Event Information	Optional - only used if you want to override an overdue event's info string	This algorithm is executed to construct an overdue event's override info string. Refer to Overdue Event Information Is Overridable for the details.

Copyright © 2026, Oracle and/or its affiliates.

Monitor Waiting Events	Optional - only used if events of this type can enter the Waiting state	This algorithm is invoked by the Overdue / Cut Event Manager for events in the Waiting state. Refer to Some Events Wait For Something Before Completing for the details.

Setting Up Overdue Process Templates

An overdue process template encapsulates the business rules that govern a given type of overdue process. Open **Admin**, and then **Credit & Collection**, and then **Overdue Process Template**, and then **Add** to set up overdue process templates.

> **Note**
>
> **Recommendation.** Before using this transaction, we strongly recommend that you review The Big Picture Of Overdue Processes.

Description of Page

Enter a unique **Overdue Process Template** and **Description** for the overdue process template.

Collecting On Object defines the type of object managed by this overdue process. This field actually references a foreign key characteristic type that references the managed object. For example, if this overdue process template manages overdue bills, you'd reference a foreign key characteristic that references the bill object.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event** (note, all system event's are optional and you can define an unlimited number of algorithms for each event).

System Event	Optional / Required	Description
Calculate Unpaid & Original Amount	Required	This algorithm is executed to calculate the unpaid and original amounts of the objects associated with the overdue process. These amounts are shown on the overdue process page and in the base-package overdue info string.

Copyright © 2026, Oracle and/or its affiliates.

Cancel Criteria	Required	This algorithm is executed to determine if an overdue process can be cancelled. Refer to How Are Overdue Processes Cancelled for the details.
Cancel Logic	Required	This algorithm is executed to cancel an overdue process. Refer to How Are Overdue Processes Cancelled for the details.
Hold Event Activation Criteria	Optional - only used if overdue processes of this type can be suspended while some condition is true	This algorithm is executed to determine if the activation of overdue and cut events should be suspended. Refer to Holding Events for the details.
Overdue Process Information	Optional - only used if you want to override an overdue process's info string	This algorithm is executed to construct an overdue process's override info string. Refer to Overdue Process Information Is Overridable for the details.

The **Event Types** control the number and type of overdue events linked to an overdue process when it is first created. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event type:

- **Event Sequence.** Sequence controls the order in which the overdue event types appear in the scroll.
- **Overdue Event Type.** Specify the type of overdue event to be created.
- **Days After.** If **Dependent on Other Events** is on, events will be triggered this many days after the completion of the dependent events (specified in the grid). Set this value to 0 (zero) if you want the event triggered immediately after the completion of the dependent events. If **Dependent on Other Events** is off, events will be triggered this many days after the creation of the overdue process. Refer to How and When Events Are Activated for the details.
- If **Dependent on Other Events** is on, define the events that must be completed or cancelled before the event will be triggered.
  - **Sequence** is system-assigned and cannot be specified or changed.
  - **Dependent on Sequence** is the sequence of the dependent event.

Setting Up Cut Event Types

A cut event type encapsulates the business rules that govern a given type of cut event. Open **Admin**, and then **Credit & Collection**, and then **Cut Event Type**, and then **Add** to set up cut event types.

> **Note**
>
> **Recommendation.** Before using this transaction, we strongly recommend that you review Cut Events Are Like Overdue Events.

Copyright © 2026, Oracle and/or its affiliates.

Description of Page

Enter a unique **Cut Event Type** code and **Description** for the cut event type.

Use **Long Description** to provide a more detailed explanation of the purpose of the cut event type.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel a cut event. Refer to How Are Events Canceled for the details.
Event Activation	Required	This algorithm is executed to activate a cut event on its trigger date. How and When Events Are Activated for the details.
Event Information	Optional - only used if you want to override a cut event's info string	This algorithm is executed to construct a cut event's override info string. Refer to Cut Event Information Is Overridable for the details.
Monitor Waiting Events	Optional - only used if events of this type can enter the Waiting state	This algorithm is invoked by the Cut / Cut Event Manager for events in the Waiting state. Refer to Some Events Wait For Something Before Completing for the details.

Setting Up Cut Process Templates

A cut process template encapsulates the business rules that govern a given type of cut process. Open **Admin**, and then **Credit & Collection**, and then **Cut Process Template**, and then **Add** to set up cut process templates.

> **Note**
>
> **Recommendation.** Before using this transaction, we strongly recommend that you review The Big Picture Of Cut Processes.

Description of Page

Copyright © 2026, Oracle and/or its affiliates.

Enter a unique **Cut Process Template** and **Description** for the cut process template.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Cancel Logic	Required	This algorithm is executed to cancel a cut process. Refer to How Are Cut Processes Cancelled for the details.
Cut Process Information	Optional - only used if you want to override a cut process's info string	This algorithm is executed to construct a cut process's override info string. Refer to Cut Process Information Is Overridable for the details.

The **Event Types** control the number and type of events linked to a cut process when it is first created. The information in the scroll defines these events and the date on which they will be triggered. The following fields are required for each event type:

- **Event Sequence.** Sequence controls the order in which the cut event types appear in the scroll.
- **Cut Event Type.** Specify the type of cut event to be created.
- **Days After.** If **Dependent on Other Events** is on, events will be triggered this many days after the completion of the dependent events (specified in the grid). Set this value to 0 (zero) if you want the event triggered immediately after the completion of the dependent events. If **Dependent on Other Events** is off, events will be triggered this many days after the creation of the cut process. Refer to How and When Events Are Activated for the details.
- If **Dependent on Other Events** is on, define the events that must be completed or cancelled before the event will be triggered.
  - **Sequence** is system-assigned and cannot be specified or changed.
  - **Dependent on Sequence** is the sequence of the dependent event.

Setting Up Collection Class Overdue Rules

Collection class overdue rules contain algorithms that impact accounts associated with a given collection class, division and currency code are managed. Open **Admin**, and then **Credit & Collection**, and then **Collection Class Overdue Rules**, and then **Add** to set up collection class overdue rules.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Recommendation.** Before using this transaction, we strongly recommend that you review Different Overdue Rules For Different Customers.

Description of Page

Enter the **Collection Class**, **CIS Division** and **Currency Code** to which the rules apply.

The **Algorithms** grid contains algorithms that control important functions. You must define the following for each algorithm:

- Specify the algorithm's **System Event** (see the following table for a description of all possible events).
- Specify the **Algorithm** to be executed when the System Event executes. Set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.
  The following table describes each **System Event**.

System Event	Optional / Required	Description
Bill-Based Payment Arrangement	Optional - only specified if your implementation uses bill-oriented payment arrangements	This algorithm is executed to handle the creation, breaking and canceling of a Bill-Oriented Payment Arrangements.
Overdue Monitor Rule	Required	This algorithm is invoked by the Overdue Monitor to analyze an account's debt. Refer to How Does The Overdue Monitor Work for the details. If you have multiple rules (and therefore multiple algorithms), please take care when assigning the sequence number, as the Overdue Monitor will invoke these rules in sequence order.
Write Off Bill	Option - only specified if your implementation writes-off bills	This algorithm is executed to handle the write-off and write-off reversal of a bill. Refer to Writing Off Bills.

Setting Up Overdue Event Cancellation Reasons

An overdue event cancel reason must be supplied before an overdue event can be canceled. Open **Admin**, and then **Credit & Collection**, and then **Overdue Event Cancel Reason** to define overdue event cancellation reasons.

Description of Page

Enter an easily recognizable **Overdue Event Cancel Reason** and **Description** for each cancellation reason.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Cut Event Cancellation Reasons

A cut event cancel reason must be supplied before a cut event can be canceled. Open **Admin**, and then **Credit & Collection**, and then **Cut Event Cancel Reason** to define cut event cancellation reasons.

Description of Page

Enter an easily recognizable **Cut Event Cancel Reason** and **Description** for each cancellation reason.

Defining Prepaid Metering Options

The topics in this section describe how to set up the system to enable prepaid metering functionality.

> **Note**
>
> **Prepaid metering is optional.** The system configuration requirements described in this section are only relevant if your organization offers prepaid metering service.

- The Big Picture of Prepaid Metering
- Setting Up The System To Enable Prepaid Metering

The Big Picture of Prepaid Metering

Prepaid metering allows customers to pay for energy before it is actually used.

- How Does Prepaid Metering Work?
- Debit Meter vs Credit Meter
- Prepaid Metering Transactions Result in Adjustments
- Interfacing Prepaid Transactions
- Prepaid Transactions Can Go Into Suspense

How Does Prepaid Metering Work?

A traditional electronic prepaid metering system typically operates at three levels:

- Meters that are installed at the customer's home. These meters dispense energy up to the amount that the customer prepaid.

Copyright © 2026, Oracle and/or its affiliates.

- Vending stations located at the utility's offices or designated payment agencies. These stations/agencies sell prepaid credits to customers.
- Master stations that group vending stations together for administration, reporting and control purposes. Master stations communicate information - e.g. customer information, tariff changes, etc - to the vending stations. The vending stations, in turn, report detailed customer sales and other transaction information to the master station.

A prepaid customer goes to a vendor and purchases credits for their meter to dispense energy. The vendor has knowledge of the customer and the rate that the customer is paying. The vendor issues the credits in any of the following forms: a prepaid card, a token (to be inserted into the meter) or a series of numbers (to be keyed into the meter).

Debit Meter vs Credit Meter

Prepaid meters are commonly referred to as debit meters. Billed meter types are also called credit meters. A meter's state of being 'credit' or 'debit' translates to a specific meter configuration. Having this property set at the meter configuration level enables meters to be switched between 'credit' and 'debit' if needed.

Prepaid credits may be purchased for either new premises or existing premises. In the latter case, the existing premise could initially have a typical 'credit' meter. In this case, fieldwork may be involved in getting the meter switched from 'credit' to 'debit' (prepaid).

Prepaid Metering Transactions Result in Adjustments

There could be numerous types of prepaid transactions. But the most common types are prepaid sales and cancellations.

A sale or cancellation transaction includes a total amount that usually includes tax. In order to post to the general ledger, the prepaid transaction must result in a financial transaction. Moreover, separate GL entries must be created for the breakdown of revenue and tax. In some cases, it may also be necessary to calculate the usage from the revenue amount.

Calculated Adjustments are used to book revenue from these types of transactions. A base Generate Adjustment algorithm is available to let rate application generate the appropriate GL calculation lines and calculate the corresponding usage, if needed.

Prepaid adjustments do not affect a customer's balance.

Interfacing Prepaid Transactions

An adjustment upload batch process exists for uploading prepaid transactions that result in adjustments. Refer to Interfacing Adjustments From External Sources for more information.

Prepaid metering options need only be set up if your organization offers prepaid metering service to your customers. Refer to Defining Prepaid Metering Options for more information.

Prepaid Transactions Can Go Into Suspense

Copyright © 2026, Oracle and/or its affiliates.

A prepaid transaction goes into suspense if a prepaid SA could not be determined for any of the following reasons:

- The badge number on the transaction is not a valid meter in the system
- The badge number on the transaction is a valid meter in the system, but is:
  - Not linked to a prepaid meter configuration type
  - Not linked to a non-closed, non-canceled prepaid SA

This can be a common occurrence because of the likelihood that transactions get uploaded prior to meter installation/exchange or service agreement information being updated in the system.

When this situation happens, you probably still want to recognize the revenue by posting an adjustment to a suspense SA that you designate. Refer to the base sample Get Prepaid SA Using Badge Number (C1-SABYBADGE) algorithm for an example of how the suspense SA is specified.

A batch process exists for automatically resolving suspense adjustments. Refer to How Are Suspense Adjustments Resolved for more information.

Setting Up The System To Enable Prepaid Metering

The following sections describe the steps in setting up control information for prepaid metering.

- Meter Types
- Characteristic Types
- Algorithms
- Rate
- Adjustment Types
- Adjustment Type Profiles
- Service Agreement Types
- Bill Cycle
- Vendor Information

Meter Types

Prepaid meters are set up just like any other meter - i.e. the meter must reference a meter type and the meter's configuration must reference a meter configuration type.

To set up prepaid meter types you must do the following:

- Define the meter configuration types that will be used for recording prepaid usage. Mark each of these configuration types as Prepaid . Define one register with the appropriate UOM (e.g. kWh).
- Define the meter types that will be used for prepaid metering. For each meter type, associate one of the Prepaid meter configuration types that you defined. You can also mark each meter type as Prepaid Capable.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Base logic does not require the meter type to be marked as Prepaid Capable. The Prepaid indicator on the meter configuration type is used to identify prepaid meters.

Characteristic Types

The following characteristic types are needed if you are going to upload prepaid transactions as adjustments and need the system to determine the prepaid SA given a prepaid meter's badge number.

Refer to Setting Up Characteristic Types for more information.

- Prepaid Entity Characteristic Type
- Badge Number Characteristic Type

Prepaid Entity Characteristic Type

Create a characteristic type that identifies whether an SA Type is used for prepaid metering. For example:

- Characteristic Type = <code>
- Description = Prepaid Entity
- Char Entities = SA Type
- Subclass = predefined list
- Values = Y, N

Badge Number Characteristic Type

Create a characteristic type for specifying a badge number. For example:

- <code>
- Description = Badge Number
- Char Entities = Adjustment, Adjustment Type
- Subclass = Adhoc

Algorithms

The following algorithms need to be set up in order to store prepaid transactions in the system.

Refer to Setting Up Algorithms for more information.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The following sections describe basic set up needed for uploading 'sale' and 'cancellation' types of prepaid transactions. Your implementation team may have to define additional specific algorithms types for any other prepaid transaction type that you need to upload into or store in the system.

- Calculation Rule - Calculation Algorithm
- Adjustment Type - Generate Adjustment
- Adjustment Type - Determine SA
- Adjustment Type - Resolve Suspense
- Adjustment Type - Adjustment Information
- Installation - Adjustment Information

Calculation Rule - Calculation Algorithm

Create an algorithm of type Determine Percent Given Total (C1-PCTGVNTOT) specifying your parameter values for UOM / TOU / SQI.

Create an algorithm of type Back Into Revenue (C1-BACKINREV) specifying your parameter values for UOM / TOU / SQI.

Adjustment Type - Generate Adjustment

Create an algorithm of type Adjustment Generation - Apply Rate (ADJG-RT), specifying your prepaid rate and parameter values for UOM / TOU / SQI.

Adjustment Type - Determine SA

Create an algorithm of type Get Prepaid SA Using Badge Number (C1-SABYBADGE), specifying your badge number characteristic type, prepaid characteristic type and suspense SA ID (i.e. SA to which the adjustment needs to post when a valid prepaid SA is not found).

Adjustment Type - Resolve Suspense

Create an algorithm of type Cancel Suspense Adjustment (C1-CANSUSADJ) specifying the following:

- Badge number characteristic type
- Prepaid entity characteristic type
- Prepaid adjustment type
- Adjustment cancel reason

Copyright © 2026, Oracle and/or its affiliates.

If you want this algorithm to create to do entries for adjustments that have been in suspense for too long, specify additional parameter values for number of days and to do type and to do role (optional).

Adjustment Type - Adjustment Information

Specify the SUSPENSE_DESCR parameter on your adjustment information algorithm if you want the information to include an indication of suspense.

Installation - Adjustment Information

Specify the SUSPENSE_DESCR parameter on your adjustment information algorithm if you want the information to include an indication of suspense.

Rate

In order to generate separate calculation lines for the revenue and tax breakdown of the prepaid amount, you need to set up a prepaid rate with components that calculate revenue amount and tax amount given just a total amount.

Create a rate schedule with the following components:

- Tax Calculation Rule
  - Calculation Rule Type = Calculation Algorithm
  - Value Type = Percentage
  - Value Source = Bill Factor
  - Bill Factor = your tax bill factor
  - Calculation Algorithm = the one created for algorithm type C1-PCTGVNTOT. Refer to Calculation Rule - Calculation Algorithm for information.
  - Distribution Code = *your distribution code for tax liability*
- Revenue Calculation Rule
  - Calculation Rule Type = Calculation Algorithm
  - Derive SQ = checked
  - Value Type = Unit Charge
  - Value Source = Bill Factor
  - Bill Factor = your rate/kWh bill factor
  - UOM/TOD/SQI to use for the resulting calculated usage, for example kWh
  - RC Cross Reference = indicate the sequence for the tax calculation rule (above)
  - Calculation Algorithm = the one created for algorithm type C1-BACKINREV. Refer to Calculation Rule - Calculation Algorithm for information.
  - Distribution Code = your distribution code for revenue

Copyright © 2026, Oracle and/or its affiliates.

Adjustment Types

You must define an adjustment type for each type of transaction that you want to store in the system. Refer to Setting Up Adjustment Types for more information.

For prepaid sales and cancellation transactions, you need to define calculated adjustment types. In addition, plug-in the following algorithms:

- Adjustment FT Creation - GL Only algorithm - so that transaction amounts do not affect the customer's balance. Specify calc lines as the source for adjustment distribution codes.
- Generate Adjustment
- Determine SA
- Resolve Suspense, if applicable.
- Adjustment Information, if applicable

Adjustment Type Profiles

Create appropriate adjustment type profiles for your prepaid service agreements.

Service Agreement Types

You must create a prepaid SA Type for each unique CIS Division. Each SA Type must be set up as non-billable and requiring a characteristic premise. The SA Types must also indicate that they are used for prepaid by specifying a prepaid entity characteristic value.

Bill Cycle

Create at least one bill cycle that will be used for prepaid accounts. This bill cycle should not have a schedule defined.

Vendor Information

A vendor could be a vending station, payment agency or master station.

You must create a person for each of your vendors. Information about the vendor may include names, an address, phone numbers, etc.

Any other miscellaneous information about the vendor can be stored as characteristics on the person.

Hierarchical relationships between vendors can be established through characteristics. For instance, a vending station's person record may contain a foreign key characteristic that points to the master station that the vending station reports to.

Copyright © 2026, Oracle and/or its affiliates.

Defining Conservation Program Options

Oracle Utilities Customer Care and Billing allows you to define conservation (or energy efficiency) programs and provide rebates to customers.

- The Big Picture of Conservation Programs
- GL Accounting Example
- Setting Up Conservation Programs

The Big Picture of Conservation Programs

The purpose of using conservation programs is to provide rebates to customers based on eligibility and verification of newly purchased appliances and hardware that are rated to conserve the demand for energy. To redeem their rebates, customers have to submit a rebate application to the utility with receipts, and the utility has to administer and report on the programs.

Admin and transaction objects are provided in the product to support the definition of conservation programs (admin data) and the subsequent rebate claims (transaction data). Portals and BPA scripts are used to maintain the conservation programs and rebate claims. Adjustments are used to recognize the expense and to process refunds.

The following sections discuss the maintenance objects that support this functionality.

- Conservation Program Maintenance Object
- Conservation Program Rebate Definition Maintenance Object
- Rebate Claim Maintenance Object
- Rebate Claim Line Maintenance Object

Conservation Program Maintenance Object

A conservation program is an admin maintenance object (MO) used to support the definition of a conservation program. It holds rules that control how the claims for a conservation program are managed. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- A business object option Display Statistics Service Script is provided for business objects of this type. The script plugged into this option retrieves information displayed on the **Conservation Program Statistics Zone**.
- A business object option Display Statistics UI Map is provided for business objects of this type. This is the map used on the **Conservation Program Statistics Zone** to display statistics.
- A separate maintenance object is provided to capture rebate definitions for a conservation program

Copyright © 2026, Oracle and/or its affiliates.

- Logs are not provided.
- The standard characteristics collection is provided.

> **NOT_SUPPORTED**
>
> For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin**, and then **Database**, and then **Maintenance Object**, and then **Search** and view the MO C1-CPROG.

Conservation Program Rebate Definition Maintenance Object

A conservation program has rebate definitions that define the rebate amounts and energy savings for different types of products. The superset of such lines is the "rebate matrix". Each row in the matrix defines the rebate and presumed energy savings for a product purchased by the consumer. If your organization wishes to use this maintenance object (MO), you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- The Rebate Definition table can be used to capture information for rebate claim lines such as, appliance type, refund amount, energy savings, and applicable manufacturers and models.
- Logs are not provided for the Rebate Definition.
- A characteristics collection is not provided.

> **NOT_SUPPORTED**
>
> For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin**, and then **Database**, and then **Maintenance Object**, and then **Search** and view the MO C1-RDEF.

Rebate Claim Maintenance Object

When a customer files a claim for a rebate, a rebate claim will be created. The rebate claim maintenance object (MO) is used to support the definition of a claim. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- Depending on the utility's requirements, you can specify single or multiple items on the Rebate Claim. A separate maintenance object is provided to capture rebate claim lines for a rebate claim.
- The standard characteristics collection is provided.
- A log is provided. This can be used to track approval information for the claim, to capture adjustments created as the claim is processed, etc.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin**, and then **Database**, and then **Maintenance Object**, and then **Search** and view the MO C1-RCLAIM.

Rebate Claim Line Maintenance Object

A rebate claim has a rebate claim line for each product eligible for a refund. You use the rebate claim line maintenance object (MO) to create a rebate line. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- SA (service): The SA table is the real service SA. There is a separate SA under which the rebate adjustments are stored.
- Logs are not provided for Rebate Claim Line
- The standard characteristics collection is provided.

NOT_SUPPORTED

For more information, about this MO and to review the business objects defined for this MO, navigate to **Admin**, and then **Database**, and then **Maintenance Object**, and then **Search** and view the MO C1-REBLN.

GL Accounting Example

The following table shows the financial transactions that are issued when processing energy conservation rebates. This example shows the financial transactions when a customer files a claim for a dishwasher and insulation.

Notice how the expense adjustments are atomized:

- Separate adjustments are created to expense each item individually.
- A single adjustment is created for the A/P check request.

Event	GL Accounting
Rebate adjustment is created - dishwasher	Rebate Expense 35 A/P <35>
Rebate adjustment is created - insulation	Rebate Expense 45.45 A/P <45.45>
A/P check request issued	A/P 80.45 Cash <80.45>

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Conservation Programs

Conservation Programs allow administrators to create and maintain conservation programs for which customer can submit rebate claims. To set up a conservation program, open **Admin**, and then **Customer**, and then **Conservation Program**, and then **Add.**

> **NOT_SUPPORTED**
>
> For additional information on working with rebate claims, see Rebate Claims.

The topics in this section describe the base-package zones that appear on the Conservation Program portal.

- Conservation Program List Zone
- Conservation Program Zone
- Rebate Definition Zone
- Rebate Claim Statistics Zone

Conservation Program List Zone

The Conservation Program List zone lists every conservation program. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent conservation program.
- Click the Add link in the zone's title bar to add a new conservation program.

This zone displays the following for each conservation program defined in the system:

- Conservation Program Information (Description, Status, Start Date, End Date)
- The number of Complete Claims
- The total Complete Claims Amount
- The number of Incomplete Claims
- The Statistics Date/Time is the date that the statistics were last calculated

Use the following procedure to create a new conservation program:

- Select **Admin**, and then **Customer**, and then **Conservation Program**, and then **Add** or click **Add** on the title bar of the Conservation Program List zone.
- Enter basic information about the conservation program in the **Main** section, including:
  - Conservation Program code
  - Description
  - Start Date/End Date
  - Statistic Update Frequency (in hours)

Copyright © 2026, Oracle and/or its affiliates.

- Enter **Financial Information** about the conservation program, including:
  - CIS Division
  - SA Type
  - A/P Adjustment Type
  - A/P Taxable Adjustment Type
- Enter **Approval Information** about the conservation program, including:
  - Approval To Do Type
  - Duplicate Claim To Do Role
  - Approval Levels and Threshold Amounts
- Click **Save.** To return to the Conservation Program portal without saving the new program, click **Cancel.**

To view a specific conservation program, click the broadcast icon for the conservation program you wish to view. The remaining zones in the Conservation Program portal open displaying details about the selected conservation program.

Conservation Program Zone

You use the Conservation Program zone to view and maintain individual conservation program. The Conservation Program zone displays the following information about the selected conservation program:

- Basic information about the conservation program, including
  - Conservation Program
  - Description
  - Start Date/End Date
  - Statistic Update Frequency (in hours)
  - Status
- Financial Information about the conservation program, including
  - CIS Division
  - SA Type
  - A/P Adjustment Type
  - A/P Taxable Adjustment Type
- Approval Information about the conservation program, including
  - Approval To Do Type
  - Duplicate Claim To Do Role
  - Approval Levels and Threshold Amounts

Please see the zone's help text for information about this zone's fields.

- Conservation Program Status
- Conservation Program Actions
- Editing Conservation Programs

Copyright © 2026, Oracle and/or its affiliates.

- Deleting Conservation Programs
- Activating Conservation Programs
- Deactivating Conservation Programs
- Setting the Status of Conservation Programs to Pending

Conservation Program Status

The Status of a conservation program indicates the current state of the program within the system. Valid statuses include:

- **Pending** indicates the program is pending. This is the initial state of a conservation program when first created.
- **Active** indicates the conservation program is currently active.
- **Inactive** indicates the conservation program is currently inactive.

Conservation Program Actions

You can perform a number of actions on a conservation program, including:

- **Edit:** Used to edit a conservation program
- **Delete:** Used to delete a conservation program
- **Activate:** Used to activate a pending or inactive conservation program
- **Refresh Statistics:** Used to refresh the statistics of an active conservation program
- **Deactivate:** Used to deactivate a pending or active conservation program
- **Pend:** Used to change the status of an active or inactive conservation program to Pending

The actions available are based on the current status of the conservation program. The table below summarizes the actions available at each status.

Status	Valid Action
Pending	Edit, Delete, Activate, Deactivate
Active	Edit, Refresh Statistics, Deactivate, Pend
Inactive	Activate, Pend

Editing Conservation Programs

Use the following procedure to edit an active or pending conservation program:

- Click **Edit.**
- Edit the details of the conservation program as needed.
- Click **Save.**

Copyright © 2026, Oracle and/or its affiliates.

Deleting Conservation Programs

Use the following procedure to delete a conservation program:

- Click **Pend** to change the status of the conservation program to Pending (if needed).
- Click **Delete.**
- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the conservation program, click **Cancel.**

Activating Conservation Programs

To activate an inactive or pending conservation program, click **Activate.**

Deactivating Conservation Programs

To deactivate an active or pending conservation program, click **Deactivate.**

Setting the Status of Conservation Programs to Pending

To set the status of an active or inactive conservation program to Pending, click **Pend.**

Rebate Definition Zone

You use the Rebate Definition zone to add, view, and edit rebate definitions associated with a conservation program. A rebate definition defines the types of rebates allowed for a specific conservation program. The Rebate Definition zone displays the following details for each claim line:

- Rebate Definition
- Status (Active or Inactive)
- Rebate Amount per Unit
- Icons to Edit, Delete, and Activate/Deactivate rebate definitions
- Adding Rebate Definitions
- Editing Rebate Definitions
- Deleting Rebate Definitions
- Activating Rebate Definitions
- Deactivating Rebate Definitions

Copyright © 2026, Oracle and/or its affiliates.

Adding Rebate Definitions

Use the following procedure to add a rebate definition:

- Click **Add** in the title bar of the Rebate Definition zone.
- Enter a Description of the rebate definition.
- Select the **Product** for the rebate definition from the dropdown list.
- Select the **Service Type** for the rebate definition from the dropdown list.
- Enter the **Rebate Per Unit Amount** for the rebate definition.
- Enter the **Rebate Unit of Measure** for the rebate definition.
- Select the **Expense Adjustment Type** for the rebate definition from the dropdown list.
- Enter the **Presumed Energy Savings** for the rebate definition.
- Select the **Presumed Energy Savings** Unit of Measure for the rebate definition from the dropdown list.
- Enter the **Manufacturer** and **Model** for each item eligible to be submitted as a rebate claim line.
- To upload manufacturer/model information from a comma-separated-values file, click **CSV File to Upload.**
  - Click **Browse** on the File Upload dialog, and browse to the file to be uploaded.
  - Click **Upload.**
- Click **Save.** To return to the Rebate Definition zone without adding the rebate definition, click **Cancel.**

Editing Rebate Definitions

Use the following procedure to edit a rebate definition:

- Click the edit icon for the rebate definition you wish to delete.
- Edit the details of the claim line as appropriate.
- Click **Save**. To return to the Rebate Definition zone without changing the rebate definition, click **Cancel.**

> **Note**
>
> You can edit a rebate definition only when it is Active.

Deleting Rebate Definitions

Use the following procedure to delete a rebate definition:

Copyright © 2026, Oracle and/or its affiliates.

- Click the delete icon for the rebate definition you wish to delete.
- Click **OK** on the Confirm Delete dialog. To close the dialog without deleting the rebate definition, click **Cancel.**

Activating Rebate Definitions

To activate an inactive rebate definition, click the **Activate** button for the rebate definition you wish to activate.

> **Note**
>
> You can only activate rebate definition that is currently Inactive.

Deactivating Rebate Definitions

To deactivate an active rebate definition, click the **Deactivate** button for the rebate definition you wish to deactivate.

> **Note**
>
> You can only deactivate rebate definition that is currently Active.

Rebate Claim Statistics Zone

You use the Rebate Claim Statistic zone to view statistics for submitted rebate claims based on the current conservation program. This zone displays the following:

- Conservation Program
- Statistics Date/Time
- Statistics Charts
- Statistics Graphs

Statistics Graphs

This zone displays statistics for submitted rebate claims using the following charts:

- **Statistics by Status:** A pie chart that displays the number and percentage of claims for each status.
- **Statistics by Month:** A line chart that displays the number of claims completed each month.

Copyright © 2026, Oracle and/or its affiliates.

- **Statistics by Product:** A pie chart that displays the total value, percentage, and number of claim lines completed for each product.
- **Statistics by Claim Age:** A line and bar chart that displays the number of new claims per month (line graph) and the number of claims per age grouping (bar graph).

To refresh statistics, click **Refresh Statistics** on the title bar of the Rebate Claim Statistics zone.

Defining Open Market Options

- Markets
- Market Participants

Markets

- Understanding Market
- Configuring Market

Understanding Market

The Market object is used to define a deregulated market which is a geographical area operating with a set of rules to manage premises and/or customers for a service type. Separate market instances are created for each geographic area and service type, e.g. Gas New York (GNY), Electric New York (ENY), Electric United Kingdom (EUK).

Each deregulated market has participants (Service Providers) operating within that market. Different types of Service Providers are grouped into Market Relationship Types. There are Service Providers that manage the customer at a premise - e.g. Local Distribution Company, Energy Service Provider (Retailer, Supplier) and some that manage the meter at a premise (Meter Operator, Data Aggregator). Service Providers can operate across several deregulated markets (e.g. Retailers) and they can enter and exit markets independently. Start and End dates are provided to manage Service Providers entering and exiting markets.

Service Provider types can be managed by the market object and others can be managed by Market Provider Configuration. Recommended Market Relationship types managed by the Market object are Meter Owners, Meter Aggregators. There may be markets where no Service Providers are managed by the Market object, even where Meter Owners are present. For example when Meter Owners don’t change.

Service Provider types managed by the Market Provider Configuration are recommended for Retailer and Distributor Market Relationship Types. This option provides greater flexibility for market transactions to be sent to service providers based on the market and billing options.

> **Note**
>
> Refer to Market Provider Configuration for Service Providers that are managed for each Market Billing Option.

Copyright © 2026, Oracle and/or its affiliates.

Markets that manage a provider of last resort or fall back service provider will configure the market Fall Back Service Provider, the start and end dates allow a solution to determine who the current Fall Back Service Provider is. Not all markets have a Fall Back Service Provider.

Configuring Market

This portal is used to display and maintain a Market.

Refer to Understanding Market for more information.

To maintain markets, open **Admin**, and then **Open Market**, and then **Market**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Market Participants

- Understanding Market Participants
- Configuring Market Participants

Understanding Market Participants

Market Participants are participants in a deregulated environment. Relationships between market participants are defined in a particular market record. Refer to Understanding Markets for more information.

Each market participant can be associated to an external system which is used to define the messages that can be sent to that market participant and how each message is sent.

Configuring Market Participants

This portal is used to display and maintain market participants.

Refer to Understanding Market Participants for more information.

You can access the portal by selecting **Admin**, then **Open Market**, then **Market Participant**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Participants List:** This zone lists all market participant records. Broadcast a record to display the details of the selected record.
- **Market Participant:** This zone provides information about the selected market participant.
- **Market Participant Characteristics**: This zone lists characteristics defined for the market participant.
- **Market Contracts**: This zone lists market contracts where the market participant is either the Buyer or Seller.
- **Processing Method List:** This zone provides the list of processing methods defined for the market participant.
- **Translation Method List:** This zone provides the list of translation methods defined for the market participant.

Copyright © 2026, Oracle and/or its affiliates.

- **Inbound BOs Send By Service Provider:** This zone lists inbound business objects that are sent by this market participant. The identification is driven by the business object having a Business Object Option of type "Sent By Service Provider" that references the current market participant.

Defining Consumer Contract Options

The topics in this section describe how to set up the system to enable consumer contract functionality.

- Understanding Consumer Contracts
- Understanding Consumer Products
- Configuring Consumer Contract Type
- Configuring Consumer Contract Milestone Type
- Configuring Consumer Product
- Configuring Consumer Product Version
- Configuring Consumer Product Rule
- Configuring Consumer Product Component
- Configuring Consumer Contract Rule

Understanding Consumer Contracts

Consumer contracts record customer relationships. These relationships can encompass multiple services, multiple sites and multiple accounts. For example a consumer contract can be for a customer's gas and electricity supply at a single site or for a customer’s water supply at multiple sites.

A consumer contract can relate to one or more consumer product versions. A consumer product version records values that are common to all consumer contracts related to that consumer product version. A consumer product version also records rules that are applied by customer service requests to set up a new consumer contract or change an existing consumer contract. These rules can for example create service agreements or stop service agreements. Refer to Understanding Consumer Products for more details. A consumer contract’s consumer contract products record the links between the consumer contract and consumer product versions.

Note a consumer contract relates to consumer product versions not to consumer products. The consumer product version linked to a consumer contract is the version of the required consumer product that is applicable for that consumer contract. This could for example be the consumer product version that is effective for the date the consumer contract was agreed.

Consumer contracts group together service agreements. The relationship between a consumer contract and a service agreement is recorded in the service agreement by linking the service agreement to one of the contract’s consumer contract products. A consumer contract can record details that are common for all of the related service agreements.

All consumer contracts have a consumer contract type. A consumer contract type limits which consumer products are valid for a consumer contract of that type. Note the consumer products that are eligible are further limited by the consumer product's eligibility criteria. Refer to Understanding Consumer Products for more details.

Copyright © 2026, Oracle and/or its affiliates.

A consumer contract can have milestones where a milestone is a date that is significant to the contract. Milestones can optional trigger an action. Refer to Understanding Consumer Contract Milestones for more details.

- Understanding Consumer Contract Milestones

Understanding Consumer Contract Milestones

Dates that are significant to a consumer contract can be recorded as consumer contract milestones. A consumer contract milestone can optionally trigger an action. For example, a consumer contract milestone could record the date when a fixed-priced period ends and trigger a change to the consumer contract when this date is reached. Another example is a consumer contract milestone that sends a notification to the customer one month prior to the end of the fixed-priced period.

A consumer contract milestone can relate to one or more of the contract's consumer contract products. A contract's consumer contract products record the links between the consumer contract and consumer product versions. In the example above of a consumer contract milestone recording the date when a fixed price period ends, the consumer contract milestone would be related to the consumer contract period whose consumer product version has the fixed priced period.

Each milestone has a consumer contract milestone type. The consumer contract milestone type indicates whether or not a milestone is executable, that is whether or not it triggers an action. For executable milestones, the action to be performed is derived from the consumer contract milestone type.

A consumer contract milestone's executable status can be: Pending, Processed, Not Executable or Canceled. Not Executable indicates that the milestone does not have a related action and Canceled indicates that the milestone is no longer applicable.

The Perform Consumer Contract Milestone Actions (C1–CCMP) batch process executes a consumer contract milestone's actions. This batch process selects the consumer contract milestones whose milestone date is today or in the past and whose executable status is Pending. The batch process performs the **Perform Action** algorithms defined on the milestone's Consumer Contract Milestone Type business object. Once a consumer contract milestone's actions have been performed, the batch process changes its executable status to Processed.

> **Note**
>
> This batch process is built using the Plug-in Driven Background Processes framework. Your implementation can introduce alternate logic, by duplicating this batch control and replacing the select and/or process records algorithms. Adjust the batch parameters according to your custom logic, if needed.

The base product provides a **Contract Milestone Type (BO) – Perform Action** algorithm that sends customer notification. Additional algorithms may exist for your implementation. To see the available algorithm(s), search for algorithm types of the **Contract Milestone Type (BO) – Perform Action** algorithm entity in the application.

Understanding Consumer Products

Copyright © 2026, Oracle and/or its affiliates.

Consumer products record attributes and rules that are used for consumer contracts. A consumer product can for example record attributes for:

- The duration of a fixed-priced period
- The amount of a contractual change, such as an exit fee
- Prices

These attributes can be used when setting up a new consumer contract or changing an existing contract. They could also be used to display details when a product is being selected or to include details in correspondence with customers.

A consumer product's rules are used to set up new consumer contracts and to change existing consumer contracts.

A consumer product's attributes and rules are recorded in the consumer product's version. Refer to Understanding Consumer Product Versions for more details.

Consumer product eligibility criteria can be used to determine which consumer products that are eligible in particular circumstances. For example a consumer product could be eligible only when autopay is being used. Consumer product eligibility criteria are optional. If a consumer product has eligibility criteria then the consumer product is eligible or not depending on the criteria. If however a consumer product does not have eligibility criteria then the consumer product is always eligible. The next sections provide more information on this functionality.

- Understanding Consumer Product Eligibility Criteria
- Understanding Consumer Product Versions
- Understanding Consumer Product Rules
- Understanding Consumer Product Components
- Understanding Consumer Product Component Eligibility Criteria
- Understanding Consumer Contract Rules

Understanding Consumer Product Eligibility Criteria

A consumer product eligibility criteria rule requires the following components:

- Operand to be compared
- Comparison operator
- Comparison value
- Comparison results

The operands to be compared by consumer product eligibility criteria rules are identified using characteristic types. The characteristic types must have a characteristic entity of **Consumer Product Eligibility Operand**. The value of these characteristics are populated by the Operands algorithms that are defined on the related Consumer Contract Type's business object. The base product provides **Consumer Contract Type (BO) – Operands** algorithm that sets a characteristic value to indicate if the customer is or will be on autopay. Additional algorithms may exist for your implementation. To see the available algorithm(s), search for algorithm types of the **Consumer Contract Type (BO) – Operands** algorithm entity in the application.

The comparison operator can be one of the following: <, <=, <>, =, >, >=, Between, In or Is Blank.

The comparison value is recorded in the consumer product eligibility criteria rule.

Copyright © 2026, Oracle and/or its affiliates.

The comparison results define what action should be taken when the criteria rule is applied. If the comparison is true or false, the result can be that the consumer product is eligible, the consumer product is not eligible or the next eligibility rule should be checked. If there is insufficient data, that is the operand to be compared was not found, the result can be that the consumer product is eligible, the consumer product is not eligible, the next eligibility rule should be checked or an error should be generated.

Understanding Consumer Product Versions

A consumer product version records the product attributes and rules that are applicable for consumer contracts related to the consumer product version.

> **Note**
>
> Consumer contracts relate to consumer product versions not to consumer products. The addition of a new consumer product version will not affect existing consumer contracts.

For a consumer contract that is created or changed using a customer service request, the consumer product version is determined using the Determine Version algorithm defined on the Consumer Product business object. The base package provides an algorithm type that determines the effective consumer product version by comparing the version's offer effective date and the date the new contract or contract change was agreed. Alternative custom algorithms could use one or more of the consumer product version's offer effective date, service effective date, offer end date, service end date and consumer product version selection. To see the list of available algorithm(s), search for algorithm types of the **Consumer Product (BO) – Determine Version** algorithm

Consumer products have two kinds of rules: consumer product rules and consumer contract rules. Consumer product rules are used to perform configurable services for a consumer product version. For example, a consumer product rule can determines the fallback product to use when a contract's product is no longer eligible. Consumer contract rules are used to set up new consumer contracts and make changes to existing consumer contracts. For example, a consumer contract rule can be used to create or stop a service agreement.

Consumer product rules relate directly to a consumer product version. Consumer contract rules are organized into groups using consumer product components. The consumer product components can be linked together into a hierarchy. The consumer product version records the consumer product component at the top of the hierarchy.

A consumer product version's attributes are recorded either as consumer product version characteristics or as consumer product component characteristics. Examples of attributes are: the duration of a fixed-price period and the amount of an exit fee.

Consumer product component eligibility criteria determine whether a consumer product component is applicable. The product attributes retrieved or consumer contract rules applied for a consumer product version will vary depending on the consumer product component eligibility criteria rules.

Understanding Consumer Product Rules

Consumer product rules are standard and custom rules that perform actions for a consumer product version. A consumer product rule class defines the type of action that is performed. A consumer product rule class is recorded in the consumer product version for each related

Copyright © 2026, Oracle and/or its affiliates.

consumer product rule. The base product provides the following consumer product rule classes:

- Determine Fallback Product
- Determine If Reapplication Required

A consumer product rule typically includes the following details:

- Basic information about the consumer product rule, including its name, description and detailed description. This is standard information for most consumer product rules.
- Specific parameters used by the consumer product rule.

The base product provides the following consumer product rules:

- **Determine Fallback Product**: A rule that determines the fallback product using a consumer product class recorded in the rule.
- **Determine if Product should be Reapplied:** A rule that determines if reapplication is necessary by deriving the current consumer product component eligibility criteria operands for the consumer product and comparing these with the previous operand values.

Additional algorithms may exist for your implementation. To see the available algorithm(s), search for algorithm types of the **Consumer Product Rule (BO) – Apply Rule** algorithm entity in the application.

Understanding Consumer Product Components

Consumer product components group together product attributes and consumer contract rules.

Consumer product component eligibility criteria rules can be created for a consumer product component. These eligibility rules will determine if a consumer product component's product attributes are applicable and if a consumer product component's consumer contract rules are applied. A consumer product version could, for example, have one consumer product component that is eligible when a customer is on autopay and another consumer product component that is eligible when a customer is not on autopay.

A consumer product component can, optionally, have one or more child consumer product components. This allows consumer product components to be organized into a hierarchy. If a consumer product component is not eligible, any child consumer product components will also not be eligible.

Understanding Consumer Product Component Eligibility Criteria

A consumer product component eligibility criteria rule requires the following components:

- Operand to be compared
- Comparison operator
- Comparison value
- Comparison results

The operands to be compared by consumer product component eligibility criteria rules are identified using characteristic types. The characteristic types must have a characteristic entity of **Consumer Product Comp Eligibility Operand**. The value of these characteristics are populated by the **Contract Operands** and **Location Operands** algorithms defined on the Consumer Product business object. The base product provides a **Consumer Product (BO) – Contract Operands** algorithm type that sets a characteristic value to indicate if the customer is

Copyright © 2026, Oracle and/or its affiliates.

or will be on autopay. Additional algorithms may exist for your implementation. To see the available algorithm(s), search for algorithm types of the **Consumer Product (BO) – Contract Operands** algorithm entity in the application.

The comparison operator can be one of the following: <, <=, <>, =, >, >=, Between, In or Is Blank.

The comparison value is recorded in the consumer product component eligibility criteria rule.

The comparison results define what action should be taken when the criteria rules is applied. If the comparison is true or false, the result can be that the consumer product component should be applied, the consumer product component should not be applied or the next eligibility rule should be checked. If there is insufficient data, that is the operand to be compared was not found, the result can be that the consumer product component should be applied, the consumer product component should not be applied, the next eligibility rule should be checked or an error should be generated.

Understanding Consumer Contract Rules

Consumer contract rules are standard and custom rules that perform actions to set up a new consumer contract or change an existing consumer contract which is related to the consumer product version. A consumer contract rule could for example create a new service agreement.

Consumer contract rules are grouped together by consumer product components. Consumer product components determine if and when a consumer contract rule will be applied. Refer to Understanding Consumer Product Components for more details.

Consumer contract rules are applied by a customer service request when a consumer contract is being created or changed. Different consumer product rules can be applied at different stages of the customer service request and for different changes. The consumer product component defines a consumer contract rule usage that determines when a consumer contract rule is applied.

The base product provides the following consumer contract rule usages:

- On Activation – Add: The consumer contract rules are applied during activation set up for the consumer product versions of consumer contract products that have been added.
- On Activation - Remove: The consumer contract rules are applied during activation set up for the consumer product versions of consumer contract products whose end date time has been set.
- On Completion – Add: The consumer contract rules are applied at completion for the consumer product versions of consumer contract products that have been added.
- On Completion – Remove: The consumer contract rules are applied at completion for the consumer product versions of consumer contract products whose end date time has been set.
- Service Location - Add: The consumer contract rules are performed when any action required to add a service point or registration point to a contract, such as field work or a market process, has been performed.
- Service Location – Remove: The consumer contract rules are performed when any action required to remove a service point or registration point from a contract, such as field work or a market process, has been performed.

The base product provides **Consumer Contract Rule (BO) – Apply Rule** algorithms that add a service agreement and that add a consumer contract milestone. Additional algorithms may

Copyright © 2026, Oracle and/or its affiliates.

exist for your implementation. To see the available algorithm(s), search for algorithm types of the **Consumer Contract Rule (BO) – Apply Rule** algorithm entity in the application.

Configuring Consumer Contract Type

This portal is used to display and maintain a Consumer Contract Type.

A Consumer Contract Type defines the Consumer Products that are valid for the type of contract. It also defines the information that is used in determining product eligibility.

Refer to Understanding Consumer Contracts for more information.

To maintain consumer contract types, open **Admin**, and then **Consumer Contract**, and then **Consumer Contract Type**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Configuring Consumer Contract Milestone Type

This portal is used to display and maintain a Consumer Contract Milestone Type.

A Consumer Contract Milestone Type defines milestone dates that can be recorded and whether an action should be performed for the type of milestone.

Refer to Understanding Consumer Contract Milestones for more information.

To maintain consumer contract milestone types, open **Admin**, and then **Consumer Contract**, and then **Consumer Contract Milestone Type**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Configuring Consumer Product

This portal is used to display and maintain a Consumer Product, including the consumer product's versions and eligibility rules.

Refer to Understanding Consumer Products for more information.

To maintain consumer products, open **Admin**, and then **Consumer Contract**, and then **Consumer Product**.

This is a standard All-In-One Portal.

Refer to the Consumer Product zone's embedded help text for more information.

This portal also shows zones for Consumer Product Versions and Consumer Product Eligibility.

Configuring Consumer Product Version

Copyright © 2026, Oracle and/or its affiliates.

This portal is used to display and maintain a Consumer Product Version.

Refer to Understanding Consumer Product Versions for more information.

To maintain consumer product versions, open **Admin**, and then **Consumer Contract**, and then **Consumer Product Version**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Configuring Consumer Product Rule

This portal is used to display and maintain a Consumer Product Rule.

Refer to Understanding Consumer Product Rules for more information.

To maintain consumer product rules, open **Admin**, and then **Consumer Contract**, and then **Consumer Product Rule**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Configuring Consumer Product Component

This portal is used to display and maintain a Consumer Product Component, including child consumer product components, eligibility rules and related consumer product versions.

Refer to Understanding Consumer Product Components for more information.

To maintain consumer product components, open **Admin**, and then **Consumer Contract**, and then **Consumer Product Component**.

This is a standard All-In-One Portal.

Refer to the Consumer Product Component zone's embedded help text for more information.

This portal also shows zones for Consumer Product Component Eligibility and Related Consumer Product Versions.

Configuring Consumer Contract Rule

This portal is used to display and maintain a Consumer Contract Rule.

Refer to Understanding Consumer Contract Rules for more information.

To maintain consumer contract rules, open **Admin**, and then **Consumer Contract**, and then **Consumer Contract Rule**.

This is a standard All-In-One Portal.

Refer to the embedded help text for more information.

Copyright © 2026, Oracle and/or its affiliates.

Configuration

This section contains additional configuration topics for Oracle Utilities Customer to Meter and Oracle Utilities Customer Care and Billing.

- Configuring Zones
- Content Migration Assistant (CMA) Addendum
- To Do Lists Addendum
- Background Processes Addendum
- Security Addendum
- Defining Converted COBOL Program Options
- Advanced Search Options

Configuring Zones

Many zones in Oracle Utilities Customer Care and Billing do not require configuration by your implementation team. For example, the base package is shipped with the Account Financial History zone that appears on the Control Central - Account Information portal. This zone does not require configuration because its zone type has no configurable options (i.e., its behavior is static).

Other zones require configuration before they can be used because their behavior is dynamic. The topics in this section provide tips and techniques on how to configure zones in Oracle Utilities Customer Care and Billing.

NOT_SUPPORTED

Refer to The Big Picture of Portals and Zones in the *Oracle Utilities Application Framework Administration Guide* for a description of portal and zone functionality.

- Configuring Timeline Zones

Configuring Timeline Zones

A timeline zone is a zone that may be configured on the account or customer information tabs of Control Central. This type of zone contains one or more "lines" where each line shows when significant events have occurred. For example, you can set up a timeline zone that has two lines: one that shows when payments have been received from a customer, and another that shows when bills have been sent to the customer.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

For a complete description of the numerous features available on a timeline zone, refer to Timeline Zone`.

Configuration is required to enable a timeline zone. A timeline zone is made up of one or more algorithms where each algorithm returns “events” for a certain type of record (payment or bill for example). The product delivers several algorithm types for the timeline zone to include “events” for many different transactions in the product. Each algorithm type uses the Account or Person in context to retrieve the related information.

- Refer to Configuring Timeline Zones in the framework portion of the administration guide for information about the type of detail each event may return. The base provided algorithm types include many parameters to allow the implementation to configure much of this information as desired.

Many of the algorithms support supplying a BPA script to launch when a user clicks on an event on a timeline. Be sure to review each algorithm type to see what types of events support a BPA script. For each event, determine if your business practice warrants the development of a BPA script.

After configuring the algorithms per your business needs, the following additional steps are needed:

- Set up a zone that references these algorithms. The zone will reference the **F1-TIMELINE** zone type.
- Link the zone to the appropriate portal(s) (e.g., Control Central - Account Information or Control Central - Customer Information).
- Update your users' portal preferences and security rights so they can see the zone in the desired location on the portal(s).

You can set up many timeline zones. For example,

- You might want different zones to appear on a portal depending on the type of user. For example, you might want one timeline for billing clerks, and a different one for customer service representatives.
- For aesthetic reasons, you might want multiple simple timeline zones to appear on a given portal rather than one complex timeline zone.
- You might want to set up context specific timeline zones. For example, you might want to have one timeline zone that is premise-oriented and another that is person-oriented.

Content Migration Assistant (CMA) Addendum

This section is an addendum to the general Content Migration Assistant section in the *Oracle Utilities Application Framework Administration Guide*. This section assumes that you are familiar with the concepts of Content Migration Assistant (CMA).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> CMA is designed to migrate configuration data only. Since base product data should be updated only through product patching mechanisms, CMA cannot currently be used to migrate master/transactional data that contains system-generated primary keys, such as customer account or billing information.

This addendum describes Content Migration Assistant (CMA) functionality that is specific to Oracle Utilities Customer Care and Billing.

The following sections provide information about what is provided in the C1-owned base package:

- Base Package Migration Plans
- Base Package Migration Request
- Wholesale and Piecemeal Migrations
- Processing Notes for Specific Objects

Base Package Migration Plans

The C1-owned base package provides a large number of migration plans to support migrating configuration and/or administration data from one environment to another.

Use the following procedure to access the base package migration plans:

1. Navigate to **Admin**, and then **Implementation Tools**, and then **Migration Plan**, and then **Search**.
2. Enter “C1” in the **Migration Plan** field.
3. **Click Refresh**.
4. Select a migration plan in the search results list. The details of the selected migration plan are displayed in the Migration Plan portal.

> **Note**
>
> The C1-owned base package plans can also be used as a basis for custom migration plans. To create a custom migration plan, select a plan to base the custom migration plan on, click the Duplicate button, and define the custom plan to meet the implementation’s requirements as described in Defining a Migration Plan in the *Oracle Utilities Application Framework Administration Guide*.

Base Package Migration Request

The C1-owned base package provides two migration request to support migrating configuration and/or administration data from one environment to another.

Use the following procedure to access the base package migration requests:

1. Navigate to **Admin**, and then **Implementation Tools**, and then **Migration Request**, and then **Search**

Copyright © 2026, Oracle and/or its affiliates.

2. Enter "C1" in the **Migration Request** field.
3. Click **Refresh.**
4. Click the **Copy and Sync Control Data** link in the search results list.

The details of the migration requests are outlined below. Please use the Migration Request portal to view additional details about this migration request.

Base Admin Data Migration Request

- **Migration Request:** C1-AdminBasic
- **Description:** Base Admin Data
- **Detailed Description:** Base Admin Data.
- **Migration Plans:** This migration request includes base package migration plans used by Oracle Utilities Customer Care and Billing.

Copy and Sync Control Data Migration Request

- **Migration Request:** C1-CopyControlTables
- **Description:** Copy and Sync Control Data
- **Detailed Description:** Copy and Sync Control Data.
- **Migration Plans:** This migration request includes base package migration plans used by Oracle Utilities Customer Care and Billing.

For additional information about these migration plans, see Base Package Migration Plans.

> **Note**
>
> The C1-owned base package request can also be used as a basis for custom migration requests. To create a custom migration request, select the base package request, click the Duplicate button, and define the request as to meet the implementation's requirements as described in Defining a Migration Request in the *Oracle Utilities Application Framework Administration Guide*.

Wholesale and Piecemeal Migrations

There are two general types of migrations used with the Content Migration Assistant: wholesale migrations and piecemeal migrations.

- Wholesale Migrations
- Piecemeal Migrations

Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or administration data from one environment to another. For example, a wholesale migration might be used when migrating admin data from a development or test environment to a production environment. For more on this type of migration, see Types of Migrations in the *Oracle Utilities Application Framework Administration Guide*.

The following is a high-level overview of the steps involved when executing a wholesale migration.

Copyright © 2026, Oracle and/or its affiliates.

1. Process the “F1-SchemaAdmin” (FW Foundation) migration request (This request contains migration plans for Field, Lookup, Char Type, Currency Code and FK Ref).
2. Process the “C1-AdminBasic” migration request. This includes copies of framework migration plans (including plans for Business Objects, Algorithms, and Feature Configurations) from the “F1-FrameworkAdmin” migration request, as well as independent base package C1-owned wholesale migration plans which can be run first.
3. Process any of the other delivered framework-based (F1-owned) migration requests as needed (except for the “F1-FrameworkAdmin” migration request which is already incorporated in #2)
4. Process the “C1-CopyControlTables” migration request. This includes base package C1-owned wholesale migration plans with dependency on “C1-AdminBasic” migration request.

Piecemeal Migrations

Piecemeal (or "non-wholesale" or “targeted”) migrations are used when migrating a small portion (or piece) of configuration and/or administration data from one environment to another. For example, a piecemeal migration might be used when migrating groups and rules from a development or test environment to a production environment. For more on this type of migration, see Types of Migrations in the *Oracle Utilities Application Framework Administration Guide*.

The C1-owned base package does not contain piecemeal migration plans for Oracle Utilities Customer Care and Billing.

Processing Notes for Specific Objects

The following limitations apply to certain Oracle Utilities Customer Care and Billing objects when using Content Migration Assistant:

- An Issuing Center object references a User. If this user does not exist in the target system, Content Migration Assistant cannot apply the requested changes.
- A Case Type object references an Application Service. If this service does not exist in the target system, Content Migration Assistant cannot apply the requested changes.
- Collection Agency and Service Provider objects reference a Person. If this person does not exist in the target system, Content Migration Assistant cannot apply the requested changes.
- Service Provider and Tender Source objects reference a Service Agreement. If this service agreement does not exist in the target system, Content Migration Assistant cannot apply the requested changes.
- If your migrateable object includes log tables, you may add your log entity to the characteristic type F1-MGO. You should also mark the log table as a “Non-Migrateable Table” in the Maintenance Object options.
- Transactions are applied in an unspecified order (and probably in order numerically by a randomly generated ID value). CMA only looks at "hard" constraints when determining what to put into the same transaction. Any "soft" constraints such as characteristics and algorithm parameters that might have FK references to other objects are not processed by CMA. Unless the migration plan ensures that related items go into the same transaction, they will end up in different ones, and must need to apply again and again until eventually all gets applied.

Copyright © 2026, Oracle and/or its affiliates.

To Do Lists Addendum

This section is an addendum to the general To Do Lists chapter. This addendum describes the To Do functionality that is specific to Oracle Utilities Customer Care and Billing.

- Assigning A To Do Role

Assigning A To Do Role

As described in To Do Entries Reference A Role, each To Do entry requires a role. To Do entries created in Oracle Utilities Customer Care and Billing may attempt to assign a role based on an account management group or division if it is applicable to the type of data related to the To Do entry.

As described in The Big Picture of To Do Lists, users are informed that something requires their attention by entries that appear in a To Do List. For example, consider what happens when billing can't find a reading (and it's not allowed to estimate):

- The billing process creates a bill segment that is in error (meter read cannot be found).
- This bill segment that's in error, in turn, triggers the creation of a To Do entry.
- The To Do entry is assigned a role. A role is one or more users who can look at / work on the entry.
- When users view a To Do List, they only see entries addressed to roles to which they belong.

You can optionally use account management groups (AMG) to define the respective role to be assigned to To Do entries that are associated with an account and To Do type. For example, you can create an AMG called Credit Risks and assign this to accounts with suspect credit. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the Credit Risks AMG. Refer to Setting Up Account Management Groups for more information.

By assigning an AMG to an account, you are telling the system to address this account's To Do list entries to the roles defined on the AMG (note, each To Do type can have a different role defined for it on an AMG).

You can optionally use division to define the respective role to be assigned to To Do entries that are associated with an account and To Do type. For example, you may have a division called California Operations and assign this to accounts located in California. Then, whenever an account-oriented To Do entry is created for such an account, it will be assigned a role based on the California Operations division. Refer to Setting Up CIS Divisions for more information.

A To Do Pre-Creation installation options plug-in is provided to determine the appropriate To Do Role for an account based on AMG and division setup. If plugged in, the logic to determine To Do role for an account is performed whenever a To Do entry is created. Refer to C1-TDCR-DFRL for further details on how this plug-in works.

Copyright © 2026, Oracle and/or its affiliates.

NOT_SUPPORTED

Refer to To Do Entries Reference A Role for the details of how an initial role is assigned to To Do entries.

Background Processes Addendum

This chapter is an addendum to the general Defining Background Processes chapter. This addendum describes the background processes that are provided with Oracle Utilities Customer to Meter.

- How To Set Up A New Extract Processes
- Choosing Which Records Are Selected
- The Big Picture of Sample & Submit

How To Set Up A New Extract Processes

Several background processes delivered with the system are used to interface information out of the system. The topics in this section describe when and how to introduce an additional extract process.

- Setting Up Meter Read Extracts
- Setting Up Automatic Payment Extracts

Setting Up Meter Read Extracts

You will need a meter read extract for every mechanism your company uses to route meter read requests to the software that handles your meter reading requests. For example:

- You will need a meter read extract to interface records to your handheld device software. The MDL process delivered with the system is intended to be used to handle this function. This process will have to be populated to format the output records in keeping with the needs of your meter reading software.
- If you interface some meter read requests to automatic meter reading software, you will need a new meter read extract process.

If you need additional meter read extract processes, set up the following information:

- Add a new batch control record. Populate the fields as follows:
- **ID.** Assign an easily recognizable unique ID for the meter read extract process.
- **Description.** Enter a description of the meter read extract process.
- **Accumulate All Instances.** Turn this switch on.
- Use Route Type to define the meter read extract process to be used for each route type.
- It may be necessary to register the process with your scheduler software.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up Automatic Payment Extracts

You will need an automatic payment extract for every mechanism your company uses to route automatic payment requests to a financial institution / clearing house. For example:

- You will need an automatic payment extract to interface records to the Automated Clearing House (ACH) if you allow customer to pay via credit card or direct debit from a checking account. The **APAYACH** and C1-APACH processes delivered with the system are intended to be used to handle this function.

If you need additional automatic payment extract processes, set up the following information:

- Add a new batch control record. Populate the fields as follows:
- **Batch Process.** Assign an easily recognizable unique ID for the automatic payment extract process.
- **Description.** Enter a description of the automatic payment extract process.
- **Accumulate All Instances.** Turn this switch on.
- Use Auto Pay Route Type to define the auto pay extract process to be used for each route type.

Choosing Which Records Are Selected

The system is delivered with some background processes that utilize the **Select Records** plug-in spot to select the records to be processed. It is important to note that these background processes do not utilize the Process Records plug-in spot. This logic is hardcoded as these background processes are some of the core system functions such as billing and collections. This functionality provides flexibility in choosing which records are processed, but not how they are processed.

The background processes delivered with the system optionally selects records based on CIS Division. One example is the batch control (BILLING) and algorithm (C1-BILL-SR). The background process and select records algorithm work as a pair. The batch control defines the parameter to be used by the select records algorithm. The algorithm defines the thread strategy, key fields, and makes the batch control parameter available to be used in the SQL algorithm parameter.

Customizing Which Records Are Selected

There are two ways to customize what is provided with the system.

If your organization processes records using the same criteria as the samples delivered in the system, in this case CIS Division, simply create a custom batch control for each CIS Division and specify the corresponding parameter value.

If your organization processes records using different criteria, the following points summarize the steps needed to implement a new background process using plug-ins for the specific functionality:

- Duplicate an existing batch control
- Remove the batch parameter not relevant to your organization and replace with the relevant batch parameter. Note: only the batch parameters associated with selecting records should be removed or updated. Parameters uses to process records must remain.

Copyright © 2026, Oracle and/or its affiliates.

- Duplicate the plug-in script associated with the select records algorithm type. Modify the script to pass the new batch parameter(s) to bind variables and remove those that are no longer batch parameters.
- Duplicate the select records algorithm type and associate with the new script.
- Create an algorithm for the new algorithm type. The SQL parameter value should be copied from the original algorithm type's algorithm parameter value and modified to make use of the newly added bind variables. (**Note:** Do not change the fields selected or their order.)
- Update the batch control to use the new select records algorithm.

The Big Picture of Sample & Submit

Sample and Submit refers to the ability to create Activity Requests. This is functionality that enables an implementer to design an ad-hoc batch process using the configuration tools.

Some examples of such processes are:

- Send a letter to customers that use credit cards for auto pay and the credit card expiration date is within 30 days of the current date.
- Stop auto pay for customers that use credit cards as the form of payment if the credit card has already expired. Notify the customer that their auto pay agreement has been terminated and that they need to call to reinstate.
- Select auto pay accounts that have more than X non-sufficient fund penalties, stop the auto pay agreement and notify the customer.

> **Note**
>
> The terms *activity request* and *sample & submit request* may be used interchangeably.

- Activity Request Type Defines Parameters
- Preview A Sample Prior To Submitting
- Credit Card Expiration Notice
- Defining a New Activity Request
- Setting Up Activity Request Types
- Maintaining Sample & Submit Requests

Activity Request Type Defines Parameters

For each type of process that your implementation wants to implement, you must configure an activity request type to capture the appropriate parameters needed by the activity request.

Preview A Sample Prior To Submitting

Copyright © 2026, Oracle and/or its affiliates.

To submit a new activity request, a user must select the appropriate activity request type and enter the desired parameter values, as applicable.

After entering the parameters, the following actions are possible

- Click **Preview** to see a sample of records that satisfy the selection criteria for this request. This information is displayed in a separate map. In addition, the map displays the total number of records that will be processed when the request is submitted. From this map you can **Save** to submit the request, go **Back** to adjust the parameters or **Cancel** the request.
- Click **Cancel** to cancel the request.
- Click **Save** to skip the preview step and submit the request.

When an activity request is saved, the job is not immediately submitted for real time processing. The record is save in the status Pending and a monitor process for this record's business object is responsible to transition the record to Complete.

As long as the record is still Pending, it may be edited to adjust the parameters. The preview logic described above may be repeated when editing a record.

The actual work of the activity request, such as generating customer contact records to send letters to a set of customers, is performed when transitioning to Complete (using an enter processing algorithm for the business object).

Credit Card Expiration Notice

The base product supplies a sample process to find customers that use credit cards for auto pay and their credit card expiration date is within *X* days of the current date.

To this functionality the following configuration tasks are needed:

- Define an appropriate customer contact class and type to use.
- Define appropriate activity request Cancellation Reasons. Cancellation reasons are defined using a customizable lookup. The lookup field name is C1_AM_CANCEL_RSN_FLG.
- Define an activity request type for the business object C1-NotifyExpiringCreditCardTyp . You may define default parameter values for the number of days for expiration and customer contact class and type.

Defining a New Activity Request

To design a new ad-hoc batch job that users can submit via Sample and Submit, first create a new Activity Request Type business object. The base product BO for the activity type C1-NotifyExpiringCreditCardTyp may be used as a sample.

The business object for the activity request includes the functionality for selecting the records to process, displaying a preview map for the user to review and to perform the actual processing. The base product BO for the activity request C1-NotifyExpiringCreditCardReq may be used as a sample. The following points highlight the important configuration for this business object:

- Special BO options are available for the activity request BOs to support the Preview Sample functionality.

Copyright © 2026, Oracle and/or its affiliates.

  - Activity Request Preview Service Script. This script is responsible for retrieving the information displayed when a user asks for a preview of a sample of records.
  - Activity Request Preview Map. This is the map that is invoked to display the preview sample results.
- The enter algorithm plugged into the Complete state is responsible for selecting all the records that satisfy the criteria and processing the records accordingly.

Setting Up Activity Request Types

Activity Request Types define the parameters to capture when submitting an activity request through Sample and Submit. To set up an activity request type, open **Admin**, and then **Customer**, and then **Activity Request Type**, and then **Add.**

The topics in this section describe the base-package zones that appear on the Activity Request Type portal.

- Activity Request Type List
- Activity Request Type

Activity Request Type List

The Activity Request Type List zone lists every activity type. The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent activity type.
- Click the Add link in the zone's title bar to add a new activity type.

Activity Request Type

The Activity Request Type zone contains display-only information about an activity request type. This zone appears when an activity request type has been broadcast from the Activity Request Type List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click the **Edit** button to start a business process that updates the activity request type.
- Click the **Delete** button to start a business process that deletes the activity request type.
- Click the **Duplicate** button to start a business process that duplicates the activity request type.
- State transition buttons are available to transition the activity request type to an appropriate next state.

Please see the zone's help text for information about this zone's fields.

Maintaining Sample & Submit Requests

Copyright © 2026, Oracle and/or its affiliates.

Use the Sample and Submit transaction to view and maintain pending or historic activity requests. Navigate using **Menu**, and then **Tools**, and then **Sample & Submit Request**, and then **Search.**

- Sample & Submit Request Query
- Sample & Submit Request Portal

Sample & Submit Request Query

Use the query portal to search for an existing sample & submit request. Once a request is selected, you are brought to the maintenance portal to view and maintain the selected record.

Sample & Submit Request Portal

This portal appears when a sample & submit request has been selected from the Sample & Submit Request Query portal.

The topics in this section describe the base-package zones that appear on this portal.

- Sample & Submit
- Sample & Submit Log

Sample & Submit

The Sample & Submit zone contains display-only information about an activity (sample & submit) request. The following functions are available:

- Click the **Edit** button to modify the parameters. Refer to Preview A Sample Prior to Submitting for more information.
- If the activity request is in a state that has valid next states, buttons to transition to each appropriate next state are displayed.

Please see the zone's help text for information about this zone's fields.

Sample & Submit Log

This is a standard log zone.

Security Addendum

This chapter is an addendum to the general Defining Security and User Options chapter. This addendum describes security functionality that is specific to Oracle Utilities Customer Care and Billing.

- Implementing Account Security
- Masking Sensitive Data
- Encrypting Sensitive Data

Copyright © 2026, Oracle and/or its affiliates.

- Data Privacy Addendum

Implementing Account Security

> **Caution**
>
> This section assumes you understand The Big Picture of Row Security.

When you create an account, you must define which users can access the account's information. For example,

- If you have customers in two geographic territories, you may need to restrict access to accounts based on the office that manages each territory. For example, only users in the northern office may manage accounts in the northern territory.
- If you have industrial and residential customers, you may need to restrict access to these different customer segments based on the skill set of the users. For example, some users are skilled in dealing with industrial customers, while others are skilled in dealing with residential customers.

By granting a user access rights to an account, you are actually granting the user access rights to the account's bills, payment, adjustments, orders, etc.

> **NOT_SUPPORTED**
>
> Refer to If You Do Not Practice Account Security for setup instructions if your organization doesn't practice account security.

> **Note**
>
> **Account security may also affect persons and premises.** Refer to Persons Can Also Be Secured for how access to person information is also restricted by account security. Refer to Premises Can Also Be Secured for how access to premise information is also restricted by account security.

The topics in this section describe how to implement account security.

- Persons Can Also Be Secured
- Premises Can Also Be Secured
- Data Becomes Invisible When Access Is Restricted
- Restricted Transactions
- Account Security Case Studies
- The Default Access Group
- If You Do Not Practice Account Security

Copyright © 2026, Oracle and/or its affiliates.

Persons Can Also Be Secured

It's important to be aware that persons can also be secured as a result of "account security". It works like this:

- If a person is linked to at least one account, users will not be allowed access to the person (or the person's related information) unless they have access to at least one of the person's accounts.
- If a person is not linked to any accounts (a rare situation), any user may access the person.

> **Note**
>
> **How are persons linked to accounts?** A person is linked to an account when an account is created using the methods described under How To Add A New Customer From Control Central and Order User Interface Flow. In addition, you may manually link and unlink persons from account using the Account - Person page.

Premises Can Also Be Secured

It's important to be aware that premises can also be secured as a result of "account security". It works like this:

- If a premise is linked to at least one account, users will not be allowed access to the premise (or the premise's related information) unless they have access to at least one of the premise's accounts.
- If a premise is not linked to an account (a rare situation), then all users may access the premise.

> **Note**
>
> **How are premises linked to accounts?** A premise is indirectly linked to an account. For the purpose of access restriction, we deem a premise as being linked to an account if at least one of its service points is linked to at least one of the account's service agreements.

Data Becomes Invisible When Access Is Restricted

The following summarizes the impact of a user not having access to an account:

- Account Security and Control Central
- Account Security and Searches (and Maintenance Pages)
- Account Security and To Do Lists

Copyright © 2026, Oracle and/or its affiliates.

- Account Security and Control Central
- Account Security and Searches (and Maintenance Pages)
- Account Security and To Do Lists

Account Security and Control Central

This section summarizes the impact of account security and searching:

- Searches are affected as follows:
- An account will only be visible if a user has access to the account's access group.
- Persons that are not linked to accounts will be visible to all users.
- If a person is linked to an account, the person will only be visible if the user has access to at least one of the person's accounts.
- Premises that are not linked to accounts will be visible to all users unless secured with the Premise/SP installation option and the user's associated CIS Divisions and the premise has a CIS Division.
- If a premise is linked to an account, the premise will only be visible if the user has access to at least one of the premise's accounts.
- The alerts that highlight the existence of "multiple relationships" are not impacted by account security. Specifically:
  - The 'Person has multiple accounts' alert will appear if the selected person is linked to multiple accounts, even if the user doesn't have access to every account. Note well, the person couldn't have been selected if the user didn't have access rights to at least one account.
  - The 'Premise has multiple accounts' alert will appear if the selected premise is linked to multiple accounts, even if the user doesn't have access to every account. Note well, the premise couldn't have been selected if the user didn't have access rights to at least one account.
- Only accounts to which the user has access will be displayed in the person tree.
- Only accounts to which the user has access will be displayed in the account tree.
- All other pages contain information related to the current account context. The current account context can never reference an inaccessible account and therefore these pages are not impacted by account security.

> **Note**
>
> The user must have access rights to search for and view an account. If your implementation wishes to allow users to search for all accounts without validating the user's access rights, this can be configured by setting up the **Search All Account** option type on the Customer Information Options Feature Configuration

Account Security and Searches (and Maintenance Pages)

Searches are the gateway to the information that appears on maintenance pages. In general, account-related information is suppressed when a user doesn't have access rights to the

Copyright © 2026, Oracle and/or its affiliates.

account. This suppression is true for rows that directly reference an account AND for rows that indirectly reference an account. For example:

- A user can only see bills associated with accounts to which they have access rights.
- A user can only see financial transactions associated with service agreements that are, in turn, associated with accounts to which they have access rights.

> **Note**
>
> **Person and premise searches are also impacted.** Keep in mind that information will be suppress from both person and premise-oriented searches if the person / premise is related to accounts. Refer to Persons Can Also Be Secured for how access to person information is also restricted by account security. Refer to Premises Can Also Be Secured for how access to premise information is also restricted by account security.

Account Security and To Do Lists

Account security does NOT impact the information that appears in a user's To Do list. Rather, we have assumed that your To Do roles (and the users assigned to these roles) are consistent with your account security requirements. This can result in anomalies. For example, it's possible for a supervisor to assign a bill segment error to a user who doesn't have access to the bill segment's account. This user will then see the related To Do entry in their Bill Segments In Error To Do list. However, when they drill down on the entry, account security will manifest itself (i.e., the user won't be able to display the bill segment that's in error). This happens because the drill down causes the bill segment search logic to execute. This logic inhibits the selection of bill segments if the user can't access the related account.

To minimize these anomalies, we recommend the following:

- Setup To Do Roles consistent with your Data Access Roles.
- Setup Account Management Groups that are consistent with your Access Groups.
- Setup default To Do Roles on your Account Management Groups for each To Do type.

Restricted Transactions

The following table lists all transactions that have some type of account security. The following notation is used to describe the type of account security:

- **Account-oriented.** This notation is used if the respective transaction uses basic account security (i.e., the user must belong to at least one data access role that has access to the account's access group in order to see the information).
- **Person-oriented.** This notation is used if the respective transaction uses person-oriented account security. Refer to Persons Can Also Be Secured for more information.
- **Premise-oriented.** This notation is used if the respective transaction uses premise-oriented account security. Refer to Premises Can Also Be Secured for more information.
- None of the above. Some unusual transactions have unusual implementations of account security. These are described below.

Copyright © 2026, Oracle and/or its affiliates.

Transaction	Type of Account Security
Account	Account-oriented
Account Bill / Payment History	Account-oriented
Account Financial History	Account-oriented
Account Interval Info	Account-oriented
Account Payment History	Account-oriented
Account Person Replicator	Account-oriented
Account SAs for Debt Class	Account-oriented
Adjustment	Account-oriented
Adjustment Calculation Line Characteristics	Account-oriented
Appointment	Premise-oriented
Bill	Account-oriented
Bill Print Group	Person-oriented
Bill Segment	Account-oriented
Billable Charge	Account-oriented
Budget Review	Account-oriented
Case	Account-oriented, Person-oriented and Premise-oriented
Collection Agency Referral	Account-oriented
Collection Process	Account-oriented
Contract Option	Account-oriented
Control Central	Account-oriented, Person-oriented and Premise-oriented
Customer Contact	Person-oriented
Cut Process	Account-oriented
Declaration	Account-oriented
Deposit Review	Account-oriented
Field Activity	Premise-oriented
Field Order	Premise-oriented
Financial Transaction	Account-oriented
Financial Transaction on a Bill	Account-oriented
Financial Transaction on a Payment	Account-oriented
Interval Profile	Account-oriented (for SA-specific profiles)
Landlord Agreement	Account-oriented
Loan	Account-oriented
Match Event	Account-oriented
Multi-Cancel/Rebill	Account-oriented
Non-Billed Budget	Account-oriented
Order	Account-oriented, Person-oriented and Premise-oriented
Overdue Process	Account-oriented
Pay Plan	Account-oriented

Copyright © 2026, Oracle and/or its affiliates.

Payment	Account-oriented
Payment Arrangement	Account-oriented
Payment Arrangement for Bills	Account-oriented
Payment Event	The user must have access to ALL accounts linked to the payment event.
Payment Event Quick Add	The user must have access to ALL accounts linked to the payment event(s).
Payment Quick Add	Account-oriented
Payment / Tender Search	Account-oriented
Person	Person-oriented
Premise	Premise-oriented
Premise Management	Premise-oriented
Quote	Account-oriented
SA Billing History	Account-oriented
SA Cash Accounting Balance	Account-oriented
SA Financial History	Account-oriented
SA Relationship	Account-oriented
Service Agreement	Account-oriented
Service Credit Event	The user must have access to ALL accounts linked to the service credit membership that has the service credit event.
Service Credit Membership	The user must have access to ALL accounts linked to the service credit membership.
Service Provider SA Relationship	Account-oriented
Severance Process	Account-oriented
Start/Stop	Account-oriented
Statement	The user can see the statement if they have access to at least one account on the statement's statement construct.
Statement Construct	The user can see the statement if they have access to at least one account on the statement construct.
Terms of Service	Account-oriented (User must have access to at least one account linked to the SA collection in the TOS.)
TOU Map	Account-oriented (for SA-specific TOU maps)
Umbrella Agreement	Account-oriented (User must have access to at least one account linked to the SA collection in the UA's TOS.)
Write Off	Account-oriented
Write Off Process	Account-oriented

Account Security Case Studies

Copyright © 2026, Oracle and/or its affiliates.

The topics in this section contain examples of how to implement account security. Use these examples to form an intuitive understanding of these objects. Once this intuition is obtained, you'll be ready to design the account security objects for your own company.

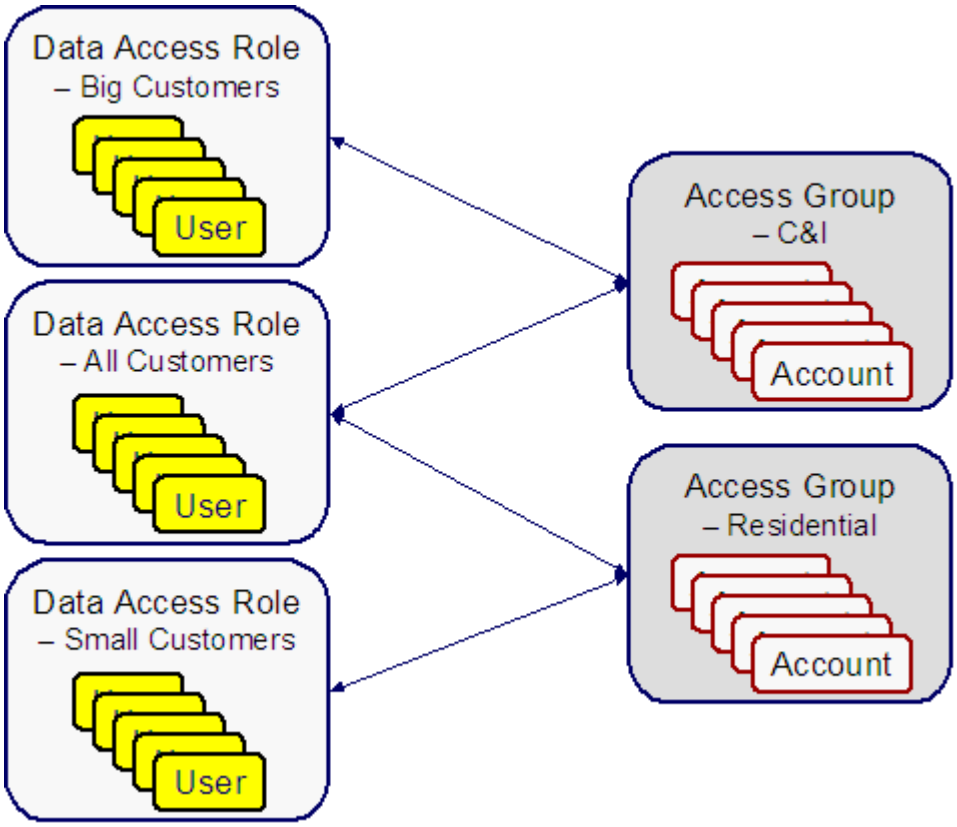
- Securing Accounts Based On Customer Class
- Securing Accounts Based On Region

Securing Accounts Based On Customer Class

Assume the following security requirement exists:

- You have two broad groups of accounts:
- Residential accounts.
- Commercial / Industrial accounts.
- Users can be classified as have one of the following access rights:
  - May access all accounts.
  - May only access residential accounts.
  - May only access commercial / industrial accounts.

The following diagram illustrates the access groups and data access roles required to implement these requirements:



Notice the following about the above:

- There are 2 access groups because access to accounts is based on whether the account is considered to be residential or commercial/industrial.

Copyright © 2026, Oracle and/or its affiliates.

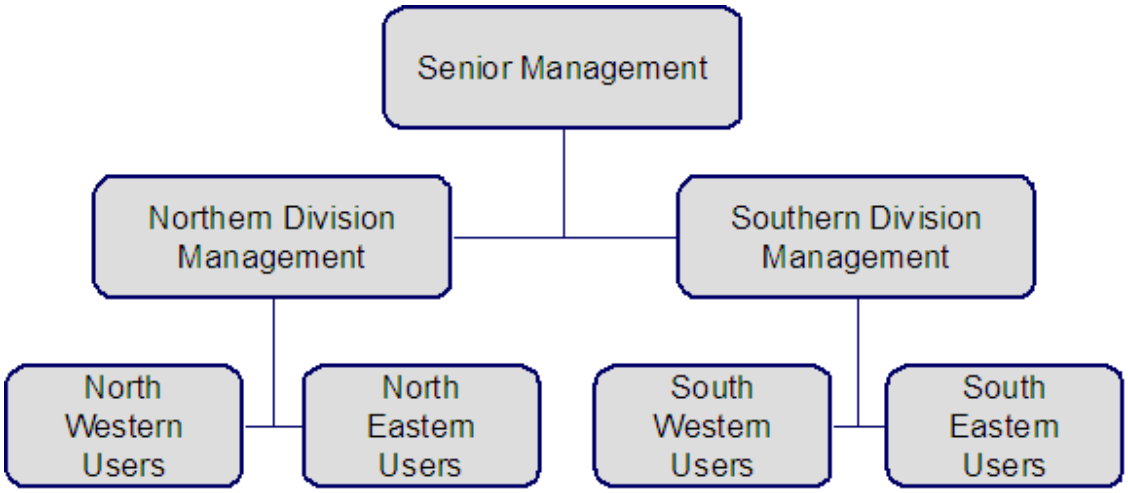
- The Big Customers data access role is only linked to the C&I access group.
- The Small Customers data access role is only linked to the Residential access group.
- The All Customers access role is linked to both the C&I and Residential access groups. Users with this role can therefore access all accounts.

Securing Accounts Based On Region

Assume that accounts are classified as belonging to one of the following regions:

- Northwestern
- Northeastern
- Southwestern
- Southeastern

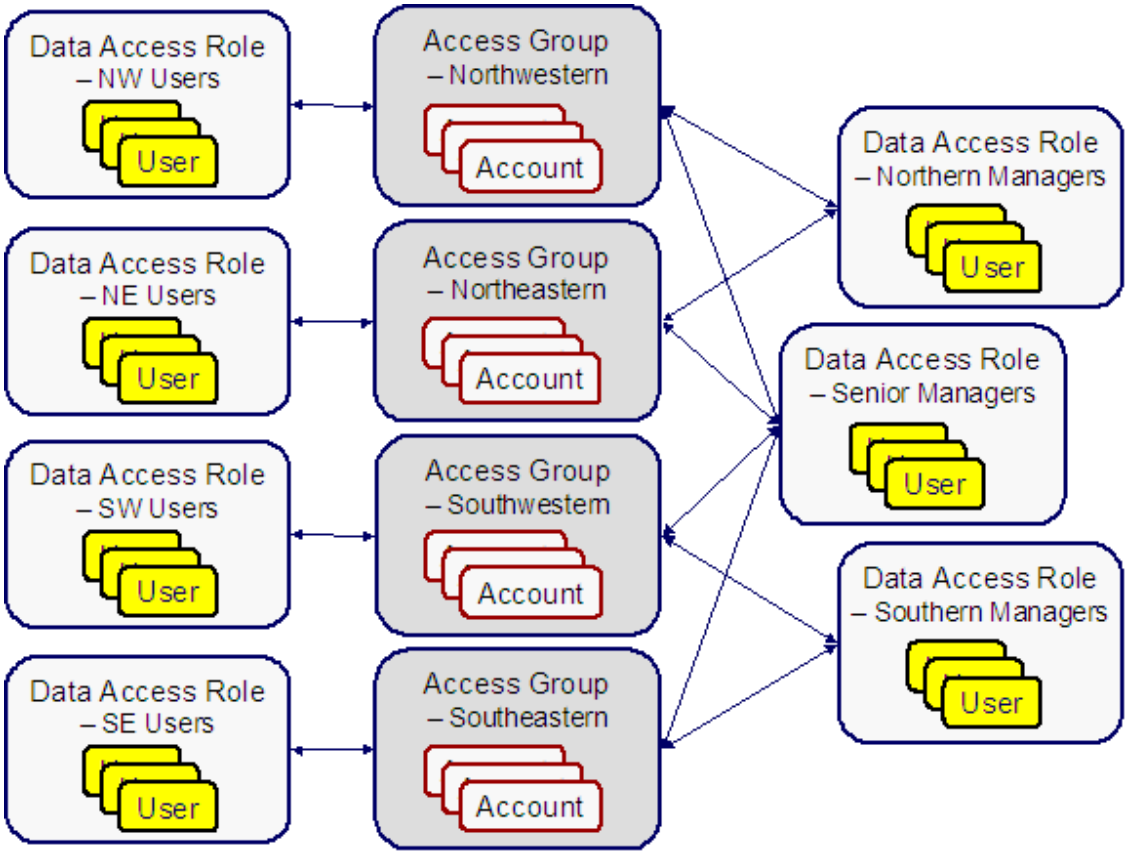
Assume the following company hierarchy exists:



- Senior Management has access to all customers
- Northern Division Management has access to all customers in the Northwestern and Northeastern divisions.
- Southern Division Management has access to all customers in the Southwestern and Southeastern divisions.
- Northwestern Users have access to all customers in the Northwestern division.
- Northeastern Users have access to all customers in the Northeastern division.
- Southwestern Users have access to all customers in the Southwestern division.
- Southeastern Users have access to all customers in the Southeastern division.

The following diagram illustrates the access groups and data access roles required to implement these requirements:

Copyright © 2026, Oracle and/or its affiliates.



Notice the following about the above:

- There are 4 access groups because access to accounts is based on the region in which they are located (and there are 4 regions).
- There are 7 data access roles because each component of every layer of the access hierarchy requires a separate data access role.

The Default Access Group

There are two ways an access group can be assigned to an account:

- The base package will default an account's access group based on the user who adds the account. It uses the user's default access group to do this.
- If you can conceive of a rule to assign an appropriate access group to a newly added account, you can have your programming staff introduce a user exit to the account row program to implement this rule. For example, a user exit could be developed that assigns an access group to an account based on its customer class. The name of the user-exit is AFTER-MOVE-CORR-ADD and the name of the account row program is CIPCACCR.

> **Warning**
>
> Regardless of the method used to assign an account's access group, please be aware that the user who adds an account must have access to this access group.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Subsequent changes to an account's access group.** A user may change an account's access group to any access group to which they have access.

If You Do Not Practice Account Security

If you do not restrict access to accounts (i.e., all users can access all accounts), you must set up one access group and one data access role and then indicate all users are part of this role. You should also define the access group as the default access group on all of your users (so that new accounts are all labeled with this access group).

Masking Sensitive Data

Refer to Masking Data for instructions describing how to configure the system to mask sensitive data like a customer's social security number or bank account number. If your implementation intends to mask any of the information that appears in the Customer Information Zone please navigate to this zone's documentation for special instructions.

Encrypting Sensitive Data

The product provides support for encrypting certain sensitive data, which may be necessary if your implementation wants to meet security standards such as those for credit card transactions, as specified by the Payment Card Industry Security Standards Council.

Please be sure that you are familiar with the information documented in Database Encryption and Masking before reading further.

The remaining details in this section provide information about the fields in the product that support encryption along with the suggested configuration for the **Encryption** feature configuration entry if your implementation opts to encrypt the data.

Note that as described in the above link, the recommended method to use for defining encryption keys is through a key ring. This method supports key rotation. For implementations that are using the keystore for defining keys, refer to Moving from Keystore to Key Ring for information about adopting the recommended method.

External Account ID and Credit Card Number Encryption

If your implementation would like to encrypt credit card IDs, the following entry should be added to the Encryption feature configuration so that the data stored in the External ID column on the Account Automatic Payment table is encrypted:

table='CI_ACCT_APAY', field='EXT_ACCT_ID', encryptedField='ENCR_EXT_ACCT_ID'

table='CI_APAY_CLR_STG', field='EXT_ACCT_ID', encryptedField='ENCR_EXT_ACCT_ID'

table='CI_PEVT_DTL_ST', field='EXT_ACCT_ID', encryptedField='ENCR_EXT_ACCT_ID'

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> All configurations must share the same key ring.

> **Note**
>
> - A customer's bank account ID or credit card number may also be stored in a column when an order is completed. A field encryption feature configuration for the Order Field table (CI_ENRL_FLD) does not need to be configured if the customer's bank account ID or credit card number is referenced on this table. The encryption for the field should be handled by a Column Reference - Preprocessing algorithm. The base product provides an Encrypt Account Auto Pay External Account Id Column Reference Value algorithm type for this purpose.
> - A field encryption feature configuration option needs to be configured for each schema field (in a CLOB / XML field) that represents a customer's bank account ID or credit card number in any of your implementation schemas. If the value of an encrypted schema field will later be populated on or compared against an encrypted table field, the schema field and the table field must share the same key ring.

If your implementation is integrated with a self-service application and one time payment details are captured in Person Self-Service Payment Option (C1_PER_SS_PAYOPT), the product supports encryption of the Credit Card Number and External Account ID fields on this table. In addition, the product supports a hash representation of this fields for searching purposes. To encrypt Credit Card Number and External Account ID, add the following entries to the Encryption feature configuration:

- **table='C1_PER_SS_PAYOPT', field='CREDIT_CARD_NBR', encryptedField='ENCR_CREDIT_CARD_NBR',hashField='HASH_CREDIT_CARD_NBR'**
- **table='C1_PER_SS_PAYOPT', field='EXT_ACCT_ID', encryptedField='ENCR_EXT_ACCT_ID', hashField='HASH_EXT_ACCT_ID'**

Bank Account Number Encryption

If your implementation would like to encrypt the bank account number stored on the back account table, the following entry should be added to the Encryption feature configuration:

table='CI_BANK_ACCOUNT', field='ACCOUNT_NBR', encryptedField='ENCR_ACCOUNT_NBR'

MICR ID Encryption

The product provides support for encrypting MICR ID associated with payments. In addition, the product supports capturing a hash representation of this field for searching purposes. This field is captured on three tables and entries should be added to the Encryption feature configuration for each as follows:

- **table='CI_PEVT_DTL_ST', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**
- **table='CI_PAY_TNDR_ST', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**

Copyright © 2026, Oracle and/or its affiliates.

- **table='CI_PAY_TNDR', field='MICR_ID', encryptedField='ENCR_MICR_ID', hashField='HASH_MICR_ID'**

> **Note**
>
> All configurations must share the same key ring.

Person ID Number Encryption

The product provides support for encrypting an identifier associated with a person. In addition, the product supports capturing a hash representation of this field for searching purposes. Because the ID collection on Person includes an ID type, your implementation may decide to encrypt all IDs captured for the person or may choose to only encrypt IDs of one or more specific ID types. If you want to encrypt multiple ID types, you must configure a field encryption option type for each ID type. The following shows an example of an encryption feature configuration entry for encrypting an ID with the type “SSN”:

table='CI_PER_ID', field='PER_ID_NBR', encryptedField='ENCR_PER_ID_NBR', hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN’

table='CI_ENRL_PER_ID', field='PER_ID_NBR', encryptedField='ENCR_PER_ID_NBR', hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN’

> **Note**
>
> All configurations must share the same key ring.

> **Note**
>
> - The Person ID Number may also be stored in a column when an order is completed. A field encryption feature configuration for the Order Field table (CI_ENRL_FLD) does not need to be configured if the Person ID Number is referenced on this table. The encryption for the field should be handled by a Column Reference - Preprocessing algorithm. The base package provides an Encrypt Column Reference Value algorithm type for this purpose.
> - A field encryption feature configuration option needs to be configured for each schema field (in a CLOB / XML field) that represents a Person ID Number in any of your implementation schemas. If the value of an encrypted schema field will later be populated on or compared against an encrypted table field, the schema field and the table field must share the same key ring.

Company ID Encryption

The product provides support for encrypting the Company ID that is populated into automatic payment extract files. If your implementation would like to encrypt company IDs on the Auto Pay Route Type table (CI_APAY_RT_TYPE), the following entry should be added to the Encryption feature configuration:

table='CI_APAY_RT_TYPE', field='CO_ID', encryptedField='ENCR_CO_ID'

Copyright © 2026, Oracle and/or its affiliates.

In addition, if your implementation allows Company ID overrides by CIS Division, the following entry should be added to the Encryption feature configuration so that the Company ID on the CIS Division Route Type table (C1_CIS_DIV_RTYP) is encrypted.

table='C1_CIS_DIV_TRYP', field='OVRD_CO_ID', encryptedField='ENCR_OVRD_CO_ID'

- Managing Encrypted Data

Managing Encrypted Data

The sections in this topic extend the information described in the framework documentation's Managing Encrypted Data topic.

Initially Encrypting Data

The current supported configuration for encryption is using the key ring. For implementations using this configuration, performing initial encryption on all the fields defined in the encryption feature configuration, the information on how to encrypt this data for the first time is described in the framework documentation topic Initially Encrypting Data

In addition, if you have encrypted data defined in a Column Reference on the Order table, you will also need to run this background process:

- C1-ECRVL: Encrypt Legacy Column Reference Values

For backwards compatibility, the product continues to support the legacy method of encryption using keystore. This is not recommended as it doesn't support key rotation. Legacy batch controls supplied to support this method of encrypting will no longer supported in an upcoming release.

Moving from Keystore to Key Ring

The framework documentation topic Moving from Keystore to Key Ring describes the basic steps to take to start using key ring to define your encryption keys rather that a keystore. For Customer Care and Billing implementations, the following additional steps are needed.

- Previously, it was documented that you did not need entries in feature configuration for encrypting the External Account ID field on the Auto Pay Clearing Staging table (CI_APAY_CLR_STG) or the Payment Event Upload Staging table (CI_PEVT_DTL_ST). Moving forward, configuration is needed for these tables as described in External Account ID and Credit Card Number Encryption.
- Previously, it was documented that you did not need entries in feature configuration for encrypting the Person ID Number on the Order Person ID table (CI_ENRL_PER_ID) table. Moving forward, configuration is needed for these tables as described in Person ID Number Encryption.

Data Privacy Addendum

This section is an addendum to the general Data Privacy chapter. This addendum describes Object Erasure functionality that is specific to Oracle Utilities Customer to Meter.

- Customer Erasure
- The Person MO Controls Customer Erasure
- Persons, Accounts And Service Agreements Are Obfuscated
- Configuring The System For Customer Erasure

Copyright © 2026, Oracle and/or its affiliates.

- Notifying External Systems About Customer Erasure

Customer Erasure

A customer's personal information can be erased from the system under certain circumstances - the most common of which is when the information is no longer necessary for the original purposes of which it was collected and processed. This is usually the case when a customer's accounts have been closed for some time and there is no other activity for that customer.

Erasure can be triggered automatically (e.g. as service agreements go into a final state) or manually (e.g. when a customer calls in to request for erasure).

- Automatic Trigger Of Customer Erasure
- Manual Trigger Of Customer Erasure

Automatic Trigger Of Customer Erasure

A person's eligibility for erasure can be automatically evaluated when key events occur on that person's accounts. These events include:

- Adding a service agreement
- Closing a service agreement
- Canceling a service agreement
- Reactivating a service agreement
- Accepting a proposal service agreement
- Declining a proposal service agreement
- Reopening a proposal service agreement
- Deleting a proposal service agreement
- Monitoring SA Triggers For Customer Erasure

Monitoring SA Triggers For Customer Erasure

The base product provides an MO Audit algorithm, Monitor SA Manage Erasure Events (C1-SAMNGEREV) that implementations need to plug in on the Service Agreement (SA) MO to enable automatic triggering of erasure.

Note that this algorithm simply creates a General Process record for the Person. This technique allows the system to perform the erasure evaluation (i.e. in the Manage Erasure Schedule algorithm) in batch. This is to minimize the number of times that the Manage Erasure Schedule algorithm is executed for the same Person. For instance, if the customer had one SA closed, another one canceled and another SA created, the erasure evaluation will only be executed once (i.e. when the C1-MNERP batch process runs) even if there were three SA events for that person.

The Manage Erasure Schedule for Person (C1-MNERP) plugin-driven batch process is provided to process the General Process records for managing person erasure. The Process Record algorithm on this batch process invokes the Manage Object Erasure (F1-ManageErasureScheduleDriver) business service. Refer to the C1-MNERP batch control in the application for more details.

Copyright © 2026, Oracle and/or its affiliates.

Manual Trigger Of Customer Erasure

A customer may call in to request an agent to erase them from the system.

The base product provides a Trigger Customer Erasure (C1-TrgCustEr) BPA script that users can configure as a favorite script in their User Preferences.

This script assumes that the user retrieves the Person before running the script – meaning, the Person is already in the current context. The script pulls in the Person ID and renders the information on the map, which also captures Erasure Date, Erasure Reason and Comments as additional inputs.

A Confirm action will invoke the same Manage Erasure Schedule processing that is executed in automatically triggered person erasures. If the customer is eligible for erasure, the Person and its associated Account(s) and Service Agreement(s) will be scheduled for erasure on the designated Erasure Date. Otherwise, an error will be displayed, indicating ineligibility for erasure.

Refer to the Trigger Customer Erasure (C1-TrgCustEr) BPA script in the application for more details about this script.

The Person MO Controls Customer Erasure

A customer's eligibility for erasure is evaluated at the Person level. This means that only the Person maintenance object will specify a Manage Erasure Schedule algorithm. Eligibility criteria in this algorithm will check the status of the Person's Account(s) and Service Agreement(s) to determine whether or not these objects can be erased. Object Erasure Schedule records are added/updated/discarded, based on the outcome of the eligibility check.

- Manage Erasure Schedule Algorithm: Person Erasure Evaluation

Manage Erasure Schedule Algorithm: Person Erasure Evaluation

The base product supplies a Maintenance Object - Manage Erasure Schedule algorithm – i.e. Person Erasure Evaluation (C1-PEREREVAL) that determines Person Erasure eligibility based on the status of the service agreements (SAs). If all SAs are in some final state - e.g. Closed, Canceled - the Person is deemed eligible for erasure and the corresponding Object Erasure Schedule records are added. Otherwise, the Person is deemed ineligible for erasure. If a person that is already scheduled for erasure happens to have new activity - e.g. started a new SA, reactivated a Closed SA - any existing Object Erasure Schedule records for that Person and its Account(s) and SA(s) will be discarded.

Persons, Accounts And Service Agreements Are Obfuscated

The following sections describe the erasure processing for Person, Account and Service Agreement information.

- Purge Versus Obfuscate
- Person, Account and Service Agreement Erasure BOs
- Person, Account and Service Agreement Obfuscation Algorithms

Copyright © 2026, Oracle and/or its affiliates.

Purge Versus Obfuscate

Purging involves physical deletion of a record. On the other hand, obfuscation involves updating certain fields in the table that are considered personally identifiable information (PII) – e.g. name, address, etc. The values of these fields are updated to blank or some value that is no longer identifiable to a person – e.g. the random-generated primary key of the Person record.

Base-supplied Person, Account and SA erasure logic performs obfuscation on the root tables of these maintenance objects (MOs). The reasons are:

- Purging could have impacts to external systems
- Purging might cause orphan data or invalid foreign key references within the system

However, base logic does purge most of the data in these MOs' child tables.

Refer to the next sections for details on each object's obfuscation processing.

Person, Account and Service Agreement Erasure BOs

The base product provides the following Object Erasure Business Objects (BOs):

- Person Erasure: C1-ErasureSchedulePerson
- Account Erasure: C1-ErasureScheduleAccoun
- Service Agreement Erasure: C1-ErasureScheduleSA

All are child BOs of the Erasure Root parent BO - F1-ErasureScheduleRoot. These BOs are delivered plugged in on their corresponding maintenance objects.

Refer to the Business Object configuration in the application for more details.

Person, Account and Service Agreement Obfuscation Algorithms

The Person, Account and Service Agreement Erasure business objects include an Enter algorithm (on the Erased state) that is responsible for obfuscating data in the maintenance object's tables. The following sections describe each algorithm.

- Person Obfuscation
- Account Obfuscation
- Service Agreement Obfuscation

Person Obfuscation

The Person Erasure (C1-ErasureSchedulePerson) BO is delivered with an enter algorithm on the Erased state - C1-PEROBJERS.

This algorithm performs the following:

- On the Person root table:
  - Updates the Address1, City and Postal fields with the PER_ID value.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> These fields are search fields and are thus indexed.

  - Deletes the contents of the Person Data Area (CLOB) field, if the algorithm is configured to do so.
- Deletes all records in the Person Seasonal Address table
- Deletes all records in the Person Identifier table
- In the Person Name table:
  - The primary name is retained and the name value is replaced with the PER_ID value.

> **Note**
>
> Deletion of the primary name makes the Person record and its Account and SA records unsearchable through the user interface - e.g. via Control Central search.

  - All other names are deleted
- Deletes all records in the Person Phone table
- Deletes all records in the Person Contact table
- Deletes all records in the Person-to-Person table
- Deletes all records in the Person Self-Service Payment Option table
- Deletes all records in Person Characteristic table, if the algorithm is configured to do so.
- Deletes all records in the Attachment table that are linked to the Person, if the algorithm is configured to do so.

Account Obfuscation

The Account Erasure (C1-ErasureScheduleAccount) BO is delivered with an enter algorithm on the Erased state - C1-ACCTOBFS.

This algorithm performs the following:

- In the Account root table:
  - Removes the link to the Mailing Premise Id
  - Clears contents of the Account Data Area
  - Deletes the contents of the Account Data Area (CLOB) field, if the algorithm is configured to do so.
- Deletes all Account Person records except the main Person.

> **Note**
>
> Deletion of the ‘main’ person makes the Account unsearchable through the user interface - e.g. via Control Central search.

- Updates the main Person's Address Source to Person

Copyright © 2026, Oracle and/or its affiliates.

- Deletes all records in the Person Address Override table
- Deletes all records in the Account Auto Pay table
- Deletes all records in the Account Alerts table
- Deletes all records in the Account Messages table
- Deletes all records in the Account Credit Rating History table
- Deletes all records in the Account Debt Monitor Review Schedule table
- Deletes all records in the Non-Cash Deposit table
- Deletes all records in the Budget Review table
- Deletes all records in the Notification Preferences table and its child tables
- Deletes all records in Account Characteristic table, if the algorithm is configured to do so.
- Deletes all records in the Attachment table that are linked to the Account, if the algorithm is configured to do so.

Service Agreement Obfuscation

The Service Agreement Erasure (C1-ErasureScheduleSA) BO is delivered with an enter algorithm on the Erased state - C1-SAOBFSCTN.

This algorithm performs the following:

- On the SA root record:
  - Removes the link to the Characteristic Premise.
  - Deletes the contents of the SA Data Area (CLOB) field, if the algorithm is configured to do so.
- Deletes all records in the SA/SP table and its child tables: SA/SP Field Activity and SA/SP Characteristics.
- Deletes all records in SA Characteristic table, if the algorithm is configured to do so.
- Deletes all records in the Attachment table that are linked to the SA, if the algorithm is configured to do so.
- Deletes all records in the SA Notification Preferences table and its child tables.

Configuring The System For Customer Erasure

The following are the steps to enable Customer Erasure:

- Plug in the Manage Erasure Schedule - Person Erasure Evaluation algorithm on the Person MO.
- Configure the Erasure Period option on each of the Person, Account and SA MOs.

> **Note**
>
> Note: The values for all three MOs should be the same, to ensure the Erasure Dates are the same on the resulting Object Erasure Schedule records.

- Plug in the Monitor SA Manage Erasure Events (C1-SAMNGEREV) Audit algorithm on the SA MO.

Copyright © 2026, Oracle and/or its affiliates.

- If using the Trigger Customer Erasure (C1-TrgCustEr) script to manually trigger person erasure:
  - Configure values for the Erasure Reason lookup field DP_CUST_ERS_RSN_FLG.
  - Configure the C1-TrgCustEr BPA script as a Favorite Script in User Preferences.

Notifying External Systems About Customer Erasure

When CCB is integrated with a meter data management system (such as Oracle Utilities Meter Data Management) and the Person and SA syncs are used to synchronize Contact and Usage Subscription information with corresponding Person and SA information, the meter data management needs to be notified when such information is already erased in CCB.

The following sync request BO schemas include a Customer Erased switch:

- MDM2 Person Sync Request: C1-MDM2PersonSyncRequest
- MDM2 SA Sync Request: C1-MDM2SASyncRequest

Logic on these BOs sets this switch to true if the related Person/SA has an Object Erasure Schedule record that is in the Erased state.

The meter data management system can use this switch to trigger its own erasure processing for Contact and Usage Subscription information. For an example, refer to the Data Privacy documentation for Oracle Utilities Meter Data Management.

Refer to the Business Object configuration in the application for more details.

Defining Converted COBOL Program Options

The topics in this section describe the transaction that allows you to define the metadata for converted COBOL programs within the current environment's database.

> ⚠ **Caution**
>
> Updating converted COBOL Programs requires technical knowledge of the system. This is an implementation and delivery issue and should not be attempted if you do not have previous experience.

- Converted COBOL Program - Main

Converted COBOL Program - Main

Use this transaction to define converted COBOL program user exits for your system. Navigate to this page using **Admin**, and then **System**, and then **Converted COBOL Program**, and then **Search.**

Description of Page

The following describes fields that are relevant to defining the user exit code that a converted COBOL Program should use:

Program Component ID represents the internal ID that is given to the converted COBOL program component.

Program Com Name is the physical name of the converted COBOL program component.

Copyright © 2026, Oracle and/or its affiliates.

Template is the template used to generate the converted COBOL program component.

Location ID *(for development purposes)*

Table *(for development purposes)*

User Exit Program. Specify if you have written user exit code for this converted COBOL program component.

Development Status Flag. Options include Note Generated and Not Regeneratable.

Short Comments provides a short description of the converted COBOL program component.

Long Comments provides more details of the converted COBOL program component.

The **Variable Information** grid (including Variable Name, Occurrence, Variable Sequence, Variable Value, and Variable Alternate Value) is *for development purposes only*.

Advanced Search Options

The product offers an option to perform expanded / fuzzy searching in Control Central Search by Name and Address. The option is only available if your implementation has appropriately configured your database for fuzzy searching.

Refer to the DBA guide in the installation guide for details on setting up the database to support fuzzy searching. Note that there are some implementations where fuzzy searching will not be possible. For example, it's only available for implementations using the Oracle database. Additionally, not all languages are supported. Refer to the Oracle Database documentation for more information about fuzzy searching.

If your implementation has configured its database to enable fuzzy searching, you must also set up a Feature Configuration to indicate that feature is supported. Find the Feature Configuration record for the **General System Configuration** feature type. (It may need to be defined if it does not exist). Choose the option type **Enable Expanded Results** and define a value of **Y**.

Integration

Oracle Utilities Customer to Meter a provides tools to facilitate the integration other systems. This section provides technical information needed by your implementers to accomplish this.

- CTI-IVR Integration
- Call Summary and Call Tagging Integration
- Analytics Integration Overview
- Self-Service Integration
- Defining DataConnect Options
- Business Flags
- Customer Engagement
- Oracle Fusion Customer Experience Cloud Service Integrations
- Customer Profile Insights

Copyright © 2026, Oracle and/or its affiliates.

- Customer Relationship Management Integration
- RightNow Knowledge Integration
- External Applications
- Configuring Master Data Synchronization
- Oracle Utilities Operational Device Management
- Analytics Integrations
- Oracle Utilities Analytics Visualization
- Oracle Utilities Network Management System
- Oracle Utilities Customer Program Cloud Service
- Asset Management System Integration
- Oracle Customer Experience for Utilities
- Oracle Utilities Data Intelligence
- Message Formats
- Data Import - Message Driven Bean Configuration
- Creating Application Links in External Applications

CTI-IVR Integration

Oracle Utilities Customer Care and Billing provides tools to facilitate the integration with your Computer Telephony Integration/Interactive Voice Response (CTI/IVR) system. The interface provides the following functionality:

- The ability to launch Control Central for a particular account ID or phone number from an external application
- The ability to accept the next call, as dictated by the CTI software,
- The ability to perform an outbound phone call from within Oracle Utilities Customer Care and Billing

This document provides technical information needed by your implementers to fully integrate with your CTI/IVR system.

- CTI-IVR Integration Configuration
- Launching The System From an External Application
- Receiving the Next Caller in the Queue
- Initiating an External Call

CTI-IVR Integration Configuration

The integration with your Computer Telephony Integration/Interactive Voice Response (CTI/IVR) system requires the setup of master configuration data that controls the Next Caller and Automated Dialer functions.

To set up the CTI/IVR integration master configuration, navigate using **Admin**, and then **General**, and then **Master Configuration**. You are brought to an all-in-one portal with options

Copyright © 2026, Oracle and/or its affiliates.

to add/edit the CTI/IVR Integration master configuration record. Refer to the embedded help for more details on how each section in this record is configured.

Launching The System From an External Application

The following section describes possible options to launch the system from an external system.

- Launching The Application Using a URL

Launching The Application Using a URL

You launch the application using a URL. With this option you can set the system to launch a script upon startup. You can also indicate to the system to automatically load an appropriate page (if this information is not part of the script).

The application includes a simple sample html page that can launch CCB Control Central Search by Phone Number or by Central Search Account Id. The page is called CTISample.HTM and is located in /cm_templates.

> **NOT_SUPPORTED**
>
> Refer to Launching A Script When Starting The System for further information.

Receiving the Next Caller in the Queue

If your CTI-IVR system allows users to request the next caller waiting in a queue, the system provides a mechanism to integrate with this functionality.

A BPA script called **Get Next Caller (C1-GtNxtClrC)** is available that can be used to request the next call waiting in an inbound queue managed by a CTI application.

When the **Get Next Caller** BPA is executed, it launches the **CTI Get Next Caller** and **Launch CTI** processing scripts that are defined in the CTI/IVR Integration master configuration. Your implementation creates the **CTI Get Next Caller** script that retrieves the next account in the queue. The base product supplies a sample **Launch CTI** BPA script that launches Control Central using the **C1-CCByAcct** BPA script and passing in the account that was returned by the **CTI Get Next Caller** script.

- Customize Integration to Your Next Caller Function

Customize Integration to Your Next Caller Function

Your implementation needs to create a **CTI Get Next Caller** processing script that will responsible for getting the next account in the queue, based on your integrated CTI application. The base-supplied **Launch CTI** processing script can be used to navigate to Control Central with the next caller’s account passed in.

Copyright © 2026, Oracle and/or its affiliates.

In order to integrate the next caller functionality with your CTI-IVR system, perform the following steps:

- Create your custom **CTI Get Next Caller** service script that retrieves the next account from the queue.
- Create a **Launch CTI** BPA script if your business rules require additional logic and/or need to launch a different page in the system..
- Plug in the **CTI Get Next Caller** and **Launch CTI scripts** on the CTI/IVR Integration master configuration.

Initiating an External Call

This section describes the automated dialer functionality provided with the system as well as information about integrating with your own automated dialer.

- Overview of Automated Dialer
- Technical Implementation of Automated Dialer
- Customize Integration to Your Automated Dialer Software
- Customize Automated Dialer User Interface

Overview of Automated Dialer

In order to initiate a call to a customer from within the system, a context menu item **Go To Automated Dialer** is available on the Person context menu. To call a customer displayed in the current context, choose this option from the person context menu and a window appears, showing a list of phone numbers defined for that person.

Select the desired phone number and click **Dial**.

> **Note**
>
> **Context Entry Secured**. The navigation key for this window automatedDialer refers to an application service to facilitate application security. If your installation does not support an integration with external dialer software, configure the security settings to ensure that users do not have access to the application service for this context entry.

Technical Implementation of Automated Dialer

The popup window is implemented as a BPA script (**C1-LAutoDial**) that calls a configurable **Automated Dialer** BPA script that is responsible for retrieving and displaying person contacts and phone numbers (i.e. if the installation option is set to Use Legacy Person Phone and Email feature configuration option is set to ‘Y’). The **C1-AutoDial** sample script provided with the system integrates to any soft phone protocol handler that launches a dialer based on overriding the “tel:” protocol from a browser, for example, Cisco IP Communicator. It launches a UI map to display the person contacts and phone numbers (i.e. if the feature option is set).

Phone Dialer Configuration

Copyright © 2026, Oracle and/or its affiliates.

If your implementation chooses to use the functionality provided with the system and integrate with a soft-phone dialer, plug in the base-supplied **C1–AutoDial** script in the CTI/IVR Integration master configuration.

Customize Integration to Your Automated Dialer Software

In order to integrate with a different automated dialer software application, your implementers must create your own **Automated Dialer** BPA script, to call your specific dialer.

- Copy the **C1-AutoDial BPA** script and make the appropriate changes to integrate with your automated dialer (e.g. change the protocol from “tel:” to “callto:”.
- Plug in your custom Automated Dialer script on the CTI/IVR Integration master configuration.

Customize Automated Dialer User Interface

Your implementation may choose to display a different user interface for the **Go To Automated Dialer** function than the one provided with the system. For example, perhaps there is more information that you would like to display in addition to the person's name and phone numbers. In order to do this, your custom **Automated Dialer** script needs to invoke a custom UI map that you create as follows:

- Copy the base-supplied **C1-AutomatedDialer** UI map and make the necessary changes.
- Invoke your custom UI map from your custom **Automated Dialer** .

Call Summary and Call Tagging Integration

Call Summary and Call Tagging uses Generative Artificial Intelligence and integration with a third-party Computer Telephony Integration (CTI) application to automatically summarize and categorize call transcripts, providing customer service agents an overview of the call. Agents can review and edit the summary and tags before saving them directly to the customer contact.

> **Note**
>
> Call Summary and Call Tagging is only available with Oracle Utilities Customer Cloud Service and Oracle Utilities Customer Care and Billing Cloud Service.

See Creating Customer Contacts Using Call Summary and Call Tagging in the *Business User Guide* for more information how users work with Call Summary and Call Tagging.

This section provides details related to implementation of Call Summary and Call Tagging, including:

- Integrating With a CTI Application
- Configuring Characteristic Types and Customer Contact Types with Characteristic Tags
- Enabling the Service Guide Zone

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Call Summary and Call Tagging relies on generative AI models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Call Summary and Call Tagging.

- Integrating with a CTI Application
- Configuring Characteristic Types and Customer Contact Types with Characteristic Tags
- Enabling the Service Guide Zone

Integrating with a CTI Application

This section provides an overview of the process to integrate a third-party Computer Telephony Integration (CTI) vendor with Call lSummary and Call Tagging, utilizing the Service Guide SDK JavaScript library. By integrating CTI, your agents can use enhanced Call lSummary and Call Tagging features such as:

- Incoming call notifications
- Call control management, including muting, holding, and ending calls
- Access to call transcripts and summaries

> **Note**
>
> Oracle does not endorse specific CTI vendors or provide implementation instructions for third-party applications. This document focuses on the integration points supported by the Service Guide SDK, detailing relevant events and APIs. Hosting third-party applications is beyond the scope of this guide.

This section contains the following:

- Understanding Computer Telephony Integration
- Prerequisites and Technical Requirements
- Configuring the CTI iFrame
- Integrating the Service Guide SDK in the Partner Application
- Understanding Computer Telephony Integration
- Prerequisites and Technical Requirements
- Configuring the CTI iFrame
- Integrating the Service Guide SDK in the Partner Application

Understanding Computer Telephony Integration

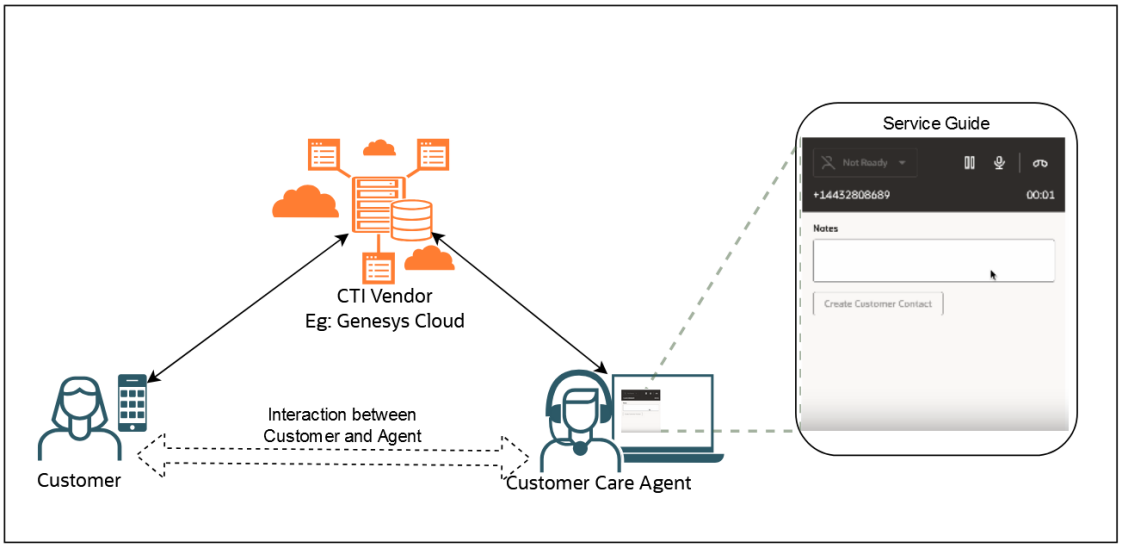
Computer Telephony Integration (CTI) is a technology that integrates computer software and telecommunication systems seamlessly. It connects a phone system to a computer, enabling users to manage phone functions directly from their desktop.

Copyright © 2026, Oracle and/or its affiliates.

With CTI, customer calls that come into the service center are routed through a CTI system and sent to an available agent. The CTI provider facilitates this process by identifying callers, providing access to customer details, and offering call control features directly from the agent's desktop. Call controls include incoming call notifications, call control options during a call (such as hold or mute), and the ability to obtain a text transcript of the call, if supported by the CTI vendor.

This image illustrates a basic flow where a call to the utility is routed via the CTI vendor and handled by an agent. The CTI provider identifies the caller, enables the agent to view customer details, and enables the agent to mute the call, put the call on hold, or transfer the call.

Figure CTI Integration Flow Example



Prerequisites and Technical Requirements

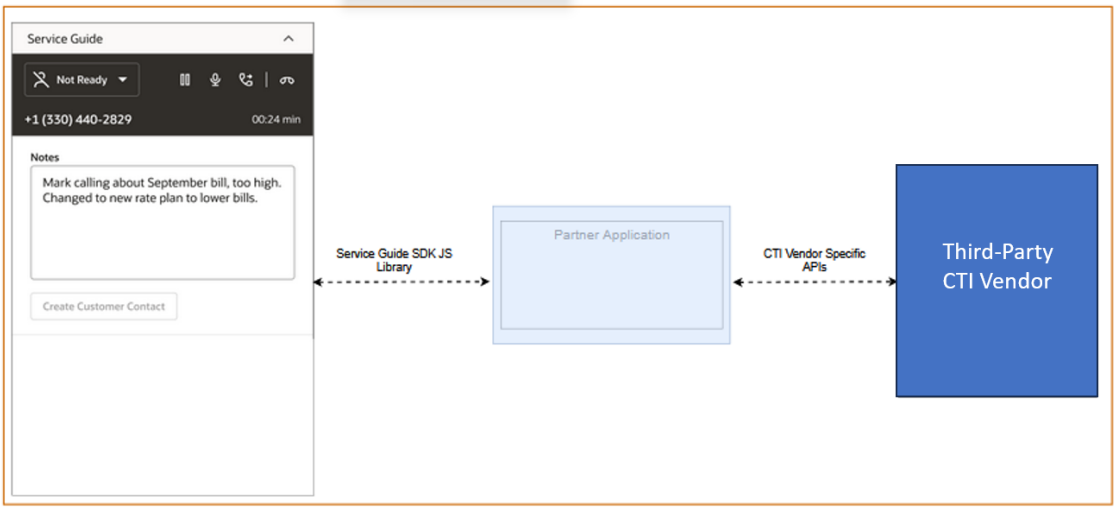
Call Summary and Call Tagging facilitates integration with third-party CTI vendors through a partner application. Oracle provides the Service Guide SDK JavaScript library for communication between the Call Summary and Call Tagging application and the CTI partner application, which runs as an invisible iFrame.

The CTI iFrame serves as a container for embedding the partner application, which is a web application developed by utility system administrators. This application communicates with the CTI backend using vendor-specific APIs to deliver the required functionality.

This image illustrates the technical components of the solution:

Copyright © 2026, Oracle and/or its affiliates.

Figure Technical Components



Before completing the steps in this guide, complete the following tasks:

1. Select a CTI vendor that meets your needs. The vendor should support both incoming and outbound calls and provide API support for accepting calls from the web application.
2. Verify that you can invoke the CTI vendor APIs.
3. Develop the partner application as a web-based solution that can be made available over https.

Configuring the CTI iFrame

After you have created your partner application, you must gather the following information, which is used to configure the application inside of the Call Summary and Call Tagging feature:

- The URL of the partner application page
- The window domain of the final page with which the partner application is loaded

URL of the Partner Application

You must provide the URL for the partner application page that will be embedded in Call Summary and Call Tagging as part of the CTI iFrame. This should include any static query parameters that the partner application expects.

Call Summary and Call Tagging appends the **`crmDomain`** query parameter to this URL, which is the window location of the Call Summary and Call Tagging application. For example:

```
https://opav12345.getway.mycti.cloud/crm/embeddableApp.html?
enableFrameworkClientId=true&dedicatedLoginWindow=true
```


Copyright © 2026, Oracle and/or its affiliates.

This example shows an iFrame URL that is used by Call Summary and Call Tagging for loading the partner application, where, `crmDomain=<Service-Guide-window-location>` is appended by Call Summary and Call Tagging.

```
https://opav23abc5a4z.getway.mycti.cloud/crm/embeddableApp.html?
enableFrameworkClientId=true&dedicatedLoginWindow=true&crmDomain=
```

Window Domain of the Final Page

You must provide the window domain, including the protocol, host and port number, of the final page with which the partner application will be loaded. The window domain may not be same as the hostname present in the iFrame URL above, as the final window may be loaded after multiple redirections. This is common when using authentication providers, and can result in a window location that is different than the initial URL. For example:

```
https://apps.usawest.mycti.cloud
```

- CTI iFrame Considerations
- Avoiding CORS Errors

CTI iFrame Considerations

When configuring the CTI iFrame, consider the following issues.

iFrame Not Visible to Agent

The CTI iFrame is not visible to the end user. Only the controls provided by Service Guide are available to the agent. However, the CTI iFrame can open any necessary windows to perform necessary operations, such as logging a user into the CTI system to establish an authenticated session with the CTI backend.

Permissions

The CTI iFrame includes the permissions listed here. These permissions allow Service Guide to handle phone calls using the browser.

```
camera *; microphone *; autoplay *; hid *
```

When prompted, agents must accept and provide these permissions in order to allow Service Guide to handle a customer call.

> **Note**
>
> Agents might have to disable pop-up blockers in their browser if the partner application uses a pop-up window for authentication purposes.

Authenticated Requests

An important aspect of CTI integration is to allow agents to make authenticated requests to the CTI system. CTI systems typically use authentication and authorization to establish a valid user login and allow the system to make appropriate API calls on behalf of the logged-in agent.

Copyright © 2026, Oracle and/or its affiliates.

Utilities can provide a window for agents to log into the CTI system, however, Oracle recommends setting up a Single Sign On (SSO) with the Oracle Customer Cloud Service application so that a single user login is recognized when an agent accesses various systems.

> **Note**
>
> Refer to your CTI vendor documentation for information about allowing API access for agents and SSO setup.

Avoiding CORS Errors

In order to embed your partner application as an <iFrame> inside Call Summary and Call Tagging, you must set up appropriate headers to avoid Cross Origin Resource Sharing (CORS) errors from browsers.

Specifically, the partner application should be set up as follows:

```
Set only "Content-Security-Policy: frame-ancestors <Service Guide Base URL>"
response header.



DO NOT SET "X-Frame-Options" response header.
```

This setup applies to the partner application and any other domains to which a request gets redirected during the initial load. When the partner application is loaded and Single Sign On (SSO) is set up, the partner application may redirect browsers to an authentication provider's URL and domain, which might differ from the partner application's domain. The browser redirects occur inside the iFrame, therefore, any such domains should follow the above recommendation to ensure successful load and redirect in the iFrame embedded in Call Summary and Call Tagging.

If the authentication provider does not allow the SSO redirect window to be embedded, the partner application can open a separate login window. The agent can then sign in using CTI system credentials or credentials for other providers that utility has configured. Once the sign in is performed and a successful authenticated session is established, this window can be closed (programmatically) and CTI initialization can proceed.

> **Note**
>
> Contact your Oracle Support team if you need assistance identifying the `<Service Guide Base URL>` for your environment.

Integrating the Service Guide SDK in the Partner Application

The partner application you implement could be a normal HTML/Javascript application. To communicate with the Call Summary and Call Tagging feature, you must inject a Javascript library, called Service Guide SDK, into your partner application. Additionally, you must:

- Invoke the APIs required for the integration, or
- Listen for necessary events

Copyright © 2026, Oracle and/or its affiliates.

The integration between Call Summary and Call Tagging and the partner application is a frontend-to-frontend integration through the javsascript SDK library.

You can import the library in the HTML page of the partner application using either method shown below.

Import Using Script Tag

Code example:

```
<script type=“module" src="https://static.oracle.com/cdn/service-guide-sdk/
<version-number>/service-guide-sdk.umd.js" onload="onSdkLoad()"></script>
        <!-- Once the above script is loaded, SDK class will be available
under window.OUSG.ServiceGuideSdk variable. -->
<script>
      function onSdkLoad() {
         const sdkInstance = new window.OUSG.ServiceGuideSdk({...});
         // You can now use "sdkInstance" variable for referring to SDK
instance.
      }
</script>
```

Once the script is loaded, SDK can be instantiated using the class available at **`window.OUSG.ServiceGuideSdk`**.

Import as Module using Dynamic Import

Code example:

```
const sdkUrl = 'https://static-stage.oracle.com/cdn/service-guide/service-
guide-sdk/<version-number>/service-guide-sdk.esm.js';
const {ServiceGuideSdk} = await import(sdkUrl);
const sdkInstance = new ServiceGuideSdk({...});
// You can now use "sdkInstance" variable for referring to SDK instance.
```

Using requirejs

Code example:

```
<!--
          First load the requirejs from appropriate source -->
<script
    id="requirejs"
    src="https://requirejs.org/docs/release/2.3.7/minified/require.js"
></script>

<!-- Now load SDK using require-js -->
<script type="module">
  const sdkUrl = 'https://static-stage.oracle.com/cdn/service-guide/service-
guide-sdk/<version-number>/service-guide-sdk.esm.js';

  require([sdkUrl], (OUSG) => {
     const sdkInstance = new OUSG.ServiceGuideSdk({...});
     // You can now use "sdkInstance" variable for referring to SDK instance.
```


Copyright © 2026, Oracle and/or its affiliates.

```
  });
</script>
```

SDK Initialization

Once the SDK library is loaded, the partner application must create an instance of the SDK by initializing it with the appropriate parameters. Call Summary and Call Tagging will add a "`crmDomain`" query parameter while loading the partner application in an iFrame. This parameter should be used while initializing SDK.

Code example:

```
// Read "crmDomain" query parameter from the window location
let url = window.location.search;
let segments = url.replace('?', '&').split('&');
const parentDomain = decodeURIComponent(segments.find(segment =>
(segment.indexOf('crmDomain') >= 0)).split('=')[1]);

//Initialize Service Guide SDK
const sdkInstance = new ServiceGuideSdkClass({
    serviceGuideDomain: parentDomain,
    isCtiInstance: true, // Most important to set
    options: {application: 'cti'} // This is also an important parameter to
include while initializing the SDK to let Service Guide know which
application the notification is coming from.
});

// You can now use "sdkInstance" variable for referring to SDK instance.
// You can also set the SDK instance on the window object in case you need to
access it later:
// window.serviceGuideSdk = sdkInstance;
```

- Operations and Events

Operations and Events

The Service Guide SDK provides two types of APIs:

1. Operations
2. Events

- Operations
- Events

Operations

When the partner application needs the Call Summary and Call Tagging feature to perform an action or send data, it must invoke an `Operation` on the SDK.

For example, the partner application wants to notify an agent, via the Service Guide zone, that it has received an incoming call. This requires the partner application to invoke the `callReceived` operation.

Available operations include:

Copyright © 2026, Oracle and/or its affiliates.

- ctiInitialized
- agentStatusUpdated
- callReceived
- initiatedCallSession
- endCallSession
- callMuted
- callUnmuted
- callOnHold
- callResumed
- transcriptReceived
- callWrappedUp
- loggedOut

To indicate that these operations are targeted towards the Service Guide interface, the operations are added under the "Ui" namespace of the SDK. Below is an example code for invoking an operation:

```
// SDK loading and initialization code as demonstrated before.

// Finish rest of the CTI specific setup
// initCti(...); //illustration purpose only

// After all initialization is done, notify Service Guide of it
// (Assuming the SDK instance is named "serviceGuideSdk")

serviceGuideSdk.Ui.ctiInitialized();

// This can also be done in some async callback if your setup needs an
asynchronous handling
// Assuming getAuthToken is an asynchronous function that returns auth token
for a logged in user.

getAuthToken(async (authToken) => {
    // assuming CTI vendor has a `ctiSdk` library that provides a client
initialization code

    await ctiSdk.ApiClient.initialize(authToken); // Another asynchronous
processing

    // At this point all CTI initialization is done, notify Service Guide.

    serviceGuideSdk.Ui.ctiInitialized();
});

...

/* Another example */
// Assuming CTI vendor has below API for subscribing to a client that invokes
the provided callback when a customer call is received.

ctiSdk.ApiClient.Interaction.subscribe(async (callData) => {
```


Copyright © 2026, Oracle and/or its affiliates.

```
    // Invoke "callReceived" operation on Service Guide SDK's "Ui" module.

    serviceGuideSdk.Ui.callReceived({
        data: {
            phoneNumber: calData.ani,
            callId: callData.id
        }
    });
});
```

Operation: ctiInitialized

This table provides details of the ctiInitialized operation:

Operation Component	Details
Description	This is the first operation the partner application invokes to notify Service Guide that the application is ready to handle CTI functionalities. It should be invoked after initial setup of the partner application is complete, which includes: • Setting up communication channels and websocket connections • Initializing various SDKs, including the Service Guide SDK • Completing all other initialization code
Payload	None
Parameter Description	Not Applicable
Notes	Service Guide expects this operation to be called when the partner application is loaded to confirm that it is ready to handle call center operations. If Service Guide does not receive a response within a predetermined time, it reloads the partner application iFrame. If the partner application can't be loaded after 3 attempts, Service Guide assumes the Partner Application is down and abandons attempts to load it. Service Guide notifies the agent about the failure to load the partner application.

Operation: agentStatusUpdated

This table provides details of the agentStatusUpdated operation:

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details
Description	This operation is used by the partner application to notify Service Guide of the agent's availability to take an incoming call. The Service Guide displays one of these agent statuses, depending upon the payload when the operation is invoked: • Ready • Not Ready When an agent is actively handling a call, the status is Not Ready. Be aware that the partner application must invoke this operation to notify Service Guide when the agent should be moved to Not Ready status. Depending on the CTI vendor, this could be an explicit notification that the vendor sends to the partner application or it could be done by the partner application in response to when a call is connected.

Payload	`{`
	`  data: {`
	`  isAgentReady: boolean`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Parameter Description	**isAgentReady**: Whether the agent is currently ready to receive calls.
Error Parameter Description	If the Partner Applications receives an error while updating the agent status, it notifies Service Guide about the error by including an "error" object in the payload instead of including a "data" object. The error object has the following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occured. Possible values: – "**READY_STATUS_UPDATE_FAILED**" – "**NOT_READY_STATUS_UPDATE_FAILED**"
Notes	None

Operation: callReceived

This table provides details of the callReceived operation:

Operation Component	Details
Description	When an incoming call is received from the CTI vendor and sent to the partner application, invoke this operation to notify Service Guide about the incoming call.

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details

Payload	`{`
	`  data: {`
	`  phoneNumber: string,`
	`  callId: string`
	`  }`
	`}`

Parameter Description	**phoneNumber**: Incoming customer phone number. **callId**: A unique identifier representing the new incoming call session.
Notes	None

Operation: initiatedCallSession

This table provides details of the initiatedCallSession operation:

Operation Component	Details
Description	To be invoked when the new call session is established with the customer in response to an incoming call. Note that the agent will not explicitly accept an incoming call, but must be automatically connected to the call using auto-answer capabilities. Depending upon the CTI vendor, this could be an explicit setting in the CTI backend, or the Partner Application might be required to invoke an explicit vendor API to accept the call when it is received. Service Guide does not display Accept or Reject options for the agent. In response to this operation, Service Guide will render the call controls that are needed for the agent to interact with the active call session.

Payload	`{`
	`  data: {`
	`  phoneNumber: string,`
	`  callId: string,`
	`  connectedTime: integer | string | Date`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details
Parameter Description	**phoneNumber**: Customer phone number. **callId**: A unique identifier representing the newly established call session. **connectedTime**: An optional parameter representing the start of the conversation. If present, it can be an integer representing UNIX timestamp OR a String representing datetime in ISO-8601 OR a javascript Date object.
Error Parameter Description	If the partner applications receives an error while establishing the call session with the customer and agent, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**CONNECT_FAILED**".
Notes	Both the callReceived and initiatedCallSession operations expect a callId parameter. This is required because a CTI vendor might not generate a new call identifier until a call is connected. Therefore, the call identifier might not be available when the callReceived operation is invoked. The "connectedTime" paramater can be used if the call was already in progress via another CTI software before the Service Guide window was opened. In this case, it is useful to send the timestamp from the beginning of the conversation so that Service Guide can correctly reflect the call length instead of starting the timer from 0.

Operation: endCallSession

This table provides details of the endCallSession operation:

Operation Component	Details
Description	The Partner Application can invoke this operation when the call is disconnected from the CTI vendor. Note that if the agent disconnects the call in the Service Guide interface, this operation should be invoked when the call is actually disconnected by the CTI vendor in response to the Agent disconnect request.

Payload	`{`
	`  data: {`
	`  callId: string,`
	`  callDuration: number (optional)`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details
Parameter Description	**callId**: A unique identifier for the call session that ended. **callDuration**: Optional call duration in seconds.
Error Parameter Description	If the partner applications receives an error while disconnecting the call session, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**DISCONNECT_FAILED**".
Notes	None

Operation: callMuted

This table provides details of the callMuted operation:

Operation Component	Details
Description	This operation should be invoked to notify Service Guide that a request to mute an ongoing call was successfully processed. This operation can also be used to notify Service Guide if the call was externally muted so that Service Guide can update itself.

Payload	`{`
	`  data: {`
	`  callId: string`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Parameter Description	**callId:** A unique identifier for the call that was muted.
Error Parameter Description	If the partner applications receives an error while muting the call, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**MUTE_FAILED**".
Notes	None

Copyright © 2026, Oracle and/or its affiliates.

Operation: callUnmuted

This table provides details of the callUnmuted operation:

Operation Component	Details
Description	This operation should be invoked to notify Service Guide that a request to un-mute an ongoing call was successfully processed. It can also be used to notify Service Guide if the call was externally un-muted so that Service Guide can update itself.

Payload	`{`
	`  data: {`
	`  callId: string`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Parameter Description	**callId:** A unique identifier for the call that was un-muted.
Error Parameter Description	If the partner applications receives an error while unmuting the call, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**UNMUTE_FAILED**".
Notes	None

Operation: callOnHold

This table provides details of the callOnHold operation:

Operation Component	Details
Description	This operation should be invoked to notify Service Guide that a request to put the call on hold was successfully processed. It can also be used to notify Service Guide if the call was externally put on hold so that Service Guide can update itself.

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details

Payload	`{`
	`  data: {`
	`  callId: string`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Parameter Description	**callId:** A unique identifier for the call that is placed on hold.
Error Parameter Description	If the partner applications receives an error while putting the call on hold, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**HOLD_FAILED**".
Notes	None

Operation: callResumed

This table provides details of the callResumed operation:

Operation Component	Details
Description	This operation should be invoked to notify Service Guide that a request to resume a call that was previously put on hold was successfully processed. It can also be used to notify Service Guide if the call was externally resumed so that Service Guide can update itself.

Payload	`{`
	`  data: {`
	`  callId: string`
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Parameter Description	**callId:** A unique identifier for the call that is resumed.

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details
Error Parameter Description	If the partner applications receives an error while resuming the call that was previously put on hold, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**RESUME_FAILED**".
Notes	None

Operation: transcriptReceived

This table provides details of the transcriptReceived operation:

Operation Component	Details
Description	This operation should be invoked in response to the request transcript event to send the text transcript of the call. The partner application should request the transcript from the CTI backend via CTI-specific APIs. It is not necessary to fetch the transcript from the CTI backend at the point when the call ends. If the appropriate CTI vendor APIs are available, a real-time transcript can be collected during the call and held in front-end memory, which then can be used at the end of the call.

Payload	`{`
	`  data: {`
	`  agentTranscripts: array<{`
	`  channel: string,`
	`  text: string,`
	`  offsetMs: number }> (optional),`
	
	`  customerTranscripts: array<{`
	`  channel: string,`
	`  text: string,`
	`  offsetMs: number }> (optional),`
	
	`  combinedTranscripts: array<{`
	`  channel: string,`
	`  text: string,`
	`  offsetMs: number }> (optional)`
	
	`  },`
	
	`  error: object<{`
	`  isError: boolean,`
	`  message: string }> (optional)`
	`  }`
	`}`

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details
Parameter Description	**agentTranscripts:** An array of collected text transcripts for the agent's side of the conversation. Each element consists of the `channel` , `text` and `offsetMS`. **customerTranscripts:** An array of collected text transcripts for Customer's side of conversation. Each element consists of the `channel` , `text` and `offsetMS`. **combinedTranscripts:** An array of collected text transcripts for the whole conversation. Each element consists of the `channel` , `text` and `offsetMS`. **channel:** Either `'internal'` or `'external'` **text:** Actual speech to text transcript **offsetMs:** Offset in milliseconds from the conversation start time.
Error Parameter Description	If the partner applications receives an error while collecting the call transcript from CTI backend, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**TRANSCRIPTION_FAILED**".
Notes	Either **only** `combinedTranscripts` **OR both** agentTranscripts and customerTranscripts must be included in the payload. The `channel` property should have one of these values, based on who the conversation owner is: **`channel='internal'`** All elements of `agentTranscripts` OR those elements in `combinedTranscripts` that belong to agent. **`channel='external'`** All elements of `customerTranscripts` OR those elements in `combinedTranscripts` that belong to customer.

Operation: callWrappedUp

This table provides details of the callWrappedUp operation:

Operation Component	Details
Description	To be invoked when the call wrap up has been completed on the CTI side and the Service Guide UI should be notified so that it can reset itself, making the agent available for the next call.

Copyright © 2026, Oracle and/or its affiliates.

Operation Component	Details

Payload	<pre>{
	  data: {
	  callId: string
	  },
	  error: object<{
	  isError: boolean,
	  message: string
	  }> (optional)
	}</pre>

Parameter Description	**callId:** A unique identifier for the call for which the wrapup was done.
Error Parameter Description	If the partner applications receives an error while capturing the wrap up information on CTI side, it can notify Service Guide about the error by including the "**error**" object in the payload instead of including the "data" object. Error object has following properties: • **isError**: Boolean representing whether or not an error occurred. • **message**: An error code representing the type of error that occurred. Possible values are "**WRAPUP_FAILED**".
Notes	None

Operation: loggedOut

This table provides details of the loggedOut operation:

Operation Component	Details
Description	This operation should be invoked in response to the "logout" event to notify the Service Guide that the partner application has finished performing any cleanup in response to the "logout".
Payload	None
Parameter Description	Not Applicable
Notes	None

Events

When the Call Summary and Call Tagging user interface needs to request data or perform an action in the CTI backend, it raises an SDK Event that the Partner Application must listen for.

For example, when an agent puts a call on hold in the Call Summary and Call Tagging user interface , the application would raise an event to request that the Partner Application place the call on hold. Similarly, when agent requests to summarize the call, the application requests the text transcript from the Partner Application by raising an event.

The Partner Application listens for these events and performs the requested action, typically using CTI-provided APIs, and responds by invoking appropriate operations with the requested data.

Supported events include:

Copyright © 2026, Oracle and/or its affiliates.

- serviceGuideCti:agent:status:update
- serviceGuideCti:call:disconnect
- serviceGuideCti:call:mute
- serviceGuideCti:call:unmute
- serviceGuideCti:call:hold
- serviceGuideCti:call:resume
- serviceGuideCti:request:transcript
- serviceGuideCti:call:wrapUp
- serviceGuideCti:agent:logout

To streamline the event emitting mechanism and provide a consistent way to respond to these events, the Service Guide SDK is defined as an instance of the `EventEmitter` class. Additionally, all of these events are wrapped inside a single event that is emitted by the SDK and targeted to the CTI iFrame. Implementers can listen for a single event and inspect the event details to know which specific event has been requested.

> **Note**
>
> To minimize data entry errors, the Service Guide SDK already defines the string constants for these event names so that implementers don't need to enter the specific names.

This example shows how to handle events using an SDK instance that was initialized in an earlier code snippet and shows how to invoke another operation on the SDK in response to an event:

```
// SDK Loading and Initialization code
...

// Assuming the SDK instance is named "serviceGuideSdk"

// Read all CTI specific event names
const events = serviceGuideSdk.getCtiSdkEvents();

// "serviceGuideSdk.getSdkEventName()" gives the single, wrapper event name.
// Since the SDK is an "EventEmitter" instance, you can directly call ".on()"
function to subscribe to it.

serviceGuideSdk.on(serviceGuideSdk.getSdkEventName(), async ({eventName,
payload}) => {

    // "eventName" will contain the actual name of the event listed above
that was emitted by Service Guide.

     switch (eventName) {

          case events.updateAgentStatus: // Refers to the constant defined by
Service Guide SDK for the "serviceGuideCti:agent:status:update" event.
               // Assuming CTI vendor has below API
               if(payload.data.isAgentReady) {
                 try {
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    const response = await
ctiSdk.ApiClient.markUserReady({...});
                    if(response.ok) {
                       // Now let Service Guide know that the agent status has
been updated to "ready"
                       serviceGuideSdk.Ui.agentStatusUpdated({ data:
{isAgentReady: true} });
                    }
                  } catch (error) {
                    // Error occurred while updating the agent status in the
backend. Let Service Guide know with an error payload.
                    serviceGuideSdk.Ui.agentStatusUpdated({ error: {isError:
true, message: 'READY_STATUS_UPDATE_FAILED'} });
                  }
                } else {
                  try {
                    const response = await
ctiSdk.ApiClient.markUserBusy({...});
                    if(response.ok) {
                       // Now let Service Guide know that the agent status has
been marked as "busy"
                       serviceGuideSdk.Ui.agentStatusUpdated({ data:
{isAgentReady: false} });
                    }
                  } catch (error) {
                    // Error occurred while updating the agent status in the
backend. Let Service Guide know with an error payload.
                    serviceGuideSdk.Ui.agentStatusUpdated({ error: {isError:
true, message: 'NOT_READY_STATUS_UPDATE_FAILED'} });
                  }
                }
                break;

           case events.disconnectCall:
                     // disconnect the call and invoke "endCallSession"
operation to let Service Guide know.
                break;

           case events.requestTranscript:
                // Assuming CTI vendor has below API to get Agent transcript
                const agentTranscript = await
ctiSdk.ApiClient.getInternalTranscript(payload.data.callId);

                // Assuming CTI vendor has below API to get Customer
transcript
                const customerTranscript = await
ctiSdk.ApiClient.getExternalTranscript(payload.data.callId);

                // map the CTI provided transcript in to the necessary format
that Service Guide understands,
                // i.e. an array of elements containing "text", "offsetMs" and
"channel" properties

                const mappedAgentTranscript =
mapToOperationFormat(agentTranscript);
                const mappedCustomerTranscript =
```


Copyright © 2026, Oracle and/or its affiliates.

```
mapToOperationFormat(customerTranscript);

               // Provide the transcript to Service Guide as part of
"transcriptReceived" operation
               serviceGuideSdk.Ui.transcriptReceived({
                    data: {
                        agentTranscripts: mappedAgentTranscript,
                        customerTranscripts: mappedCustomerTranscript
                    }
               });
               break;
          ...
     }
});

// You can also take modularized approach and define multiple "handlers" to
handle specific event

serviceGuideSdk.on(serviceGuideSdk.getSdkEventName(),
handleAgentStatusUpdate);
serviceGuideSdk.on(serviceGuideSdk.getSdkEventName(),
handleTranscriptGeneration);
...

// and define individual handlers above as
const handleAgentStatusUpdate = async ({eventName, payload}) => {
    if (eventName === events.updateAgentStatus) {
        ... // update agent status and let Service Guide know.
    }
});

const handleTranscriptGeneration = async ({eventName, payload}) => {
    if (eventName === events.requestTranscript) {
        ... // Retrieve the transcripts and send them to Service Guide.
    }
});
```

Event: serviceGuideCti:agent:status:update

This table provides additional details about the serviceGuideCti:agent:status:update event:

Description	Payload	Parameter Description	Notes

This event is raised when an agent updates their status to Ready or Not Ready, indicating whether they are ready to receive another call.
The Partner Application should call the appropriate CTI-specific API to mark the agent's presence as defined in the payload.
	`{`
	`  data: {`
	`  isAgentReady: boolean`
	`  }`
	`}`
		isAgentReady:
		Whether the agent is ready to receive calls.
			The partner application should respond to this event by invoking "**agentStatusUpdated**" operation. This should be done after receiving a confirmation from the CTI backend that the update was successful, otherwise invoke it with an error payload.

Copyright © 2026, Oracle and/or its affiliates.

Event: serviceGuideCti:call:disconnect

This table provides additional details about the serviceGuideCti:call:disconnect event:

Description	Payload	Parameter Description	Notes
This event is raised by Service Guide when the agent disconnects the call.	`{ data: { callId: string } }`	**callId:** A unique identifier representing the call session that needs to be disconnected.	The partner application should respond to this event by invoking "**endCallSession**" operation. This should be done after receiving a confirmation from the CTI backend that the call was actually disconnected, otherwise invoke it with an error payload.

Event: serviceGuideCti:call:mute

This table provides additional details about the serviceGuideCti:call:mute event:

Description	Payload	Parameter Description	Notes
This event is raised by Service Guide when the agent wants to mute their audio on the call.	`{ data: { callId: string } }`	**callId:** A unique identifier for the call to be muted.	The partner application should respond to this event by invoking "**callMuted**" operation. This should be done after receiving a confirmation from the CTI backend that the call was actually muted, otherwise invoke it with an error payload.

Event: serviceGuideCti:call:unmute

This table provides additional details about the serviceGuideCti:call:unmute event:

Description	Payload	Parameter Description	Notes
This event is raised by Service Guide when the agent wants to reconnect their audio after it has been muted.	`{ data: { callId: string } }`	**callId:** A unique identifier for the call to be un-muted.	The partner application should respond to this event by invoking "**callUnmuted**" operation. This should be done after receiving a confirmation from the CTI backend that the call was actually unmuted, otherwise invoke it with an error payload.

Copyright © 2026, Oracle and/or its affiliates.

Event: serviceGuideCti:call:hold

This table provides additional details about the serviceGuideCti:call:hold event:

Description	Payload	Parameter Description	Notes
This event is raised by Service Guide when the agent places the customer on hold.	`{ data: { callId: string } }`	**callId:** A unique identifier for the call to be placed on hold.	The Partner Application should respond to this event by invoking "**callOnHold**" operation. This should be done after receiving a confirmation from the CTI backend that the call was actually placed on hold, otherwise invoke it with an error payload.

Event: serviceGuideCti:call:resume

This table provides additional details about the serviceGuideCti:call:resume event:

Description	Payload	Parameter Description	Notes
This event is raised by Service Guide when the agent removes the hold placed on the call.	`{ data: { callId: string } }`	**callId:** A unique identifier for the call to be resumed.	The partner application should respond to this event by invoking "**callResumed**" operation. This should be done after receiving a confirmation from the CTI backend that the call that was previously put on hold was actually resumed (hold was lifted), otherwise invoke it with an error payload.

Event: serviceGuideCti:request:transcript

This table provides additional details about the serviceGuideCti:request:transcript event:

Copyright © 2026, Oracle and/or its affiliates.

Description	Payload	Parameter Description	Notes

This event is raised by Service Guide when an agent requests the call transcript.	{
	data: {
	callId: string
	}
	}
		callId:
		A unique identifier for the call to be answered.
			The Partner Application should respond to this event by invoking "**transcriptReceived**" operation. This should be done after collecting the relevant transcript from the CTI backend. If for some reason, the transcript cannot be collected, the partner application should invoke the operation with an error payload.

Event: serviceGuideCti:call:wrapUp

This table provides additional details about the serviceGuideCti:call:wrapUp event:

Description	Payload	Parameter Description	Notes

After ending the call, when the agent finishes the after-call work and is ready to move to the next call, this event allows the partner application to capture any after-call work details (wrap up) in the CTI backend.
Depending upon the CTI Vendor, this might not be a mandatory operation. Utilities should handle this event after consulting the CTI Vendor documentation.
	{
	data: {
	callId: string,
	agentNotes: string,
	tags: array<string>
	}
	}
		callId: A unique identifier for the call for which the wrapup should be done.
		agentNotes: Any notes an agent might have collected during the call. This is included to capture any after-call notes (if the CTI Vendor makes it mandatory). This is optional and may or may not be present.
		tags: An array of strings representing the call topics that the agent wants to capture as part of wrap up. These are optional and may or may not be present.
			The partner application should respond to this event by invoking "**callWrappedUp**" operation. This should be done after receiving a confirmation from the CTI backend that the wrapup update was successful. If the update was not successful, the partner application should invoke the operation with an error payload.

Event: serviceGuideCti:agent:logout

This table provides additional details about the serviceGuideCti:agent:logout event:

Copyright © 2026, Oracle and/or its affiliates.

Description	Payload	Parameter Description	Notes
This event is raised when an agent logs out of the Service Guide. The partner application may wish to perform cleanup as part of logout or it may wish to perform a logout of its own to end the authenticated session it may have established at the beginning. This event should be used to initiate that logout and cleanup process.	None	Not Applicable	The Partner Application should respond to this event by invoking "**loggedOut**" operation, only after necessary cleanup was done. Service Guide application will wait a certain amount of time to let the partner application finish the cleanup. The partner application must invoke the "**loggedOut**" operation within that time. If not done, Service Guide will assume that the cleanup was already done and will proceed to complete the logout process on the Service Guide side.

Configuring Characteristic Types and Customer Contact Types with Characteristic Tags

The tags generated by the Call Summary and Call Tagging functionality are based on characteristic types to which the "Characteristic Tag" characteristic entity has been assigned.

In order to apply tags (characteristic types) to customer contacts, the appropriate characteristic types must be created and configured on the appropriate customer contact types. This involves the following:

1. In the **Characteristic Type** portal, add the following **Characteristic Entities** to any characteristic types to be used as tags:
   - Customer Contact
   - Characteristic Tag
2. In the **Customer Contact Type** portal, add the appropriate characteristic types to each customer contact type.

For example, if your implementation includes a "General Inquiry" customer contact type as well as "Billing Inquiry", "Payments", and "Rates" tags (characteristic types), you would do the following:

1. Created "Billing Inquiry", "Payments", and "Rates" characteristic types. The Value Type for tag characteristic values must be "Adhoc Value"
2. Add the "Customer Contact" and "Characteristic Tag" characteristic entities to each characteristic type created in step 1.
3. Create a "General Inquiry" customer contact type.

Copyright © 2026, Oracle and/or its affiliates.

4. Add the "Billing Inquiry", "Payments", and "Rates" characteristic types to to the "General Inquiry" customer contact type.

See Characteristic Types, Setting Up Customer Contact Types, and Customer Contact Tags for more information.

Enabling the Service Guide Zone

To enable access to the **Service Guide** zone, add the Service Guide Zone (C1-SVCGUIDE) application service to the appropriate user groups.

Analytics Integration Overview

Oracle Utilities Customer to Meter provides tools to facilitate the integration with Oracle Utilities Analytics.

- Analytics Configuration

Analytics Configuration

This section describes functionality provided for integrating with Oracle Utilities Analytics (OUA).

- **Customer Analytics Configuration Portal**

  Oracle Utilities Customer to Meter provides a Customer Analytics Configuration portal that holds information on the customer analytics configuration activities. This portal provides a summary of how much configuration has been set up, and also provides links and guidelines for the areas that need configuration, at the minimum, to successfully extract data from Oracle Utilities Customer Care and Billing to OUA. The following provides more information about this functionality.

  1. **Customer Analytics Configuration Details** – These data load parameters are used to extract, load, and transform data from Oracle Utilities Customer Care and Billing to Oracle Utilities Analytics. These parameters are used to identify and/or filter data when loading them into the data warehouse. Refer to the *Oracle Utilities Customer Care and Billing* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each extract parameter.
  2. **Bucket Configuration List** – This is a summary of the bucket configurations that are needed to classify and group extracted data within Oracle Utilities Analytics. The bucket configurations provide the ability to readily report and analyze data by buckets, or groups. Refer to the *Oracle Utilities Customer Care and Billing* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each bucket configuration supported by Oracle Utilities Customer Care and Billing.

- **Characteristic Mapping**

  Oracle Utilities Customer Care and Billing has identified the objects for which characteristic extraction is supported. This includes mapping the source characteristic entity to thetarget dimension, and defining which target columns are allowed to be mapped for which target dimension. The following table shows the supported mapping.

Object	Supported Target Dimension	Supported Target Columns

Copyright © 2026, Oracle and/or its affiliates.

Account Characteristic	**Account Dimension**	**User Defined Fields 6 - 15**
Person Characteristic	**Person Dimension**	**User Defined Fields 1 - 6**
Premise Characteristic	**Premise Dimension**	**User Defined Fields 6 - 18**
Premise Characteristic	**Address Dimension**	**User Defined Fields 7 - 16**
Service Agreement Characteristic	**Service Agreement Dimension**	**User Defined Fields 9 - 19**

Meter Analytics Configuration Portal

The Meter Analytics Configuration portal holds information on the meter data analytics configuration activities. This portal provides a summary of how much configuration has been set up, and also provides links and guidelines for the areas that need configuration, at the minimum, to successfully extract data from Oracle Utilities Meter Data Management to OUA. The following provides more information about this functionality.

- **Analytics - Oriented Master Configuration:** These data load parameters are used to extract, load, and transform data from Oracle Utilities Meter Data Management to Oracle Utilities Analytics. They are used to identify and/or filter data when loading them into the data warehouse. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on each extract parameter
- **Analytics - Oriented Extendable Lookup List:** The meter data analytics related Extendable Lookup Business Objects are listed in this section. It provides a navigation link to the extendable Lookup Maintenance portal where the lookup values can be configured. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for detailed information on meter data analytics related extendable lookups.
- **Service Point Configuration List:** This lists the service point types within the system and indicates for each whether or not the meter analytics configuration has been setup. It provides a navigation link to the service point type where the necessary configuration can be setup or modified. Refer to the *Oracle Utilities Meter Data Management* section in the ***Configuring Oracle Utilities Analytics*** chapter of the *Oracle Utilities Analytics Administration Guide* for further information.

Batch Controls

Batch Code	**Name / Description**
D1-SPSDL	**Service Point Snapshot Download**
D2-DUGDL	**Usage Subscription Usage Group Download**
D2-MVREF	**Materialized View Refresh**
D2-USGDL	**Service Point Usage Snapshot Download**
D2-UUSDL	**SP Unreported Usage Snapshot Download**
D2-VEEDL	**SP VEE Exception Snapshot Download**

References

Refer to the following documentation for more information about Oracle Utilities Analytics:

- *Oracle Utilities Analytics Admin Guide*
- *Oracle Utilities Analytics Dashboards for Oracle Utilities Customer Analytics, Revenue Analytics and Credit & Collections Analytics Metric Reference Guide*

Copyright © 2026, Oracle and/or its affiliates.

- *Oracle Utilities Analytics Dashboards for Oracle Utilities Exception Analytics Metric Reference Guide*
- *Oracle Utilities Analytics Dashboards for Oracle Utilities Meter Data Analytics Metric Reference Guide*
- *Oracle Utilities Extractors and Schema for Oracle Utilities Customer Care and Billing Data Mapping Guide*
- *Oracle Utilities Extractors and Schema for Oracle Utilities Meter Data Management Data Mapping Guide*

Self-Service Integration

The following sections describe functionality provided for the integration with self-service applications such as Oracle Utilities Customer Self Service and Oracle Utilities Digital Self Service.

- About Self-Service Tasks
- Setting Up Self-Service Task Configuration
- Maintaining Self-Service Tasks
- Self-Service Start/Stop/Transfer Overview
- Self-Service Start/Stop/Transfer Operations
- Configuring Rate Compare Usage Adjustment Profiles

About Self-Service Tasks

Self-service tasks hold information about self-service operations that customers initiate from an integrated self-service application, such as Oracle Utilities Customer Self-Service and Oracle Utilities Digital Self Service.

A few examples of self-service tasks are:

- One time payments
- Account verification
- Start/Stop requests
- Auto pay setup
- Creating customer meter readings

Refer to the Oracle Utilities Customer Self-Service documentation or Oracle Utilities Digital Self Service documentation for more details on supported self-service operations.

The following sections highlight self-service integration functionality.

- Self-Service Task Type Defines Self-Service Task Behavior
- Creating Self-Service Tasks

Copyright © 2026, Oracle and/or its affiliates.

Self-Service Task Type Defines Self-Service Task Behavior

Each type of self-service operation has specific processing requirements.

Self-service task type controls how a self-service task is processed. It defines the business object to use when creating the self-service task record. The business object defines the lifecycle of self-service task record.

The self-service task type can also specify common configuration data, such as settings for error handling.

Creating Self-Service Tasks

The customer self-service application invokes inbound web services to create self-service tasks. The system supplies an inbound web service for each supported self-service operation.

Refer to the Oracle Utilities Customer Self-Service documentation or Oracle Utilities Digital Self Service documentation for more details on supplied inbound web services.

Each inbound web service references a service script that processes the specific self-service operation. Refer to the service script system data for processing details.

Setting Up Self-Service Task Configuration

- Self-Service Integration Configuration
- Defining Self-Service Task Types

Self-Service Integration Configuration

The integration with a self-service application such as Oracle Utilities Customer Self Service or Oracle Utilities Digital Self Service requires the setup of master configuration data that controls the processing of your self-service operations.

To set up self-service integration master configuration, navigate using **Admin**, and then **General**, and then **Master Configuration.** Two master configurations are required:

- **CCB Self Service Integration:** This master configuration is used for the integration of customer data with a self service application.
- **MDM Self Service Master Configuration:** This master configuration is used for the integration of meter data with a self service application.

Refer to the embedded help for more details on how each section in these records is configured.

Defining Self-Service Task Types

A self-service task type defines properties that control how a self-service task is processed.

Refer to About Self-Service Tasks for an overview of service task functionality.

To maintain the types applicable to your product, open **Menu**, and then **Self-Service**, and then **Self-Service Task Type.**

This is a standard All-in-One portal.

The information captured on the self-service task type depends on the business objects supported by your product. Refer to the embedded help text for more information.

Copyright © 2026, Oracle and/or its affiliates.

Maintaining Self-Service Tasks

This section describes the functionality supported for viewing and maintaining self-service tasks.

Refer to About Self-Service Tasks for an overview of self-service task functionality.

Navigate using **Menu**, and then **Self-Service**, and then **Self-Service Task**, and then **Search**. You are brought to a query portal with options for searching for self-service tasks.

Once a self-service task record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Self-Service Task** zone provides basic information about a self-service task. Refer to the embedded help for more information.

Self-Service Start/Stop/Transfer Overview

> **Warning**
>
> This topic is only applicable for the integration with Oracle Utilities Digital Self Service.

The Start Stop Transfer Tasks (C1-StartStopTransferTasks) inbound web service is used by the self-service application to support the following start/stop/transfer transactions:

- New Service
- Add Service
- Stop Service
- Transfer Service

Distinct action method roles represent each type of transaction:

- Self-Service New Service
- Self-Service Add Service
- Self-Service Stop Service
- Self-Service Transfer Service

At the beginning of the start/stop/transfer process, the action method role is used to identify the type of transaction. It is determined using the Request Mode (Start/Stop/Transfer), the New Customer indicator (Yes/No) and the New Account indicator (Yes/No):

- If Request Mode is 'Start', the Action Method Role is 'Self-Service New Service' if either the New Customer or the New Account indicator is 'Yes'. If both New Customer and New Account indicators are 'No', Action Method Role is 'Self-Service Add Service'.
- If Request Mode is 'Stop', the Action Method Role is 'Self-Service Stop Service'
- If Request Mode is 'Transfer' , the Action Method Role is 'Self-Service Transfer Service'.

Configuration information are maintained using Self-Service Task Type and Customer Service Type. Self-Service Task Type defines the self-service task BO to be used. Customer Service Request Type defines the customer service request BOs used to persist data, and the controls

Copyright © 2026, Oracle and/or its affiliates.

used throughout the transaction. Refer to the "Enhanced Start, Stop, and Transfer Service Configuration Options" section of the digital self-service user guide for more about the controls.

At the beginning of the transaction, configuration information are determined and a self-service start/stop/transfer task is created. The same service task is used for the entire transaction. The service task is used to store the information received from the self-service application and operation processing results. It will remain in Pending status until the transaction is finalized. When the transaction is finalized, the service task is transitioned out of the Pending status and the collected information are persisted.

The following is the expected sequence of IWS operation calls per type of transaction:

New Service

1. Retrieve Request Type Details
2. Retrieve Service-Related Customer Details - operation is called if customer exists (Person ID is available).
3. Service Eligibility
4. Process Other Person
5. Clear Other Person
6. Determine Services To Start
7. Validate Start SP Selection
8. Service Questions
9. Process Services To Start
10. Deposit Assessment
11. Start Service Finalization

Add Service

1. Retrieve Request Type Details
2. Retrieve Service-Related Customer Details
3. Service Eligibility
4. Determine Services To Start
5. Validate Start SP Selection
6. Service Questions
7. Process Services To Start
8. Deposit Assessment
9. Start Service Finalization

Stop Service

1. Retrieve Request Type Details
2. Retrieve Service-Related Customer Details
3. Determine Services To Stop
4. Retrieve Services To Stop
5. Validate Stop Service Agreement Selection
6. Service Questions

Copyright © 2026, Oracle and/or its affiliates.

7. C1_98SelfServiceIntegration_Self_Service_Start_Stop_Transfer_Operations.dita#C1_98SelfServiceIntegration_Self_Service_Start_Stop_Transfer_Operations
8. Stop Service Finalization

Transfer Service

1. Retrieve RC1_98SelfServiceIntegration_Self_Service_Start_Stop_Transfer_Operations.dita#C1_98SelfServiceIntegration_Self_Service_Start_Stop_Transfer_Operationsequest Type Details
2. Retrieve Service-Related Customer Details
3. Service Eligibility
4. Determine Services To Start
5. Validate Start SP Selection
6. Service Questions - operation is called with Question Placement Group is 'Start Service'
7. Process Services To Start
8. Deposit Assessment
9. Determine Services To Stop
10. Retrieve Services To Stop
11. Validate Stop Service Agreement Selection
12. Service Questions - operation is called with Question Placement Group is 'Stop Service'
13. Process Services To Stop
14. Transfer Service Finalization

Self-Service Start/Stop/Transfer Operations

> **Warning**
>
> This topic is only applicable for the integration with Oracle Utilities Digital Self Service.

Retrieve Request Type Details

This operation is used to retrieve the following configuration information at the beginning of the start, stop and transfer transaction:

- Action Method Role
- Action Method
- Self-Service Task Type
- Customer Service Request Type
- Person/Business Controls
- Start Service Controls
- Stop Service Controls

Action Method Role is determined based on the Request Mode, New Customer indicator and New Account indicator inputs. The Action Method Role is used to determine the Action Method. The Action Method defines the Self-Service Task Type and Customer Service

Copyright © 2026, Oracle and/or its affiliates.

Request Type configuration. Note that the Self-Service Task Type and Customer Service Request Type can differ by CIS Division and/or Customer Class. The Self-Service Task Type defines the self-service task BO to be used. The Customer Service Request Type defines the customer service request BOs used to persist data, and the controls used throughout the transaction. Controls determine the behavior of the self-service application UI, operations, and service task processing. Succeeding operations expect the Self-Service Task Type and Customer Service Request Type as inputs.

Person/Business Controls are based on the input person/business flag. Start Service Controls are returned if the input Request Mode is either 'Start' or 'Transfer'. Stop Service Controls are returned if the input Request Mode is either 'Stop' or 'Transfer'.

Time Zone is returned as part of the Start Service and Stop Service Controls. It comes from one of the following sources:

1. Customer Service Request Type
2. Premise - Input premise if available. Otherwise, premise linked to the input account's most recent service.
3. Installation Options

Retrieve Service-Related Customer Details

This operation is used to retrieve the following service-related information for the input account and/or person:

- Name
- Mailing Address
- Contact Details

If the input account is populated, the input person is either the account owner (main customer) or a guest who was invited by the owner. Name, Mailing Address and Contact Details are based on the account's main customer. If input account is not populated, Name, Mailing Address and Contact Details are based on the input person. Phone and email contact information are always returned. Other contact types are returned based on the corresponding control.

This operation also uses the Should Prompt For ID controls to determine if ID should be prompted for the input account and/or person. If the input account is populated, the account's main customer is used. Otherwise, the input person is used. If ID should be prompted, the valid ID Types controls are returned for input person.

Finally, this operation indicates if a non-final self-service start/stop/transfer task exists for the input account.

Service Eligibility

This operation determines if a customer is eligible to start service. It is used when starting new service, adding service, or transferring service. If the customer is eligible, a new self-service start/stop/transfer task is created where the input customer information and the results of processing performed throughout the transaction are persisted. The service task will remain in Pending status until it is transitioned by the finalize operation. Abandoned service tasks are discarded automatically via a monitor algorithm on the Pending status. The service task ID is returned as well as the customer's eligible and ineligible accounts.

The Retrieve Request Type Details operation that determines configuration information should be called before this operation can be used.

Copyright © 2026, Oracle and/or its affiliates.

When starting new service for new customer (input person is not populated), the Check For Existing Person Algorithm control is used to determine if an existing person in the system matches the input name, ID, and contact information. If a match is found, the operation returns a match indicator, and no further processing is performed.

When starting new service or adding new service, person identity checks may be performed based on the ID check controls. The Should Prompt For ID controls are used to determine if ID should be prompted for the input person. This is the same step performed in the Retrieve Service-Related Customer Details operation and both should have the same result. It is necessary to do this step in this operation because the result from the Retrieve Service-Related Customer Details operation is not persisted.

The Start Service Eligibility Algorithms control determines eligibility. It uses the input information to this operation, the ID check results, and the ID prompt result. This control also determines the customer's eligible and ineligible accounts.

Process Other Person

This operation process additional persons when starting new service. The main customer's eligibility should be evaluated, and a pending self-service start/stop/transfer task already exists before this operation can be used. Eligible persons are persisted on the service task. Ineligible persons may be persisted based on the service task based on the Other Person Eligibility controls. Each person's eligibility is returned as output. Note that any additional persons that are currently in the self-service start/stop/transfer task are replaced with the input when this operation is called.

The Search For Person Algorithm control is used to find an existing person in the system that matches the input name, ID, and contact information. The search result (Match / No Match / Ambiguous Match) is returned as output. If the search resulted in an ambiguous match, the person is skipped. Person identity checks may be performed based on the ID check controls.

For new a new person, eligibility is based on ID check results. Note that if ID check is not performed, the person is considered eligible. Only eligible new persons are persisted on the service task. The Other Person Eligibility control determine the eligibility of an existing person. It is either based on ID check or a set of utility-specific eligibility algorithms. Eligibility based on ID check is evaluated the same way as a new person. Existing persons that are eligible are persisted on the service task. Existing persons that are ineligible are persisted on the service task if the Other Person Processing control is set to 'Review'.

Clear Other Person

This operation removes any additional persons that are currently in the pending self-service start/stop/transfer task.

Determine Services To Start

This operation determines the SP-based services to start at premise when starting new service, adding service, or transferring service. The main customer's eligibility should be evaluated, and a pending self-service start/stop/transfer task already exists before this operation can be used. It uses the Determine Services To Start Algorithm control to get a list of SA to start for SPs at the premise for the input start date. This information is stored in the service task's start service details. Note that SA and SP information currently in the service task are replaced when this operation is called. The operation returns the determined SPs along with badge number of the meter currently installed on the SP.

Note that the start date must be within the range defined by the start service controls. The range is calculated as follows:

- Start Date Low Limit = Current Date + Lead Days

Copyright © 2026, Oracle and/or its affiliates.

- Start Date High Limit = Start Date Low Limit + Maximum Start Days in Future control

Lead Days is initially set to the Lead Days control. If the current time is past the Cutoff Time control, an extra day is added to it. Also, weekends and holidays based on the Work Calendar control are skipped when incrementing the Start Date Low Limit.

Validate Start SP Selection

This operation uses the Validate SPs To Start Algorithms controls to validate the selected SPs where service is being started if Start Level control is 'Individual Services'. It is used when starting new service, adding service, or transferring service. The SP-based services to start at the premise should be determined before this operation can be used.

Service Questions

This operation uses the Read Questions algorithm on the CS request type BO to retrieve the applicable service questions when starting new service, adding service, stopping service, or transferring service. The SP-based services to start at a premise and/or the SP-based services to stop for an account should be determined, and a pending self-service start/stop/transfer task already exist before this operation can be used.

Note that when transferring service, this operation is called for two scenarios - start service and stop service. The operation uses the input Question Placement Group to distinguish between these calls. Question Placement Group is used to logically group questions, based on where the questions are shown on the user interface.

- If Question Placement Group is 'Start Service', the questions are filtered using the input premise and SPs.
- If Question Placement Group is 'Stop Service', the questions are filtered using the input premise only. Any further filtering (e.g. SP Type) will be done by the calling system using the applicability list that is returned for each question.

When invoking the algorithm, the input account and SPs are used. If the input Question Placement Group is 'Stop Service' and Stop Level control is 'Account', all premises in the service task's stop details are used. Otherwise, the input premises are used.

Process Services To Start

This operation marks which of the SPs and SP-based SAs previously determined by the Determine Services To Start operation will be started based on the input premise or SPs when starting new service, adding service, or transferring service. If Start Level control is 'Premise' all SPs and SAs are marked. If Start Level control is 'Individual Services' only the input SPs and its SAs are marked.

This operation also process response to start service questions using Process Question Responses routine. In this operation, the routine may change which SAs are marked for start, set an SA's start option, or add non SP-based SAs to the services being started.

All services being started are validated using Validate Service To Start Algorithms control.

Deposit Assessment

This operation determines the deposit requirements when starting new service, adding service, or transferring service. The main customer's eligibility should be evaluated, and a pending self-service start/stop/transfer task already exists before this operation can be used. It uses the deposit controls and the customer information in the service task to determine if deposit is required. If deposit is required, the deposit amount is calculated using the Recommendation Algorithm associated with the Deposit SA Type control's deposit class. The deposit required

Copyright © 2026, Oracle and/or its affiliates.

indicator and deposit amount are persisted on the service task and returned as output. The Allow Pay In Full control is also returned as output.

If the deposit is required, the operation returns deposit installation options collection. Each entry has two attributes:

- Number of Installments
- Installment Amount

Number of Installments is based on the Number of Installments control. The corresponding Installment Amount is the deposit amount divided by the Number of Installments. Note that Installment Amount must be greater than or equal to the Minimum Installment Amount control an option to be included in the collection. If Allow Pay In Full control is 'true', the following entry is included in the collection:

- Number of Installments = 1
- Installment Amount = Deposit Amount

The operation will fail if deposit is required and not one deposit installation option can be determined.

Start Service Finalization

This is the final operation called when starting new service or adding service. It is used to update additional information on the self-service start/stop/transfer task and to nudge it out of the pending status. The additional information includes:

- Number of Installments
- Main customer's contact information
- Paperless Billing switch
- Auto Pay Information
- Web User Information

Algorithms in the service task's lifecycle are responsible for persisting the collected information to the appropriate objects (person, account, premise, SA and SP). Refer to the self-service task BO (C1-SSStartStopXferServiceTask) for more information.

The operation returns one of the following task statuses:

- In Progress – Service Task is in a non-final status.
- Complete – Service Task is in the completed status.
- Error – Service Task is in an error status.

Processing errors are returned as a list of messages. If service task processing resulted in a new account being added, the account ID is returned as an output.

Determine Services To Stop

This operation uses the Determine Services To Stop Algorithm control to determine the SP-based services to stop for an account at a given stop date when stopping service or transferring service. It returns number of premises that have stoppable services.

This operation can be called read-only mode. This mode is used by the self-service application to determine if stop service is available as an option to the user. When not in read-only mode, a self-service start/stop/transfer task containing the determined SP-based services is created if one doesn't exist (initial call when stopping service). The service task will remain in Pending status until it is transitioned by the finalize operation. The ID of the newly created service task

Copyright © 2026, Oracle and/or its affiliates.

is returned as output. If service task already exists (transferring service or stop date is changed when stopping service), the service task ID is passed to this operation and its stop details are replaced with the determined SP-based services.

Note that the stop date must be within the range defined by the stop service controls. The range is calculated as follows:

- Stop Date Low Limit = Current Date + Lead Days
- Stop Date High Limit = Stop Date Low Limit + Maximum Stop Days in Future control

Lead Days is initially set to the Lead Days control. If the current time is past the Cutoff Time control, an extra day is added to it. Note that weekends and holidays based on the Work Calendar control are skipped when incrementing the Stop Date Low Limit.

Retrieve Services To Stop

This operation retrieves the premises with stoppable service that were previously determined and persisted on the self-service start/stop/transfer task. A list of SAs is returned for each premise. Each entry has two attributes:

- Service Type
- Badge Number of the meters currently installed on the SPs associated with the SA

Pagination is applied to the returned premise list using the page offset and page limit input parameters.

Validate Stop Service Agreement Selection

This operation uses the Stop Level, Require SAs At Same SP To Behave Together, and Validate Services To Stop Algorithm controls to validate the selected SP-based services to stop. It is used when stopping service or transferring service. The SP-based services to stop for the account should be determined before this operation can be used.

Process Services To Stop

This operation marks which of the premises and SP-based SAs previously determined by the Determine Services To Stop operation will be stopped based on the input premises and SAs when stopping service or transferring service. If Stop Level control is 'Account' all premises and SAs are marked. If Stop Level control is 'Premise', all SAs for the selected premises are marked. If Stop Level control is 'SA', only the selected SAs are marked.

This operation also process response to stop service questions using Process Question Responses routine. In this operation, the routine may change which SAs are marked for stop or add non SP-based SAs to the services being stopped.

All services being stopped are validated using the Validate Services To Stop Algorithms control.

Stop Service Finalization

This is the final operation called when stopping or transferring service. It is used to update the self-service start/stop/transfer task with any additional contact information for the main customer and to transition it out of the pending status.

Algorithms in the service task's lifecycle are responsible for persisting the collected information to the appropriate objects (person, account, premise, SA and SP).

The operation returns one of the following task statuses:

- In Progress – Service Task is in a non-final status.

Copyright © 2026, Oracle and/or its affiliates.

- Complete – Service Task is in the completed status.
- Error – Service Task is in an error status.

Processing errors are returned as a list of messages.

Transfer Service Finalization

This is the final operation called when transferring service. It is used to update additional information on the self-service start/stop/transfer task and to nudge it out of the pending status. The additional information includes:

- Number of Installments
- Main customer's contact information
- Paperless Billing switch
- Auto Pay Information
- Web User Information

Algorithms in the service task's lifecycle are responsible for persisting the collected information to the appropriate objects (person, account, premise, SA and SP).

The operation returns one of the following task statuses:

- In Progress – Service Task is in a non-final status.
- Complete – Service Task is in the completed status.
- Error – Service Task is in an error status.

Processing errors are returned as a list of messages. If service task processing resulted in a new account being added, the account ID is returned as an output.

Configuring Rate Compare Usage Adjustment Profiles

This section describes the steps involved in setting up usage adjustment profiles for use with the Rate Compare feature in Oracle Utilities Customer Self- Service.

1. Create a characteristic type and values for each type of usage adjustment to be made available for rate comparison purposes. For example, if you wanted to allow adjustments for installation of solar panels, use of an energy-efficient appliance, or use of an electric vehicle, you would create three characteristic types (one for each type of adjustment), and values for each option within each type (each type of appliance or electric vehicle supported).
2. Create profile (stand-alone) measuring components that correspond to each characteristic type value.
3. Create profile data for each measuring component. This is interval data that represents the impact of the usage adjustment on a customer's consumption.
4. Create a profile factor for each usage adjustment type.
5. Create factor values for each factor. These values correspond to the characteristic type value, and link characteristic type values to profile measuring components.
6. Create one (or more) Profile Accumulation usage calculation rules that will apply the usage adjustment profile to the customer's interval consumption when calculating usage for the rate comparison request.

Copyright © 2026, Oracle and/or its affiliates.

7. Create entries in the Customer Rate Schedule extendable lookup to associate the usage calculation group that contains the "Profile Accumulation" usage calculation rule to a rate schedule.
8. Create entries in the Usage Adjustment Types extendable lookup for each usage adjustment type.
9. Set up the Rate Compare Configuration section of the MDM Self-Service Master Configuration to link usage adjustment factors and usage adjustment types to rate schedules Oracle Utilities Customer Care and Billing (defined in the Customer Rate Schedule extendable lookup).

Characteristic Types

Create a characteristic type and values for each type of usage adjustment you wish to make available to customer self-service users. For example, to create a usage adjustment profile for use of an electric vehicle, you would create an "electric vehicle" characteristic type and define values for each type of electric vehicle users can select.

Example Characteristic Type:

- Characteristic Type: ELEC_VEH
- Description: Electric Vehicles
- Type of Char Value: Predefined Value
- Characteristic Values:

Characteristic Value	Description
LEAF	Nissan Leaf
TESLA	Tesla Model S
VOLT	Chevrolet Volt

Profile Measuring Components

Create profile measuring components for each characteristic type value. These measuring components will be used to store profile data for each type of usage adjustment.

Note: The base package does not include stand-alone measuring component/measuring component type business objects, but the demonstration database contains "Standalone Interval" and "Standalone Interval Measuring Component Type" business objects that can be used to create profile measuring components and types.

Example Profile Measuring Component Type:

- Measuring Component Type: KWH-PROFILE
- Description: KWH Profile
- Measuring Component Business Object: Standalone Interval (demo)
- Measurement Business Object: Measurement
- Service Type: Electric
- Allow Negative Consumption: Allowed
- Consumptive / Subtractive: Consumptive
- Seconds Per Interval: 01:00:00
- Value Identifiers:

Copyright © 2026, Oracle and/or its affiliates.

- Value Identifier Type: Measurement
- Short-Hand Description: kWh
- UOM: Kilowatt-Hours

Example "Nissan Leaf" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Nissan Leaf

Example "Tesla Model S" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Tesla Model S

Example "Chevrolet Volt" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Chevrolet Volt

Profile Data

Create profile data for each profile measuring component. This data represents the impact of the usage adjustment on a customer's consumption.

Note that this profile data can (and often will) include negative interval values to represent the difference in consumption applicable for the usage adjustment type. For example, if an energy-efficient electric clothes dryer uses an average of 30 kilowatt hours less per month than an average electric clothes dryer, profile data for that appliance might be a "straight line" hourly profile (a profile in which all intervals are of the same value) in which each value equals "0.042" (30 kWh per month divided by an average of 720 hours per month).

Copyright © 2026, Oracle and/or its affiliates.

Factors / Factor Values

Create a profile factor for each usage adjustment type. This factor should use the "Factor Characteristic Source N/A Algorithm" to derive the appropriate characteristic type value based on the factor value, and should reference the characteristic type created earlier.

Example Electric Vehicle Factor:

- **Factor**: ELECTRIC_VEHICLE
- **Description**: Electric Vehicle
- **Factor Class**: Profile
- **Characteristic Source Algorithm**: Factor Characteristic Source N/A Algorithm
- **Factor Characteristic Type**: Electric Vehicle

Example "Nissan Leaf" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Nissan Leaf, KWH Profile**

Example "Tesla Model S" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Tesla Model S, KWH - 60 Minutes**

Example "Chevrolet Volt" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Chevrolet Volt / KWH - 60 Minutes**

Profile Accumulation Usage Calculation Group and Rule

Create a usage calculation group that contains a "Profile Accumulation" usage calculation rule. This rule will calculate usage by accumulating historical usage with profile data. based on a selected profile factor value.

Note: Profile Accumulation rules should use eligibility criteria to ensure they are only executed when the "Calculation Mode" on the usage transaction is set to "Hypothetical Calculation" (D2HC).

Example: Profile Accumulation Usage Calculation Rule

- **Usage Calculation Group: Electric Residential Interval KWH**

Copyright © 2026, Oracle and/or its affiliates.

- **Usage Calculation Rule: KWH_PROFILE_ACCUMULATION**
- **Sequence: 10**
- **Description: KWH Profile Accumulation**
- **Category: Usage Calculation**
- **Vector Source Configuration:**
- **Vector Type**: Channels Linked to Usage Subscription
- **Unit of Measure**: Kilowatt-Hours
- **Time of Use**:
- **Service Quantity Identifier**:
- **Target SPI**: 01:00:00
- **Result Processing Configuration**:
- **Apply TOU Map to Derived Vector: Yes**
- **TOU Map: Summer / Winter, 15 minute interval**
- **Result Storage Configuration**:
- **Insert Primary SQ Entry:** Yes
- **Save Derived Vector:** No
- **Service Quantity Identifier:**
- **Extract Interval Data:** No

Customer Rate Schedule Extendable Lookup

Create an entry in the "Customer Rate Schedule" extendable lookup to associate the usage calculation group that contains the "Profile Accumulation" usage calculation rule to an applicable rate schedule in Oracle Utilities Customer Care and Billing.

Example Customer Rate Schedule extendable lookup:

- **Rate**: E-INT-RES
- **Description**: Electric Residential Interval Rate
- **Default Usage Calculation Group**: Electric Residential Interval KWH (E-INT-RES)

Usage Adjustment Types Extendable Lookup

Create an entry in the "Usage Adjustment Type" extendable lookup for each type of usage adjustment that will be available to customer self-service users. These entries are used in the Rate Compare Configuration section of the Self-Service Master Configuration (see below).

Example Electric Vehicle entry:

- **Usage Adjustment Type: ELEC_VEHICLE**
- **Description: Purchase of Electric Vehicle**
- **Override Description: Purchase of Electric Vehicle**
- **External Reference ID: Purchase of Electric Vehicle**

Copyright © 2026, Oracle and/or its affiliates.

MDM Self-Service Master Configuration - Rate Compare Configuration

Configure the "Rate Compare Configuration" section of the MDM Self-Service Master Configuration to associate usage adjustment factors and usage adjustment types with an applicable rate schedule in Oracle Utilities Customer Care and Billing (defined in the Customer Rate Schedule extendable lookup).

Example MDM Self-Service Master Configuration:

- **Factor Characteristic Type Indicating No Value Variation: N/A**
- **External Reference Factor Value Characteristic Type: External Reference ID**
- **Minimum Days of Usage Adjustment Data: 2**
- **Rate / Usage Adjustments:**
- **Rate**: Electric Residential - Interval KWH
- **Usage Adjustments**:
- Usage Adjustment Factor: Electric Vehicle
- Usage Adjustment Type: Purchase of Electric Vehicle

Defining DataConnect Options

Oracle Utilities DataConnect facilitates data extraction from Oracle Utilities applications such as Oracle Utilities Customer to Meter and Oracle Utilities Customer Care and Billing for use in external applications; such as, analytics applications and energy management systems.

This section describes how Oracle Utilities DataConnect works and how to implement and configure the system to support data extract processing.

> **Note**
>
> There are two versions of DataConnect. The initial version is referred to as “Legacy DataConnect”. The current and recommended version is simply referred to as “DataConnect” or “Current DataConnect”. Implementations are recommended to use the current version. “Legacy DataConnect” remains available for implementations that used the initial version, but is not recommended for new implementations.

- DataConnect Data Extracts
- Customer and Premise Data Extracts
- Characteristic Data Extracts
- Billing Data Extracts
- Consumption Data Extracts
- Extract Flat File Formats
- File Naming Formats

DataConnect Data Extracts

Oracle Utilities DataConnect can be used to extract the following types of data:

Copyright © 2026, Oracle and/or its affiliates.

- **Customer Data**

  (see Customer and Premise Data Extracts for more information)

  - Account information (including service agreement, account, and person related information)
  - Service agreement's service point information
  - Mailing address information
- **Premise Data**

  (see Customer and Premise Data Extracts for more information)

  - Premise information
  - Service point information
  - Device installation information
  - Device information
  - Measuring component information
- **Billing Data**

  (see Billing Data Extracts for more information)

  - Bill
  - Bill segment
  - Bill segment calculation lines
  - Bill segment service quantities
- **Measurement Data**

  (see Consumption Data Extracts for more information)

The data extracted can be correlated in external systems using the service agreement and/or service point data that is included in each type of extract. All data extracts contain a combination of the following data elements that can be used for this correlation:

- **CCB Service Point ID**: The service point ID in Oracle Utilities Customer to Meter and/or
- **MDM Service Point ID**: The service point ID in Oracle Utilities Meter Data Management and/or
- **Service Agreement ID:** The service agreement ID in Oracle Utilities Customer to Meter

External systems receiving data extracted from the Oracle Utilities DataConnect facilities can use these IDs to associate billing data and measurement data to extracted master data.

> **Note**
>
> There is a version 1 and version 1+ for the DataConnect extracts. In the following pages where a system object such as a batch control or data area is noted the objects are listed as (v1/v1+). When only a single object is noted it applies to both v1 and v1+.

Customer and Premise Data Extracts

Customer and premises data extracts are performed through the use of data synchronization, audit algorithms, business objects, and related batch processes.

Copyright © 2026, Oracle and/or its affiliates.

Batch processes are used to create initial load extracts for service agreement / account / person, service point, device configuration, device, and install event related master data. The initial load includes the historical data in the system (either all or a subset, depending on how the batch process is configured). Following the initial load, data synchronization requests are created when master data is added or changed.

- Customer and Premise Data Extracts: Maintenance Object Audit Algorithms
- Customer and Premise Data Extracts: Business Objects and Algorithms
- Customer and Premise Data Extracts: Batch Controls
- Setting Up and Executing Customer and Premise Data Extracts

Customer and Premise Data Extracts: Maintenance Object Audit Algorithms

Detecting changes in customer and premise data can be identified by audit algorithms on the following maintenance objects:

- Service Agreement
- Account
- Person
- Service Point (MDM Service Point)
- Install Event
- Device

The Generic Change Data Capture algorithm type (F1-GCHG-CDCP), which is provided in the base package, can be used to create audit algorithms for following maintenance objects:

- Service Agreement
- Service Point (MDM Service Point)
- Install Event

In addition, the following audit algorithm types can be used to create audit algorithms for customer and premise data:

- Account Change Data Capture (Customer Data Extract) (C1-ACCECDCCU)
- Person Change Data Capture (Customer Data Extract) (C1-PERECDCCU)
- Account Mailing Address Change Data Capture (C1ACCMACDC)
- Device Change Data Capture (SP/Meter-Based) (D1-SPDV-CDCP)
- Install Event Change Data Capture (SP/Meter-Based) (D1-SPIE-CDCP)
- Device Change Data Capture (SP/Meter-Based) (D1-SPDC-CDCP)

The following algorithm types are used with v1 ONLY.

- An **Account Change Data Capture (SA-Based) – Include Reactivated SAs** algorithm type (C1-ACCCDCSAR) is provided in the base package. This algorithm type can be used to create an audit algorithm on the Account maintenance object to determine if a service agreement sync request record is to be created for each of the account's non-closed, non-canceled service agreement(s). A change in a related account's details will instantiate a Service Agreement Extract Sync Request (of the type defined for the SA Sync Request BO algorithm parameter) if one does not already exist in the initial state for the service agreement. The base package also contains an algorithm (C1-ACCECDCSA) that is based on this algorithm type.

Copyright © 2026, Oracle and/or its affiliates.

- A **Person Change Data Capture (SA-Based) – Include Reactivated SAs** algorithm type (C1-ACCCDCSAR) is provided in the base package. This algorithm type can be used to create an audit algorithm on the Person maintenance object to determine if a service agreement sync request record is to be created for each of the person's non-closed, non-canceled service agreement(s). A change in the main person details linked to the service agreement's account will instantiate a Service Agreement Extract Sync Request (of the type defined for the SA Sync Request BO algorithm parameter) if one does not already exist in the initial state for the service agreement. The base package also contains an algorithm (C1-PERECDCSA) that is based on this algorithm type.

Customer and Premise Data Extracts: Business Objects and Algorithms

The maintenance object audit algorithms create data synchronization requests in the following manner:

- If the Generic Change Data Capture (F1-GCHG-CDCP) algorithm type (provided in the base package) is used, the synchronization requests are based on the Sync Request BO maintenance object options.
- Alternatively, if the base package supplied Maintenance Object – Audit algorithm types, which are outlined in the Customer and Premise Data Extracts: Maintenance Object Audit Algorithms section are used, the synchronization requests are based on the Sync Request BO defined in the algorithm parameters.

Customer Data

Extraction of customer data, including account, service agreement, and person information, is supported by the following data synchronization business objects.

- Current DataConnect
  - **Customer Data Extract for DataConnect** (C1-ExternalRepoPlusCusDataSync): used to extract service agreement / account / person related information. This business object should be defined as a value for the Sync Request BO option on the Service Agreement maintenance object.
  - **SA/SP Data Extract for DataConnect** (C1-ExternalRepoPlusSASPSync): used to extract service agreement service point information. This business object should be defined as a value for the Sync Request BO option on the Service Agreement maintenance object.
  - **Mailing Address Data for DataConnect** (C1-ExternalRepoPlusMailAddSync): used to extract mailing address information. This business object should be defined as a value for the "Sync Request BO" options on the Person maintenance object.
  - **SP/Meter Sync for DataConnect** (D1-ExternalRepositorySPMtrSync): used to extract service point and meter information. This business object should be defined as a value for the "Sync Request BO" options on the Service Point, Device Configuration, and Device maintenance objects.
- Legacy DataConnect
  - **SA/Account/Person Sync for Legacy DataConnect** (C1-ExternalRepoSASync): used to extract service agreement / account / person related information. This business object should be defined as a value for the Sync Request BO option on the Service Agreement maintenance object. (Used with v1)
  - **SP Sync for Legacy DataConnect** (D1-ExternalRepositorySPSync): used to extract service point information. This business object should be defined as a value for the "Sync Request BO" options on the Service Point maintenance object. (Used with v1)

Copyright © 2026, Oracle and/or its affiliates.

- **Install Event Sync for Legacy DataConnect** (D1-ExternalRepositoryIESync): used to extract install event information. This business object should be defined as a value for the "Sync Request BO" options on the Install Event maintenance object. (Used with v1)

These business objects use the following Pre-Processing algorithms to take initial data snapshots, and define the batch control used to extract data and export to a flat file:

- Current DataConnect
  - Capture Customer Data's Initial Snapshot for DataConnect(C1-CEXTRCSAD)
  - Capture SA/SP Data's Initial Snapshot (C1CAPSASPI)
  - Capture Mailing Address Data Initial Snapshot (C1CAPMLADDI)
- Legacy DataConnect
  - Capture SP/Meter Initial Snapshot for Legacy DataConnect (D1-SPMINISNP)
  - Capture SP Initial Snapshot for Legacy DataConnect (D1-SPEINISNP)
  - Capture SA-Based Initial Snapshot for Legacy DataConnect (C1-CEXTRCSAI)
  - Capture Install Event Initial Snapshot for Legacy DataConnect (D1-IEEINISNP)

These algorithms specify the batch control used for the extract process (see below).

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for other pending related synchronization requests, and, if none, transitions them to the Determine if Sync Needed state.

Enter algorithms on the Determine If Sync Needed states verify if a synchronization is needed by capturing the final snapshot of the data and comparing it against the initial snapshot. If changes are detected, the final snapshots of the data are stamped on the synchronization requests for export.

- Current DataConnect
  - Capture SA-Based Final Snapshot for DataConnect(C1-CEXTRPSAF)
  - Capture SA/SP Data's Final Snapshot (C1CAPSASPF)
  - Capture Mailing Address Data’s Final Snapshot (C1CAPMLADDF)
  - Capture SP and Meter Data's Final Snapshot for DataConnect (C1-CEXTRSPMF)
  - Capture SP/Meter Final Snapshot for DataConnect (D1-SPMFINSNP)
- Legacy DataConnect
  - Capture SA-Based Final Snapshot for Legacy DataConnect (C1-CEXTRCSAF)

The Prepare Delimited Extract Data (Ignore Custom Sync Records) Enter algorithm (C1-PRPEXDNCU/C1-PRTBDEXDT) or the Prepare Delimited Extract Data Enter algorithm (D1-PRPEXTDTA) on the Send Request state prepares and formats the data for extraction, and creates a general process record for the synchronization request (based on the batch control defined by the pre-processing algorithm).

Premise Data

Extraction of premise data, including premise, service point, device installation, device, and measuring component informatiob, is supported by the following data synchronization business objects.

- **SP/Meter Sync for DataConnect** (D1-ExternalRepositorySPMtrSync): used to extract service point and device related information. This business object should be defined as a

Copyright © 2026, Oracle and/or its affiliates.

value for the Sync Request BO option on the Service Point, Device Configuration, and Device maintenance objects.

These business objects use the following Pre-Processing algorithms to take initial data snapshots, and define the batch control used to extract data and export to a flat file:

- Capture SP/Meter Initial Snapshot for Legacy DataConnect (D1-SPMINISNP)

These algorithms specify the batch control used for the extract process (see below).

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for other pending related synchronization requests, and, if none, transitions them to the Determine if Sync Needed state.

Enter algorithms on the Determine If Sync Needed states verify if a synchronization is needed by capturing the final snapshot of the data and comparing it against the initial snapshot. If changes are detected, the final snapshots of the data are stamped on the synchronization requests for export.

- Capture SP/Meter Final Snapshot for DataConnect (D1-SPMFINSNP)

The Prepare Delimited Extract Data (Ignore Custom Sync Records) Enter algorithm (C1-PRPEXDNCU/C1-PRTBDEXDT) or the Prepare Delimited Extract Data Enter algorithm (D1-PRPEXTDTA) on the Send Request state prepares and formats the data for extraction, and creates a general process record for the synchronization request (based on the batch control defined by the pre-processing algorithm).

Customer and Premise Data Extracts: Batch Controls

Customer Data

The customer data extraction process uses a set of batch processes to create data synchronization requests and extract files.

The following batch processes are used to create initial synchronization requests:

- Current DataConnect
  - **Customer Data Initial Load for DataConnect (C1-CDTIL):** This batch control creates initial synchronization requests for account data (service agreements/accounts/persons) and SA's SP data (SA/SP). The appropriate sync business objects for the two extracts should be plugged in as parameters to the batch job.
  - **Mail Address Initial Load DataConnect (C1-MADIL):** This batch control creates initial synchronization requests for accounts.
- Legacy DataConnect
  - **SA-Based Initial Load for Legacy DataConnect (C1-SAEIL):** This batch control creates initial synchronization requests for service agreements / accounts / persons.
  - **SP Initial Load for Legacy DataConnect (D1-SPEIL):** This batch control creates initial synchronization requests for service points.

The following batch processes are used to create extract files from synchronization requests.

> **Note**
>
> These batch processes should be run to create initial load data synchronization requests based on the Sync for DataConnect business objects.

Copyright © 2026, Oracle and/or its affiliates.

- Current DataConnect
  - **Customer Data Extract for DataConnect (C1-CUDTX):** This batch control creates extract file(s) for service agreements / accounts / persons related information.
  - **SA/SP Data Extract for DataConnect (C1-SSDTX)**
  - **Mailing Address Extract DataConnect (C1-MADTX):** This batch control creates extract file(s) that contain mailing address information.
  - **SA-Based Extract for DataConnect (C1-CUDTX):** This batch control creates extract file(s) for service agreements / accounts / persons related information.
- Legacy DataConnect
  - **SP Extract for Legacy DataConnect (D1-SPESR):** This batch control creates extract file(s) that contain service point information.

These batch controls are defined as values for the Batch Control for Extract algorithm parameters on the Pre-Processing algorithms on the Sync for DataConnect business objects (see above).

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Premise Data

The premise data extraction process uses a set of batch processes to create data synchronization requests and extract files.

The following batch processes are used to create initial synchronization requests:

- Current DataConnect
  - **SP/Meter Initial Load for DataConnect (D1-SPMIL)**: This batch control creates initial synchronization requests for service points, device configurations, and devices.
- Legacy DataConnect
  - **Install Event Initial Load for Legacy DataConnect (D1-IEEIL):** This batch control creates initial synchronization requests for install events.

The following batch processes are used to create extract files from synchronization requests.

> **Note**
>
> These batch processes should be run to create initial load data synchronization requests based on the Sync for DataConnect business objects.

- Current DataConnect
  - **SP/Meter Extract for DataConnect (D1-SPMSR)**: This batch control creates extract file(s) that contain service point, device configuration, and device information.
- Legacy DataConnect
  - **Install Event Extract for Legacy DataConnect (D1-IEESR):** This batch control creates extract file(s) that contain install event information.

These batch controls are defined as values for the Batch Control for Extract algorithm parameters on the Pre-Processing algorithms on the Sync for DataConnect business objects (see above).

Copyright © 2026, Oracle and/or its affiliates.

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Setting Up and Executing Customer and Premise Data Extracts

Setting up master data extracts involves the following steps:

1. Add audit algorithms on the Service Agreement, Account, Person, MDM Service Point, Install Event, and Device maintenance objects.
    a. Add the Generic Change Data Capture algorithm as an audit algorithm on the Service Agreement, MDM Service Point, and Install Event maintenance objects.
    b. Add the Account Change Data Capture (SA-Based) - Include Reactivated SAs algorithm as an audit algorithm on the Account maintenance object.
    c. Add the Person Change Data Capture (SA-Based) - Include Reactivated SAs algorithm as an audit algorithm on the Person maintenance object.
    d. Add the Device Change Data Capture (Install Event-Based) algorithm as an audit algorithm on the Device maintenance object.
2. Add the DataConnect synchronization request business objects as Sync Request BO options on the Service Agreement, Service Point and Install Event maintenance objects.
3. Execute initial load batch processes related to service agreements, service points and install events.
4. Once the initial load synchronization has been executed, changes to service agreements, accounts, persons, service points, install events or related devices will trigger the creation of new synchronization requests and resulting extraction files.
5. Execute the sync request monitor process.
6. Execute the extract batch processes related to service agreements, service points and install events.

Characteristic Data Extracts

Characteristic data extracts are performed through the use of audit algorithms, data synchronization business objects, and related batch processes.

Batch processes are used to create initial load extracts for Account, Person, Premise, Service Agreement, and Service Point characteristic data. The initial load includes the historical data in the system (either all or a subset, depending on how the batch process is configured). Following the initial load, data synchronization requests are created when characteristic data is added or changed.

Maintenance Object Audit Algorithms

Detecting changes in characteristic data can be identified by the Characteristic Change Data Capture (C1MOCHARCDC) audit algorithm on the following maintenance objects:

- Account
- Person
- Premise
- Service Agreement

Copyright © 2026, Oracle and/or its affiliates.

- Service Point - CCB
- Service Point- MDM

Detecting changes in characteristic type data can be identified by the Characteristic Type Change Data Capture (C1CHARTYPCDC) audit algorithm on the Characteristic Type (CHAR TYPE) maintenance object.

Business Objects and Algorithms

The maintenance object audit algorithms create data synchronization requests based on the Sync Request BO defined in the audit algorithm parameters:

Audit Algorithm	Data Synchronization Request Business Object
Characteristic Change Data Capture (C1MOCHARCDC)	Characteristic Data Sync for DataConnect (C1-ExternalRepoPlusCharDatSync)
Characteristic Type Change Data Capture (C1CHARTYPCDC)	Characteristic Type Sync for DataConnect (C1-ExternalRepoPlusCharTypSync)

Extraction of characteristic data, including account, person, premise, service agreement, and service point information, is supported by the following data synchronization business objects.

- **Characteristic Data Sync for DataConnect** (C1-ExternalRepoPlusCharDatSync): used to extract characteristics data for accounts, persons, premises, service agreements, and service points. The specific maintenance objects for which characteristics data is extracted is defined in the Characteristic Entities Included in DataConnect Extract (C1-DataConnectIncludedCharEnty) extendable lookup. This lookup maps characteristic entities and maintenance objects to mappings business objects that define the characteristic data to be extracted.
- **Characteristic Type Sync for DataConnect** (C1-ExternalRepoPlusCharTypSync): used to extract data for characteristic types.

The Characteristic Data Sync for DataConnect (C1-ExternalRepoPlusCharDatSync) business object uses the following Enter algorithm on the Pending state to take initial data snapshots:

- Capture Characteristic Data's Initial Snapshot (C1CAPMOCHARI)

The Characteristic Type Sync for DataConnect (C1-ExternalRepoPlusCharTypSync) business object uses the following Pre-Processing algorithms to take initial data snapshots, and define the batch control used to extract data and export to a flat file:

- Capture Characteristic Type Data's Initial Snapshot (C1CAPCHTYI)

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for other pending related synchronization requests, and, if none, transitions them to the Determine if Sync Needed state.

Enter algorithms on the Determine If Sync Needed states verify if a synchronization is needed by capturing the final snapshot of the data and comparing it against the initial snapshot. If changes are detected, the final snapshots of the data are stamped on the synchronization requests for export.

- Capture Characteristic Data's Final Snapshot (C1CAPMOCHARF)
- Capture Characteristic Type Data's Final Snapshot (C1CAPCHTYF)

The Prepare Tab-delimited Extract Data (C1-PRTBDEXDT) algorithm on the Send Request state prepares and formats the data for extraction, and creates a general process record for the synchronization request (based on the batch control defined by the pre-processing algorithm).

Copyright © 2026, Oracle and/or its affiliates.

Batch Controls

The characteristic data extraction process uses a set of batch processes to create data synchronization requests and extract files.

The following batch processes are used to create initial synchronization requests:

- Account Characteristic Data Initial Load for DataConnect (C1-CHACI): This batch process creates initial sync request records for Account Characteristics.
- Person Characteristic Data Initial Load for DataConnect (C1-CHPEI): This batch process creates initial sync request records for Person Characteristics.
- Premise Characteristic Data Initial Load for DataConnect (C1-CHPRI): This batch process creates initial sync request records for Premise Characteristics.
- SA Characteristic Data Initial Load for DataConnect (C1-CHSAI): This batch process creates initial sync request records for Service Agreement Characteristics.
- SP - CCB Characteristic Data Initial Load for DataConnect (C1-CHSPI): This batch process creates initial sync request records for SP - CCB Characteristics.
- SP - MDM Characteristic Data Initial Load for DataConnect (X1-CHSPI): This batch process creates initial sync request records for SP - MDM Characteristics.
- Characteristic Type Data Initial Load for DataConnect (C1-CHTIL): This batch process creates initial Characteristic Type Data sync request records for the specified CHAR TYPE maintenance object

The following batch processes are used to create extract files from synchronization requests:

- Account Characteristic Data Extract for DataConnect (C1-ACCHX): This batch process creates Account flat file(s).
- Person Characteristic Data Extract for DataConnect (C1-PECHX): This batch process creates Person flat file(s).
- Premise Characteristic Data Extract for DataConnect (C1-PRCHX): This batch process creates Premise flat file(s).
- SA Characteristic Data Extract for DataConnect (C1-SACHX): This batch process creates service agreement flat file(s).
- SP - CCB Characteristic Data Extract for DataConnect (C1-SPCHX): This batch process creates CCB service point flat file(s).
- SP - MDM Characteristic Data Extract for DataConnect (X1-SPCHX): This batch process creates MDM service point flat file(s).
- Characteristic Type Data Extract for DataConnect (C1-CHTYX): This batch process creates characteristic type flat file(s). This batch control is defined as values for the Batch Process for Extract algorithm parameter on the Pre-Processing algorithm on the Characteristic Type Sync for DataConnect (C1-ExternalRepoPlusCharTypSync) business objects (see above).

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Setting Up and Executing Characteristic Data Extracts

Setting up master data extracts involves the following steps:

Copyright © 2026, Oracle and/or its affiliates.

1. Add audit algorithms on the Account, Characteristic Type, Person, Premise, Service Agreement, Service Point - CCB, and Service Point - MDM maintenance objects.
    a. Add the Characteristic Change Data Capture (C1MOCHARCDC) algorithm as an audit algorithm on the Account, Characteristic Type, Person, Premise, Service Agreement, Service Point - CCB, and Service Point - MDM maintenance objects.
    b. Add the Characteristic Type Change Data Capture (C1CHARTYPCDC) algorithm as an audit algorithm on the Characteristic Type maintenance object.
2. Add Sync Request BO options on the Account, Person, Characteristic Type, Premise, Service Agreement, Service Point - CCB, and Service Point - MDM maintenance objects
    a. Add the Characteristic Data Sync for DataConnect (C1-ExternalRepoPlusCharDatSync) business object as Sync Request BO options on the Account, Person, Premise, Service Agreement, Service Point - CCB, and Service Point - MDM maintenance objects.
    b. Add the Characteristic Type Sync for DataConnect (C1-ExternalRepoPlusCharTypSync) business object as a Sync Request BO option on the Characteristic Type maintenance object.
3. Execute initial load batch processes related to accounts, persons, premises, service agreements, and service points.
4. Once the initial load synchronization has been executed, changes to accounts, characteristic types, persons, premises, service agreements, and service points will trigger the creation of new synchronization requests and resulting extraction files.
5. Execute the sync request monitor process.
6. Execute the extract batch processes related to accounts, characteristic types, persons, premises, service agreements, and service points.

Billing Data Extracts

When sending billing related data to an external system, both historical and current data needs to be extracted.

Historical data can be extracted as part of an initial load process, and only needs to be provided during initial setup of the integration. Historical data includes data history for all bill segments for a specified historical period that are:

- Linked to completed bills.
- Not linked to bills but are linked to frozen FTs that are linked to completed bills.

Current data should be extracted on a regular ongoing (or incremental) basis. However, in addition to sending current data, any billing corrections should be extracted as well.

Extraction of billing data is performed through use of data synchronization, post bill completion algorithms, business objects and related batch processes.

- Billing Data Extracts: Business Objects and Algorithms
- Billing Data Extracts: Batch Controls
- Setting Up and Executing Billing Data Extracts
- Billing Data DataConnect Master Configurations

Billing Data Extracts: Business Objects and Algorithms

Copyright © 2026, Oracle and/or its affiliates.

Billing generates synchronization requests when bills complete through a Customer Class – Post Bill Completion algorithm. The Change Data Capture of Billing Data for DataConnect (C1-CAPBILLEX) algorithm type provided in the base package can be used to create a Customer Class – Post Bill Completion algorithm that creates a billing synchronization request record for each of the bill's bill segments if one does not exist in the initial state.

Billing related information extracts are supported by the following data synchronization business objects.

- Current DataConnect
  - Bill Sync for DataConnect (C1-ExternalRepoPlusBillSync)
  - Bill Segment Sync for DataConnect (C1-ExternalRepoPlusBillSegSync)
- **Legacy DataConnect**
  - Billing Sync for Legacy DataConnect (Tab-delimted) (C1-ExternalRepoPlusBillingSync)
  - Billing Sync for Legacy DataConnect (C1-ExternalRepoBillingSync)

These business objects should be defined as an algorithm parameter on the algorithm based on the Change Data Capture of Billing Data for DataConnect (C1-CAPBILLEX) algorithm type. These business object use the following Pre-Processing algorithms to take an initial data snapshot, and define the batch control used to extract data and export to a flat file. However, since billing data are always synched when it gets completed, no actual initial snapshot is taken. Instead, the parameters used for building the snapshots are taken and stored on the sync request record for later use when building the final snapshot.

Current DataConnect

Business Object	Pre-Processing Algorithm
Bill Sync for DataConnect (C1-ExternalRepoPlusBillSync)	Capture Bill Data Initial Snapshot (C1EXTCBILLI)
Bill Segment Sync for DataConnect (C1-ExternalRepoPlusBillSegSync)	Capture Bill Segment Data Initial Snapshot (C1EXTRCBSI)

Legacy DataConnect

Business Object	Pre-Processing Algorithm
Billing Sync for Legacy DataConnect (C1-ExternalRepoBillingSync)	Capture Billing Data Initial Snapshot for Legacy DataConnect (C1-EXTRCBLIP)
Billing Sync for Legacy DataConnect (Tab-delimited) (C1-ExternalRepoPlusBillingSync)	Billing Data Initial Snapshot for Legacy DataConnect (C1-EXTRCBLIP)

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for pending related synchronization requests, and, if none, transitions them to the Determine if Sync Needed state.

Enter algorithms on the Determine If Sync Needed states extract a final snapshot of the data if the billing related data is to be extracted and exported. These include:

- Algorithms that check if there is a need to export the billing data. It will discard the synchronization request if one of the following is true:
  - The synchronization request is for a cancelled bill segment whose original usage has not been previously sent out.

Copyright © 2026, Oracle and/or its affiliates.

– The synchronization request has been marked for non-synchronization (such as, the bill segments related SA Type is not configured to be extracted on the Billing Data Configuration for DataConnect master configuration record).

Business Object	Algorithm
Billing Sync for Legacy DataConnect (Tab—delimted) (C1-ExternalRepoPlusBillingSync)	Determine if the Billing Sync is Needed (C1-DSCANUSNS)
Billing Sync for Legacy DataConnect (C1-ExternalRepoBillingSync)	Determine if the Billing Sync is Needed (C1-DSCANUSNS)

- Algorithms that capture the billing data information and stores it on the synchronization request.

  Current DataConnect

Business Object	Algorithm
Bill Sync for DataConnect (C1-ExternalRepoPlusBillSync)	Capture Bill Data Final Snapshot (C1EXTCBILLF)
Bill Segment Sync for DataConnect (C1-ExternalRepoPlusBillSegSync)	Capture Bill Segment Data Final Snapshot (C1EXTRCBSF)

  Legacy DataConnect

Business Object	Algorithm
Billing Sync for Legacy DataConnect (C1-ExternalRepoBillingSync)	Capture Billing Data Final Snapshot for Legacy DataConnect (C1-CAPBILUSN)
Billing Sync for Legacy DataConnect (Tab-delimted) (C1-ExternalRepoPlusBillingSync)	Capture Billing Data Final Snapshot for Legacy DataConnect (C1-CAPBILUSN)

When using Legacy DataConnect, if a single extract file that combines both the billing and master data (customer and premise) information is preferred, the following algorithms can be used to capture the master data information of the bill. These algorithms should be plugged in on the Determine if Sync Needed state after the Capture Billing Data Final Snapshot for DataConnect algorithm.

- Capture Billing's SA Final Snapshot for DataConnect (C1-CEXTRCSA) - This is the algorithm that captures the service agreement, account and person information associated with the bill.
- Capture Billing's SP Final Snapshot for DataConnect (C1-CEXTRCSP) - This is the algorithm that captures the service point and premise information associated with the bill.
- Capture Billing's Installation Event Final Snapshot for DataConnect (X1-CEXTRCIE). This is the algorithm that captures install event information associated with the bill.

When using DataConnect v1, the Prepare Delimited Extract Data (Process All) Enter algorithm (C1-PRPEXDCUS/C1-PRPTDEXDT) on the Send Request state prepares and formats the data for extraction, and creates a general process record for the synchronization request (based on the extract batch control defined by the pre-processing algorithm).

Copyright © 2026, Oracle and/or its affiliates.

Billing Data Extracts: Batch Controls

The billing data extraction process uses a set of batch processes to create data synchronization requests and extract files.

Creating Initial Synchronization Requests

The following batch processes are used to create initial synchronization requests for bills:

- Current DataConnect
  - Bill Data Initial Load for DataConnect (C1-BLDIL)
  - Bill Segment Data Initial Load for DataConnect (C1-BSDIL)
- **Legacy DataConnect**
  - Billing Data Initial Load for Legacy DataConnect (C1-BLEIL)

These batch processes should be run to create initial load data synchronization requests based on the sync business objects described above.

Creating Extract Files

The following batch process is used to create extract files from synchronization requests for billing related information.

- Current DataConnect
  - Bill Data Extract for DataConnect (C1-BLDTX)
  - Bill Segment Data Extract for DataConnect (C1-BSDTX)
- **Legacy DataConnect**
  - Billing Data Extract for Legacy DataConnect (C1-BSYEX)

These batch controls are defined as values for the Batch Control for Extract algorithm parameters on the Pre-Processing algorithms on the sync business objects described above.

The following batch controls are used to extract bill segment lines and service quantities:

- Bill Seg Lines Extract - DataConnect (C1-BSLDX)
- Bill Segment SQ Extract - DataConnect (C1-BSSQX)

These should be run after bill segment extract has been run.

Use the **Batch Control** portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file) for a delimited flat file containing extracted data.

Setting Up and Executing Billing Data Extracts

Setting up billing data extracts involves the following steps:

1. Configure the Billing Data Configuration for the DataConnect master configuration record. This is used to configure the UOM, TOU, and SQI of the SA Type's service quantity and charges that will be extracted. Additionally, you can also configure the UOM, TOU and SQI at the rate level.
2. Add the Capture Bill and Bill Segment Extract for DataConnect v1+ (C1CAPBILBSEG) related post bill completion algorithm to the applicable customer classes.

Copyright © 2026, Oracle and/or its affiliates.

If using DataConnect v1, use the Capture Billing Data for DataConnect (Comma-delimited) (C1-CAPBILLEX) algorithm.

3. Execute initial load batch processes related to billing data.
4. Once the initial load synchronization has been executed, current data should be extracted on a regular ongoing (or incremental) basis. In addition to sending current data, any billing corrections will be extracted as well (data related to bill segments not linked to bills but are linked to frozen FTs that are linked to completed bills).
5. Execute the sync request monitor process.
6. Execute the extract batch processes related to billing.

Billing Data DataConnect Master Configurations

The master configuration business objects for billing data Configuration for DataConnect define general configuration details for the DataConnect functionality.

Billing Data Configuration for Legacy DataConnect

The Billing Data Configuration for Legacy DataConnect master configuration is used to configure the UOM, TOU, and SQI of the SA Type's service quantity and charges that will be extracted. Additionally, you can also configure the UOM, TOU and SQI at the rate level.

Billing Extract Master Configuration for DataConnect

The Billing Extract Master Configuration for DataConnect master configuration is used to define the configuration used to extract billing data. This includes the extraction types, including bill and bill segment sync request business objects, as well as CIS Divisions and SA Types for each, the number of historical periods for initial load, and the calculation line's category based on the rating engine used (billing extract assumes that only one rating engine is used by the system).

Navigate using**Admin**, then **General**, then **Master Configuration.** In the Master Configuration list, scroll to the appropriate master configuration and click the add icon if there is no record yet or the edit icon to modify an existing record.

For more information about specific fields in the master configuration, refer to the embedded help.

Consumption Data Extracts

When sending interval measurement usage data to an external system, both historical and current data needs to be extracted. Historical data can be extracted as part of an initial load process, and only needs to be provided during initial setup of the integration. Historical data should include data history for all active service points for a specified historical period. Current data should be extracted on a regular ongoing (or incremental) basis. However, in addition to sending current data, any historical corrections received by the system should be extracted as well.

Extract Requests

There are several types of consumption extract requests:

- **Initial Load**: Initial load extract requests are created and submitted manually via the Consumption Extract Request portal. Consumption extract requests are based on a

Copyright © 2026, Oracle and/or its affiliates.

specified Consumption Extract Type (see below) and extraction date range. An initial load request must be created and submitted for each consumption extract type defined in the system.

- **Incremental / Ongoing (Current Data)**: Incremental / ongoing extract requests can be manually created, but more often will be created via a batch process. The "Create Daily Consumption Extract Requests" batch control scans active consumption extract types and creates a request for each one that has Frequency of "Automated Daily." Ad-hoc incremental requests can be created and submitted manually if needed.
- **Historical Correction**: Historical correction extracts are created via batch process. Algorithms on the Finalized state of the initial measurement and measurement business objects determine if a finalized initial measurement or rederived values are historical corrections. These algorithms create records which are evaluated by a batch process which extracts the measurements for the related initial measurements.

Consumption Extract Type

Consumption Extract Types define the specific parameters used when processing a consumption extract request. Consumption Extract Types control the service point type, type of measurement, and how the measurements are grouped into TOU periods if applicable. Consumption extract types also define the algorithm and batch processes to use when extracting data for different types of requests (initial load, incremental, and historical).

There are two consumption extract type business objects provided with the base package:

- **Energy Management Extract (D2-ConsumptionExtractType):** This business object retrieves interval data and converts it to a specified Target UOM and Interval Size. This business object does not support TOU mapping.
- **Energy Management Extract with TOU Mapping (D2-ConsumptionExtractTypeTOU):** This business object retrieves interval data, converts it to a specified Target UOM and Interval Size, and maps it to a specific TOU map prior to extraction.

Historical Versus Current Data

The "Extract Through Date/Time" field on the Consumption Extract Type is used to differentiate between current data (the most recently extracted data) and historical corrections, and is set to the last date on which data was extracted for that extract type. For example, if data is extracted on June 1, 2015, the "Extract Through Date/Time" would be set to "June 1, 2015 12:00AM." If/ when data is extracted the next day, "Extract Through Date/Time" would be updated to "June 2, 2015 12:00AM."

When evaluating data for extract:

- Interval data is considered current if its measurement date time is after the "Extract Through Date/Time".
- Interval data is considered a historical correction if its measurement date time is on or before the "Extract Through Date/ Time".

Historical data changes to an initial measurement can be detected when it enters the Finalized state. If the initial measurement is determined to be for a historical period by comparing its end date/time against the "Extracted Through Date/Time" on the Consumption Extract Type, a general process record will be written for the initial measurement so that measurements for it can be extracted. In addition, re-derived values on final measurements can also trigger the creation of a general process record for related initial measurements.

The following algorithms are used in this process:

- The "Create General Process Record if IMD is Historical Correction" algorithm is used to determine if a finalized initial measurement data is a historical correction. If it is, the

Copyright © 2026, Oracle and/or its affiliates.

algorithm creates a general process record for the initial measurement. This algorithm is provided in the base package, but not specified on initial measurement business objects by default. This algorithm should be defined as an Enter algorithm on the Finalized state of the initial measurement business objects.

- The "Create General Process Record for Re-derived Values" algorithm creates general process records for initial measurements associated with re-derived values. Processing will proceed as if a historical correction came in through an initial measurement. This algorithm is provided in the base package, but not specified on measurement business object by default. This algorithm should be defined as an Enter algorithm on the Re-derive state of the final measurement business object.

- Consumption Extract Requests
- Consumption Extract Requests - Batch Controls
- Consumption Extract Requests - Setup Steps

Consumption Extract Requests

Initial load and ongoing consumption extracts are created via Consumption Extract Requests. While extracts of these types can be created via adhoc submission of a batch job, requests are the preferred method for these types of consumption extracts.

The consumption extract request business object lifecycle includes logic that maintains and updates the "Extraction Through Date/Time" field on Consumption Extract Types, which is used to determine if daily requests should be created by the "Create Daily Consumption Extract Requests" batch control, and detect historical corrections.

A single business object is provided in the base package for consumption extract requests:

- **Consumption Request (D2-IntervalDataExtRepository):** This business object contains the information and lifecycle responsible for submitting the extract job, monitoring the run until it's finished, and then updating the Consumption Extract Type's "Extract Through Date/Time" on the Consumption Extract Type. This business object is based on the Request (F1-REQ) maintenance object.

The consumption extract process uses a set of base package algorithms to extract and format the interval data for export. These algorithms are specified in the '"Algorithms" section on the Consumption Extract Type as appropriate. These algorithms can be configured to allow for extraction of data for frequently-read scalar measuring components as well as interval measuring components. Frequently-read scalar measuring components are defined as scalar measuring components whose Read Method is set to "Automatic Read." When extracting measurements for frequently-read scalar measuring components, scalar measurements are converted to interval measurements as part of the extraction process. This conversion uses the profile associated with the measuring component type. If no profile can be found, the interval data uses a flat line profile.

Initial Load / Incremental / Ongoing Requests: the following algorithms are used to extract and format interval data for initial load and incremental / ongoing requests:

- **Extract Initial Load/Ongoing Consumption for DataConnect (D2-IDEXTPRD):** This algorithm retrieves a service point's consumption for a given period and writes the results to a flat file.
- **Extract Tab-Delimited Initial Load/Ongoing Consumption (with UTC Offset)** (D2-IDEXTPRDU): This algorithm retrieves a service point's consumption for a given period and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.

Copyright © 2026, Oracle and/or its affiliates.

- **Extract Initial/Ongoing Consumption and Apply TOU Map for DataConnect (D2-IDEXTPTOU):** This algorithm retrieves a service point's consumption for a given period, applies a TOU Map to the consumption, and writes the results to a flat file
- **Extract Tab-Delimited Initial/Ongoing Consumption and Apply TOU Map (with UTC Offset)** (D2-IDEXTPTUU): This algorithm retrieves a service point's consumption for a given period, applies a TOU Map to the consumption, and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.

Historical Corrections: the following algorithms are used to extract and format interval data for historical correction requests.

- **Extract Historical Correction Consumption for DataConnect (D2-IDEXTIMD):** This algorithm retrieves historical correction consumption for a service point and writes the results to a flat file.
- **Extract Tab-Delimited Historical Correction Consumption (with UTC Offset)** (D2-IDEXTIMDU): This algorithm retrieves historical correction consumption for a service point and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.
- **Extract Historical Corrections and Apply TOU Map for DataConnect (D2-IDEXTITOU):** This algorithm retrieves historical correction consumption for a service point, applies a TOU map to the consumption, and writes the results to a flat file.
- **Extract Tab-Delimited Historical Corrections and Apply TOU Map (with UTC Offset)** (D2-IDEXTITUU): This algorithm retrieves historical correction consumption for a service point, applies a TOU map to the consumption, and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.

Consumption Extract Requests - Batch Controls

The consumption extraction process uses a set of base package batch controls to extract and format the interval data for export.

- **Create Daily Consumption Extract Requests (D2-CRERQ):** This batch process scans for active Consumption Extract Types, and for each one that has Frequency of Automated Daily creates a pending request. This process should be scheduled to run daily (or at another regular interval).

The following sample batch controls are provided to extract and format interval data. Unique batch controls of each of these is required for each consumption extract type. You should create custom versions of the above batch controls for each consumption extract type in your implementation. Extract type-specific versions of these batch controls should be specified in the "Batch Control" section on the Consumption Extract Type as appropriate.

- **Initial Load/Ongoing Consumption Extract (D2-IDEPD):** This batch process extracts interval data for a specified period. This batch control uses the "Initial Load/Ongoing Extract Algorithm" defined on the Consumption Extract Type.
- **Historical Corrections Consumption Extract (D2-IDEHC):** This batch process extracts interval data for historical corrections. This batch control uses the "Historical Corrections Extract Algorithm" defined on the Consumption Extract Type.

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Copyright © 2026, Oracle and/or its affiliates.

Consumption Extract Requests - Setup Steps

Example Setup Steps

Setting up consumption extracts involves the following steps:

1. Create Consumption Request Types: you should create a consumption request type for each unique type of request required by your implementation.
2. Create Consumption Extract Types: you should create a consumption extract type for each unique combination of output details (target UOM, target interval size, TOU map/template) and measurement selection criteria.
3. Add the "Create General Process Record if IMD is Historical Correction" historical correction algorithm as an Enter algorithm on the Finalized state of the initial measurement business objects.
4. Add the "Create General Process Record for Re-derived Values" historical correction algorithm as an Enter algorithm on the Finalized state on the measurement business object.
5. Create and submit initial load consumption extract requests for each consumption extract type you created earlier.
6. Set up batch processes for daily extract requests and historical corrections. Batch processing for consumption extracts should include the following:
    a. The **Create Daily Consumption Extract Requests** (D2-CRERQ) batch control should be configured to run on a regular (i.e. daily) schedule to create ongoing consumption extract requests.
    b. The **Request Monitor (Deferred)** (F1-SUBRQ) batch control should be used to monitor for pending requests and transition them to the "Submit Job" state.
    c. **Historical corrections consumption extract batch controls** (based on D2-IDEHC, one for each consumption extract type), should be configured to run on a regular basis to check for and create historical correction extracts.

Extract Flat File Formats

This section outlines the format of output flat files created by master data, billing data, and consumption data extracts.

Data Areas:

- C1-AcctMailingAddrDataSnapshot
- C1-BillDataSnapshot
- C1-BillSegmentDataSnapshot
- C1-CustomerDataSnapshot
- C1-ExternalRepoSABasedSnapshot
- D1-ExternalRepositorySPSnapsht
- D1-ExternalRepositorySPMtrSnap
- D1-ExternalRepositoryIESnapsht
- C1-ExternalRepoBillingSnapshot

Copyright © 2026, Oracle and/or its affiliates.

- D2-IntervalDataExtRepoSnapshot (used for interval consumption extract for Legacy DataConnect)
- D2-IntervalDataExtRepoTOUSnap (used for interval consumption extract mapped to TOU periods for Legacy DataConnect)
- D2-IntervalDataExtRepoSnpstUTC (used for interval consumption extract)
- D2-IntervalDataExtRepoTOUSnUTC (used for interval consumption extract mapped to TOU periods)

Specifics for how the flat files are created are defined on batch controls and algorithms used by the extract process.

- **File Name and Path:** Parameters on batch controls define the file name, path, and other details about the output file.
- **File Size and Contents:** Batch controls for initial load/ongoing consumption requests include a parameter to specify the number of service points to be included in each file.

C1-AcctMailingAddrDataSnapshot

C1-AcctMailingAddrDataSnapshot is used for mailing address data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Account ID	accountId	ACCOUNT_ID
Person Id	personId	PERSON_ID
Is Seasonal Address	isSeasonalAddress	
Seasonal Address Start Date/ Time	seasonStart	IS_SEASONAL_ADDRESS
Seasonal Address End Date/Time	seasonEnd	SEASONAL_ADDRESS_END_DTTM
Mailing Address 1	address1	MAIL_ADDRESS_1
Mailing Address 2	address2	MAIL_ADDRESS_2
Mailing Address 3	address3	MAIL_ADDRESS_3
Mailing Address 4	address4	MAIL_ADDRESS_4
Locality	city	LOCALITY
Postal Code	postal	MAIL_POSTAL_CODE
Admin Area 1	state	ADMIN_AREA_1
Admin Area 2	adminArea2	ADMIN_AREA_2
Admin Area 3	adminArea3	ADMIN_AREA_3
Country	country	COUNTRY
Updated at	personAddressUpdatedAt	PERSON_ADDRESS_UPDATED_AT

Top of Page

C1-BillDataSnapshot

C1-BillDataSnapshot is used for the bill-based data extracts, and includes the following information:

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Bill ID	billId	BILL_ID
Account ID	accountId	ACCOUNT_ID
Bill Date	billDate	BILL_DATE
Due Date	dueDate	DUE_DATE
Currency Code	currencyCode	CURRENCY_CD
Current Balance	currentBalance	CURRENT_BALANCE
Payoff Balance	payoffBalance	PAYOFF_BALANCE
Previous Period Balance	previousPeriodBalance	PREVIOUS_PERIOD_BALANCE
Total Payments	totalPayments	TOTAL_PAYMENTS
Total Adjustments	totalAdjustments	TOTAL_ADJUSTMENTS
Total Bill Corrections	totalBillCorrections	TOTAL_BILL_CORRECTIONS
Total Current Billing Charges	totalCurrentBillingCharges	TOTAL_CURRENT_BILLING_CHARGES
Total Balance Current Period	totalBalanceCurrentPeriod	TOTAL_BALANCE_CURRENT_PERIOD
Last Update Date/Time	lastUpdateDateTime	BILL_UPDATED_AT

Top of Page

C1-BillSegmentDataSnapshot

C1-BillSegmentDataSnapshot is used for the bill segment-based data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Bill ID	billId	BILL_ID
Bill Segment ID	billSegmentId	BILL_SEGMENT_ID
Service Agreement	serviceAgreement	SERVICE_AGREEMENT_ID
Start Date	startDate	BILL_SEGMENT_START_DATE
End Date	endDate	BILL_SEGMENT_END_DATE
Usage Start Date/Time	usageStartDateTime	USAGE_PERIOD_START_DATE
Usage End Date/Time	usageEndDateTime	USAGE_PERIOD_END_DATE
Estimated	isEstimate	ESTIMATED
Closing Billing Statement	isClosingBillSegment	CLOSING
Bill Segment Status	billSegmentStatus	STATUS
Bill Cycle	billCycle	BILL_CYCLE_CODE
Bill Cycle Description	billCycleDescr	BILL_CYCLE_DESCRIPTION
Premide ID	premiseId	PREMISE_ID
Current Amount	currentAmount	CURRENT_AMOUNT
Outstanding Amount	outstandingAmount	OUTSTANDING_AMOUNT
Master Bill Segment ID	masterBillSegmentId	MASTER_BILL_SEGMENT_ID
Last Update Date/Time	lastUpdateDateTime	BILL_SEGMENT_UPDATED_AT

Copyright © 2026, Oracle and/or its affiliates.

Top of Page

C1-CustomerDataSnapshot

C1-CustomerDataSnapshot is used for customer data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Account ID	accountId	ACCOUNT_ID
Account Name	accountName	ACCOUNT_NAME
Account Setup Date/Time	accountSetUpDateTime	SETUP_DATETIME
Customer Class	customerClass	CUSTOMER_CLASS
Bill Cycle	billCycle	BILL_CYCLE_CODE
Bill Cycle Description	billCycleDescription	BILL_CYCLE_DESCRIPTION
Account Updated at	accountUpdatedAt	ACCOUNT_UPDATED_AT
Person ID	personId	PERSON_ID
Person Type	personOrBusiness	PERSON_TYPE
First NamE	firstName	FIRST_NAME
Last Name	lastName	LAST_NAME
Business Name	businessName	BUSINESS_NAME
Language Preference	language	LANGUAGE_PREFERENCE
Is Main Customer	isMainCustomer	IS_MAIN_CUSTOMER
Bill Route Typ	billRouteType	BILL_ROUTE_TYPE
Person Updated at	personUpdatedAt	PERSON_UPDATED_AT
Phone Type	phoneType	PHONE_TYPE_
Phone Number	phoneNumber	PHONE_NUMBER_
Phone Extension	phoneExtension	PHONE_EXTENSION_
Email	email	EMAIL_ADDRESS_
Email Type	emailType	EMAIL_TYPE_
Service Agreement ID	saId	SERVICE_AGREEMENT_ID
Service Agreement Name	saName	SERVICE_AGREEMENT_NAME
Start Date/Time	startDateTime	SA_START_DATETIME
End Date/Time	endDateTime	SA_END_DATETIME
Service Type	serviceType	SERVICE_TYPE_CODE
Service Type Description	serviceTypeDescription	SERVICE_TYPE_DESC
SA Status	saStatus	STATUS_CODE
SA Status Description	saStatusDescription	STATUS_DESCRIPTION
SA Type	saType	SERVICE_AGREEMENT_TYPE_CODE
SA Type Description	saTypeDescription	SA_TYPE_DESCRIPTION
Special Role Flag	specialRole	SPECIAL_ROLE_FLAG_CODE
Special Role Flag Description	specialRoleDescription	SPECIAL_ROLE_FLAG_DESCRIPTION
CIS Division	cisDivision	CIS_DIVISION

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
SIC	sic	BUSINESS_CLASSIFICATION_CODE
SA Updated at	saUpdatedAt	SERVICE_AGREEMENT_UPDATED_AT
SA Rate Code	saRateCode	SA_RATE_CODE
SA Rate Effective Date/Time	saRateEffectiveDateTime	EFFECTIVE_DATETIME
SA Rate Updated at	saRateUpdatedAt	SA_RATE_UPDATED_AT

Top of Page

C1-ExternalRepoSABasedSnapshot

C1-ExternalRepoSABasedSnapshot is used for the service agreement-based master data extracts, and includes the following information:

For DataConnect v1+ please refer to data area C1-CustomerDataSnapshot in the application for the layout/schema.

Field Name	Schema Element	Metadata Field
Service Agreement Id	saId	SA_ID
Account Id	accountId	ACCT_ID
Person Id	personId	PER_ID
Bill Cycle	billCycle	BILL_CYC_CD
Bill Cycle Description	billCycleDescription	DESCR
Collection Class	collectionClass	COLL_CL_CD
Collection Class Description	collectionClassDescription	DESCR
Account Information	accountInformation	ACCT_INFO
Customer Class	customerClass	CUST_CL_CD
Customer Class Description	customerClassDescription	DESCR
Landlord Indicator	landlordIndicator	C1_LANDLORD_IND
Mailing Address	mailingAddress1	ADDRESS1
Mailing Address 2	mailingAddress2	ADDRESS2
Mailing Address 3	mailingAddress3	ADDRESS3
Mailing City	mailingCity	CITY
Mailing State	mailingState	STATE
Mailing Postal	mailingPostal	POSTAL
First Name	firstName	ENTITY_NAME
Last Name	lastName	ENTITY_NAME
Phone Number 1	phone1	PHONE
Phone Number 2	phone2	PHONE
Email Address	email	EMAILID
Special Role	specialRole	SPECIAL_ROLE_FLG
Special Role Description	specialRoleDescription	DESCR
Service Type	serviceType	SVC_TYPE_CD

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Service Type Description	serviceTypeDescription	DESCR
CIS Division	cisDivision	CIS_DIVISION
CIS Division Description	cisDivisionDescription	DESCR
SA Type	saType	SA_TYPE_CD
SA Type Description	saTypeDescription	DESCR
Revenue Class	revenueClass	REV_CL_CD
Revenue Class Description	revenueClassDescription	DESCR
SIC	sic	SIC_CD
SIC Code Description	sicDescription	DESCR
Deposit Class	depositClass	DEP_CL_CD
Deposit Class Description	depositClassDescription	DESCR
Campaign	campaign	CAMPAIGN_CD
Campaign Description	campaignDescription	DESCR
Debt Class	debtClass	DEBT_CL_CD
Debt Class Description	debtClassDescription	DESCR
SA Start Date	startDate	START_DT
SA End Date	endDate	END_DT
SA Status	saStatus	SA_STATUS_FLG
SA Status Description	saStatusDescription	DESCR
Rate Schedule	rateSchedule	RS_CD
Rate Schedule Description	rateScheduleDescription	DESCR
SA's Service Point(s) Id	spId	SP_ID
SA's Service Point(s) Start Date/ Time	startDateTime	START_DTTM
SA's Service Point(s) End Date/ Time	stopDateTime	STOP_DTTM

Top of Page

D1-ExternalRepositorySPSnapsht

D1-ExternalRepositorySPSnapsht is used for service point-based master data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSpId	CIS_EXT_SP_ID
Address	address1	DESCR254
City	city	CITY
State	state	STATE
Postal	postal	POSTAL
Service Point Type Code	spType	D1_SP_TYPE_CD
Service Point Type Description	spTypeDescription	DESCR100

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Service Type Code	serviceType	D1_SVC_TYPE_CD
Service Type Description	serviceTypeDescription	DESCR100
Market Code	market	MKT_CD
Market Description	marketDescription	DESCR100
Service Point Status Code	boStatus	BO_STATUS_CD
Service Point Status Description	boStatusDescription	DESCR100
Disconnect Location Flag	disconnectLocation	DISCONN_LOC_FLG
Disconnect Location Description	disconnectLocationDescription	DESCR100
Measurement Cycle Code	measurementCycle	MSRMT_CYC_CD
Measurement Cycle Description	measurementCycleDescription	DESCR100
Measurement Cycle Route Code	measurementCycleRoute	MSRMT_CYC_RTE_CD
Measurement Cycle Route Description	measurementCycleRouteDescription	DESCR100
Geocode Latitude	geocodeLatitude	D1_GEO_LAT
Geocode Longitude	geocodeLongitude	D1_GEO_LONG
Facility Level 1	facilityLevel1	FACILITY_LEVEL1
Facility Level 2	facilityLevel2	FACILITY_LEVEL2
Facility Level 13	facilityLevel3	FACILITY_LEVEL3

Top of Page

D1–ExternalRepositorySPMtrSnap

D1-ExternalRepositorySPMtrSnap is used for service points, device configurations, and devices master data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Premise ID	premiseId	D1_PREMISE_ID
Premise Name	premiseName	PREMISE_NAME_LBL
External Premise ID	externalPremiseId	D1_SP_EXT_PREMISE_ID
Parent Premise ID	parentPremiseId	D1_PREMISE_ID
Premise Type	premiseType	D1_PREMISE_TYPE
Premise Type Description	premiseTypeDescription	PREMISE_TYPE_DESCR_LBL
Address 1	address1	SP_ADDRESS1_LBL
Address 2	address2	SP_ADDRESS2_LBL
Address 3	address3	SP_ADDRESS3_LBL
Address 4	address4	SP_ADDRESS4_LBL
City	city	CITY
County	county	DESCR50
Postal	postal	POSTAL
State	state	STATE
Country	country	COUNTRY

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Admin Area 3	adminArea3	D1_ADMIN_AREA_3
Division	division	DIVISION_CD
Premise Time Zone	premiseTimeZone	PREMISE_TIME_ZONE_CD
Geocode Latitude	geocodeLatitude	D1_GEO_LAT
Geocode Longitude	geocodeLongitude	D1_GEO_LONG
Landlord Agreement	landlordAgreementId	D1_LL_ID
Premise Updated At	premiseUpdatedAt	D1_PREMISE_UPDATED_AT
Service Point ID	spId	D1_SP_ID
Service Point Name	spName	SP_NAME_LBL
SP Type Description	spTypeDescription	SP_TYPE_DESCR_LBL
Service Point Type Category	spTypeCategory	SP_CATEGORY_FLG
SP Type	spType	D1_SP_TYPE_CD
Most Recent SA ID	mostRecentSAId	US_ID
Service Agreement Start Date/ Time	SAStartDateTime	START_DTTM
Service Agreement Stop Date/ Time	SAStopDateTime	D2_STOP_DTTM
Measurement Cycle	measurementCycle	MSRMT_CYC_CD
Measurement Cycle Description	measurementCycleDescription	MSRMT_CYCLE_DESCR_LBL
Measurement Cycle Route	measurementCycleRoute	MSRMT_CYC_RTE_CD
Measurement Cycle Route Description	measurementCycleRouteDescription	MSRMT_CYCLE_RT_DESCR_LBL
Facility Level 1	facilityLevel1	FACILITY_LEVEL1
Facility Level 2	facilityLevel2	FACILITY_LEVEL2
Facility Level 3	facilityLevel3	FACILITY_LEVEL3
Life Support / Senstive Load	lifeSupportSensitiveLoad	D1_LS_SL_FLG
UTC Offset	utcOffset	UTC_OFFSET
Business Object Status	boStatus	SP_BO_STATUS_LBL
Business Object Status Description	boStatusDescription	SP_BO_STATUS_DESCR_LBL
Service Source Status	spSourceStatus	SP_SRC_STAT_FLG
Service Source Status Description	spSourceStatusDescription	SP_SOURCE_STATUS_DESCR_LBL
Disconnect Location	disconnectLocation	DISCONN_LOC_FLG
Disconnect Location Description	disconnectLocationDescription	DISCON_LOC_DESCR_LBL
Service Point Time Zone	spTimeZone	SP_TIME_ZONE_CD
Service Type	erviceType	D1_SVC_TYPE_CD
Service Type Description	serviceTypeDescription	SERVICE_TYPE_DESCR_LBL
Disconnect for NonPay	disconnectForNonPay	D1_DISCONNECT_NON_PAY
Service Point Type Category Description	spTypeCategoryDescription	SP_TYPE_CAT_DESCR_LBL

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Remote Disconnect	remoteDisconnect	D1_SGG_DISC_CFG
Install Event ID	installEventId	INSTALL_EVT_ID
Install Event External ID	deviceInstallationExternalId	D1_DVC_INSTALL_EXT_ID
Meter ID	meterId	D1_DEVICE_ID
Meter Name	meterName	D1_DEVICE_NAME
Install Date/Time	installDateTime	D1_INSTALL_DTTM
Removal Date/Time	removalDateTime	D1_REMOVAL_DTTM
Installation Event BO Status	installationBoStatus	IE_BO_STATUS_LBL
Installation Event BO Status Description	installationBoStatusDescription	IE_BO_STATUS_DESCR_LBL
Arming Status	armingStatus	ARMING_REQ_FLG
Device On/Off Status	deviceOnOffStatus	ONOFF_HIST_FLG
Installation Constant	installationConstant	INSTALLATION_CONST
Meter External ID	meterExternalId	D1_DVC_EXTERNAL_ID
Device Type	deviceType	DEVICE_TYPE_CD
Device Type Description	deviceTypeDescription	DEVICE_TYPE_DESCR_LBL
Manufacturer	manufacturer	MANUFACTURER_CD
Manufacturer Description	manufacturerDescription	MANUFACTURER_DESCR_LBL
Model	model	D1_MODEL_CD
Model Description	modelDescription	MODEL_DESCR_LBL
Device Manufacture Date	deviceManufactureDate	D1_DEVICE_MANUFACTURE_DATE
Dial Count	dialCount	D1_DIAL_COUNT
Sub Meter	subMeter	SUB_METER_SW
Net Meter	netMeter	NET_METER_SW
Device Business Object Status	deviceBoStatus	DEVICE_BO_STATUS_LBL
Device BO Status Description	deviceBoStatusDescription	DVC_BO_STATUS_DESCR_LBL
Device Service Type	deviceServiceType	DEVICE_SERVICE_TYPE_LBL
Device Service Type Description	deviceServiceTypeDescription	DEVICE_SERVICE_TYPE_DESCR_LBL
Head End System	headEndSystem	D1_SPR_CD
Head End System Description	eadEndSystemDescription	HEAD_END_SYSTEM_DESCR_LBL
Head End Registration Status	headEndRegistrationStatus	HEAD_END_REGISTR_STATUS_FLG
Device Classification	deviceClassification	DVC_CLASS_FLG
Measuring Component ID	measuringComponentId	MEASR_COMP_ID
External ID	externalId	D1_MC_EXTERNAL_ID
Channel ID	channelId	CHANNEL_ID
Number of Digits Left	numberOfDigitsLeft	D1_NBR_OF_DGTS_LFT
Number of Digits Right	numberOfDigitsRight	D1_NBR_OF_DGTS_RGT
Channel Multiplier	channelMultiplier	CHANNEL_MULTIPLIER

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Measuring Component Type	measuringComponentType	MEASR_COMP_TYPE_CD
Measuring Component Type Description	measuringComponentTypeDescription	MC_TYPE_DESCR_LBL
Measuring Component Class	measuringComponentClass	MC_CLASS_FLG
Seconds per Interval (SPI)	secondsPerInterval	SEC_PER_INTRVL
Interval/Scalar	intervalScalar	INTERVAL_SCALAR_FLG
UOM	uom	UNIT_OF_MEASR_CD
TOU	tou	TIME_OF_USE_CD
SQI	sqi	SVC_QTY_IDNTFR_CD
Allow Negative Consumption	allowNegativeConsumption	ALLOW_NEG_CONSUMPTION_FLG
CIS Service Point ID	cisSPId	CSI_EXT_SP_ID

Top of Page

D1-ExternalRepositoryIESnapsht

D1-ExternalRepositoryIESnapsht used for install event-based master data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSpId	CIS_EXT_SP_ID
Meter ID	meterId	D1_DEVICE_ID
Device Type Code	deviceType	DEVICE_TYPE_CD
Device Type Description	deviceTypeDescription	DESCR100
Device Status Code	deviceBoStatus	BO_STATUS_CD
Device Status Description	deviceBoStatusDescription	DESCR100
Manufacturer Code	manufacturer	MANUFACTURER_CD
Manufacturer Description	manufacturerDescription	DESCR100
Model Code	model	D1_MODEL_CD
Model Description	modelDescription	DESCR100
Head-End System Code	headEndSystem	D1_SPR_CD
Head-End System Description	headEndSystemDescription	DESCR100
Badge Number	badgeNumber	D1_BADGE_NBR
Serial Number	serialNumber	D1_SERIAL_NBR
CIS Meter ID	cisMeterId	CIS_EXT_DEVICE_ID
Install Date/Time	installDateTime	D1_INSTALL_DTTM
Install Date/Time with UTC Offset	installDttmUtcOffset	D1_INSTALL_DTTM_UTC_OFFSET
Removal Date/Time	removalDateTime	D1_REMOVAL_DTTM
Removal Date/Time with UTC Offset	removalDttmUtdOffset	D1_REMOVAL_DTTM_UTC_OFFSET

Copyright © 2026, Oracle and/or its affiliates.

Top of Page

C1-ExternalRepoBillingSnapshot

C1-ExternalRepoBillingSnapshot is used for the billing-based extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Billable Service Quantity	billSQ	BILL_SQ
Bill Segment End Date	endDate	END_DT
Consumption Days	duration	CONSUMPTION_DAYS
Bill Segment Estimate Indicator	estimateIndicator	C1_BS_ESTIMATE_IND
Calculated Amount	calculatedAmount	CALC_AMT
Unit of Measure	uom	UOM_CD
Time of Use	tou	TOU_CD
SQI	sqi	SQI_CD
Service Agreement ID	saId	SA_ID
Service Point ID	spId	SP_ID

Top of Page

D2-IntervalDataExtRepoSnapshot

D2-IntervalDataExtRepoSnapshot is used for interval-based consumption data extracts for Legacy DataConnect, and includes the following information:

Field Name	Schema Element	Metadata Field
CIS Service Agreement ID	cisSAId	CIS_EXT_SA_ID
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSPId	CIS_EXT_SP_ID
Unit of Measure	uom	D1_UOM_CD
Service Quantity Identifier	sqi	SVC_QTY_IDNTFR_CD
Seconds Per Interval	secondsPerInterval	SECONDS_PER_INTERVAL
Measurement Value	measurementValue	MSRMT_VAL
Measurement Date	measurementDate	MSRMT_DATE
Measurement Time	measurementTime	MSRMT_TIME
Estimate Indicator	estimateIndicator	ESTIMATION_IND_FLG
Estimate Indicator Description	estimateIndicatorDescription	DESCR100
UTC Offset	utcOffset	UTC_OFFSET

Top of Page

D2-IntervalDataExtRepoTOUSnap

D2-IntervalDataExtRepoTOUSnap is used for interval consumption data extracts mapped to TOU periods for Legacy DataConnect, and includes the following information:

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
CIS Service Agreement ID	cisSAId	CIS_EXT_SA_ID
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSPId	CIS_EXT_SP_ID
Unit of Measure	uom	D1_UOM_CD
Time of Use Period	tou	D1_TOU_CD
Service Quantity Identifier	sqi	SVC_QTY_IDNTFR_CD
Seconds Per Interval	secondsPerInterval	SECONDS_PER_INTERVAL
Measurement Value	measurementValue	MSRMT_VAL
Start Date	startDate	START_DT
Start Time	startTime	START_TM
Start Time UTC Offset	startUtcOffset	UTC_OFFSET
End Date	endDate	END_DT
End Time	endTime	D2_END_TIME
End UTC Offset	endUtcOffset	UTC_OFFSET

Top of Page

D2-IntervalDataExtRepoSnpstUTC

D2-IntervalDataExtRepoSnpstUTC is used for interval-based consumption data extracts, and includes the following information:

Field Name	Schema Element	Metadata Field
Service Point ID	spId	D1_¿SP_¿ID
Measurement Value	measurementValue	MSRMT_¿VAL
Measurement Date	measurementDate	MSRMT_¿DATE
Measurement Time	measurementTime	MSRMT_¿TIME
Unit of Measure	uom	D1_¿UOM_¿CD
Estimate Indicator	estimateIndicator	ESTIMATION_¿IND_¿FLG
UTC Offset	utcOffset	UTC_¿OFFSET
Register ID	mcId	REGISTER_ID
CIS Service Point ID	cisSPId	CIS_¿EXT_¿SP_¿ID
Device ID	deviceId	D1_DEVICE_ID
Service Type	serviceType	D1_SVC_TYPE_CD
Service Quantity Identifier	sqi	SVC_¿QTY_¿IDNTFR_¿CD
Time of Use	tou	D1_TOU_CD

Top of Page

D2-IntervalDataExtRepoTOUSnUTC

D2-IntervalDataExtRepoTOUSnUTC is used for interval consumption data extracts mapped to TOU periods, and includes the following information:

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Schema Element	Metadata Field
Service Point ID	spId	D1_¿SP_¿ID
Unit of Measure	uom	UNIT
Seconds Per Interval	secondsPerInterval	SECONDS_¿PER_¿INTERVAL
Measurement Value	measurementValue	MSRMT_¿VAL
Start Date	startDate	START_¿DT
Start Time	startTime	START_¿TM
Start Time UTC Offset	startUtcOffset	UTC_¿OFFSET
End Date	endDate	END_¿DT
End Time	endTime	D2_¿END_¿TIME
End UTC Offset	endUtcOffset	UTC_¿OFFSET
Register ID	mcId	REGISTER_ID
CIS Service Point ID	cisSPId	CIS_¿EXT_¿SP_¿ID
Device ID	deviceId	D1_DEVICE_ID
Service Type	serviceType	D1_SVC_TYPE_CD
Service Quantity Identifier	sqi	SVC_¿QTY_¿IDNTFR_¿CD
Time of Use	tou	D1_TOU_CD

Top of Page

Example Extract File

The following example illustrates comma-separated interval data extracts based on the Consumption Extract Snapshot () data area:

```
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,07.00.00,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,08.00.0,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366.2015-01-01,09.00.0,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,10.00.00,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,11.00.00,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,12.00.00,No
19502793-60E-KMUS,714532246966,19502793-60E-
KMSP,KWH, ,3,600,1.366,2015-01-01,13.00.00,No
```

File Naming Formats

This topic provides details concerning how DataConnect extract files are named.

Master Data and Billing Data Extracts

Extract files for master data and billing data extracts (except extracts for Bill Segment Lines and Service Quantities, see below) use the following naming format:

Copyright © 2026, Oracle and/or its affiliates.

<provided_file_name><data_source_indicator><batch_number><batch_thread_number><sysdate date/time>.dat

where:

- <provided_file_name> is the file name as specified for the Flat File Name parameter on the batch control.
- <data_source_indicator>is the Data Source Indicator value, which comes from the Environment ID field on the Installation Options table (CI_INSTALLATION.ENV_ID). This field is 6 digits, with left padded zeros as appropriate.
- <batch_number> is the batch number of the batch process that created the file. This field is 10 digits, with left padded zeros as appropriate.
- <batch_thread_number> is the thread number of the process that created the file. This field is 3 digits, with left padded zeros as appropriate.
- <sysdate date/time> is the date and time at which the file was created, if the Include Run Date/Time on Flat File Name parameter on the batch control was set to "Y", displayed as YYYYMMDDHHMMSS.

Example: If run with the following parameters and values:

- D_ACCT_EXT is the value in the Flat File Name parameter
- 000004 is the Data Source Indicator value
- 0000000003 is the batch number
- 001 is the thread number
- 20141001123020 is the system date/time (2014-10-01.12.30.20)

the extract file name would be as follows:

```
D_ACCT_EXT000004000000000300120141001123020.DAT
```

This file naming format is used for extracts created using the following batch controls:

- Account Characteristic Data Extract for DataConnect (C1-ACCHX)
- Bill Data Extract for DataConnect (C1-BLDTX)
- Bill Segment Data Extract for DataConnect (C1-BSDTX)
- Characteristic Type Data Extract for DataConnect (C1-CHTYX)
- Customer Data Extract for DataConnect (C1-CUDTX)
- Mailing Address Extract for DataConnect (C1-MADTX)
- Person Characteristic Data Extract for DataConnect (C1-PECHX)
- Premise Characteristic Data Extract for DataConnect (C1-PRCHX)
- SA/SP Data Extract for DataConnect (C1-SSDTX)
- SA Characteristic Data Extract for DataConnect (C1-SACHX)
- SP and Meter Data Extract for DataConnect (C1-SPMSY)
- SP - CCB Characteristic Data Extract for DataConnect (C1-SPCHX)
- SP - MDM Characteristic Data Extract for DataConnect (X1-SPCHX)
- Billing Data Extract for Legacy DataConnect (C1-BSYEX)
- Meter History Extract for Legacy DataConnect (C1-SMSYX)

Copyright © 2026, Oracle and/or its affiliates.

- SA-Based Extract for Legacy DataConnect (C1-SASYX)
- SP-Based Extract for Legacy DataConnect (C1-SPSYX)

Bill Segment Line and Service Quantity Extracts

Extract files for for bill segment lines and service quantities use a naming format as specified on the File Name parameter. This can include the following system variables

- {BC} or {batchCode}: the batch code for the batch control.
- {BN} or {batchNumber}: the batch number of the batch process that created the file. This field is 4 digits, with left padded zeros as appropriate.
- {RN} or {rerunNumber}: the rerun number of the batch process that created the file if rerun. This field is 2 digits, with left padded zeros as appropriate.
- {TN} or {threadNumber}: the thread number of the process that created the file. This field is3 digits, with left padded zeros as appropriate. For multi-threaded runs, {threadNumber} will be inserted if not supplied.
- {RDT} or {runDateTime}: the date and time at which the file was created, displayed as yyyyMMddHHmmss.
- {RD} or {runDate}: the date on which the file was created, displayed as yyyyMMdd.
- {RT} or {runTime}: the time at which the file was created, displayed as HHmmss.

Example: Specifting 'EXTRACT_FILE_{BN}_{RN}_{TN}.zip' as the File Name will create the following files for batch number 10, rerun number 0, with 3 threads:

```
EXTRACT_FILE_0010_00_001.zip


EXTRACT_FILE_0010_00_002.zip


EXTRACT_FILE_0010_00_003.zip
```

This file naming format is used for extracts created using the following batch controls:

- Bill Segment Lines Extract for DataConnect (C1-BSLDX)
- Bill Segment SQ Extract for DataConnect Extract (C1-BSSQX)

Business Flags

It is possible that information detected in one product may be useful or even critical to share with another product. The framework provides functionality for receiving information from an external system that acts as a type of flag or alert that may need investigation. This allows any system to store detected business flags in a common way and share that information with one or more other systems. See Business Flags for more information.

This section provides further information about business flags and their use in the system.

- Understanding Business Flags
- Business Flag Available Actions
- Business Flag Can Prevent Bills from Freezing
- Field Activities Can Determine Business Flag Confidence

Copyright © 2026, Oracle and/or its affiliates.

- Configuring Business Flags
- Business Flags Can Require Analysis
- Configuring Business Flag Integration
- Meter Operations Insights Business Flags
- A Field Activity Can Determine Business Flag Confidence
- Business Flags Can Create Alerts

Understanding Business Flags

Refer to About Business Flags in the *Oracle Utilities Framework Administration User Guide* for detailed information about the components of business flags.

Customer to Meter allows for business flags that typically represent situations, exist at a customer's service point, and are shared across external applications; such as, Oracle DataRaker, Oracle Utilities Mobile Workforce Management / Oracle Field Service Cloud. and Oracle Utilities Operational Device Management. The situations that a business flag can represent range from critical knowledge that requires manual analysis for resolution to purely informational notifications.

Sharing information allows users of each system to quickly understand the status of a service point (or device, or network location in some cases) as it relates to situations or insights that would benefit users of the system. For example, the advanced analytics system (Oracle DataRaker) may identify potential theft situations that should be investigated prior to releasing the next bills from CCB with usage transactions from MDM for the related service point's consumption.

The subsequent sections describe business flag functionality that is specific to the service point business flag, as implemented by this product.

When creating a new MDM device type there are the following options:

Name	Details	Business Object
Service Point Business Flag Type	Provides configuration for business flags associated to a service point. This type offers an ability to hold a business flag for manual analysis.	D1-SPBusinessFlagType
Service Point Monitor Business Flag Type	Provides configuration for business flags that should initiate service issue monitors and convey results of a field activity to the intended subscribers.	D1-SPMonitorBusinessFlagType

Important Business Flag Business Object Options

Below are a list of business object options that are defined on the business flag business object associated to a given business flag type:

- **Valid Action**: identifies a BPA script that can be used to guide the user though an action that can be taken on a business flag. Note: this option should be paired with a business object lifecycle option of "Execution of Valid Actions Permitted" with a value of "true" for any lifecycle status that should support the execution of valid actions.

Copyright © 2026, Oracle and/or its affiliates.

More detail about these options can be found by visiting a measuring component business object and inspecting the business object options.

Business Flag Available Actions

Allowed actions are implemented as BPA scripts that will guide the user through a particular method of investigating or resolving the business flag. The following actions are supported:

- **Create Field Activity:** This will initiate a field activity associated to the business flag's service point. The type of field activity will be selected by the user.
- **Create Case:** This will initiate a case associated to the business flag's service point. The type of case will be selected by the user.
- **Create Notification:** This will initiate communication with the customer for the business flag's service point. The type of communication will be determined by the notification type selected by the user.
- **Create Service Issue Monitor:** This will initiate a Service Issue Monitor (SIM) process associated to the business flag's service point. The SIM type is identified on the Business Flag Type configuration. If there are already existing SIM records of this type for the service point, the most recent SIM process will be linked to the business flag.
- Business Flag Adding Available Actions
- Making Allowed Actions Available

Business Flag Adding Available Actions

To extend the actions that are available to the user for investigating a business flag the following steps can be taken:

1. A new BPA script should be developed that provides the functionality for your particular business use case.
2. That BPA script will be added to the transactional business object for the business flag using the option type "Valid Action".
3. Edit the business flag type to include the newly created action in the "Allowed Actions" section of the business flag type.

Making Allowed Actions Available

Allowed actions are only accessible on those business object statuses that have been configured to support them. To permit access to the available actions on a given status one must add a business object status option for the type "Execution of Valid Actions Permitted" with a value of "Y".

Business Flag Can Prevent Bills from Freezing

Some business flags, particularly those that require investigation, should place a hold on the billing process to prevent potentially incorrect bills from being issued to the customer. When certain types of business flags are present on a service point it indicates that there may be a problem with the reported usage that is underlying the bill and as such it may not be desirable to allow the billing process to complete until the investigation into that usage has been completed. For example, if the business flag indicated potential theft or abnormally high usage

Copyright © 2026, Oracle and/or its affiliates.

it may be preferable to delay issuing the bill until an investigation can be performed and the issue rejected or resolved.

This functionality can be enabled through configuration of the Prevent Bill Segment Freeze Due to Business Flag algorithm.

See algorithm type Prevent Bill Segment Freeze Due to Business Flag (C1-PRVBSFBF) for more detail.

In MDM, Usage Transactions can also be prevented by configuring the *Valid Business Processes for Hold* with *Usage Processing* on the Service Point Business Flag Type.

Field Activities Can Determine Business Flag Confidence

When a business flag results in a field activity being issued and a field team is sent to investigate the service point the result of that field activity investigation will determine the confidence of the business flag. This is done through the field remarks left by the field team. The field activity remarks can either be defined as confirming the business flag or rejecting it. By selecting the appropriate field activity remark the field team will conclude the investigation into the business flag and the final confidence will be updated as appropriate.

For those field activity remarks that should either confirm or reject the business flag the Business Flag Confidence Update algorithm type should be provided for the field activity remark - activation plug-in spot.

Refer to the **Assign Business Flag Final Confidence** (D1-ABFFC) algorithm type for additional details.

Configuring Business Flags

For information on configuring business flags, see Setting Up Business Flag Configuration in the Oracle Utilities Application Framework *Administrative User Guide*.

Business Flags Can Require Analysis

For more serious situations a business flag can require manual analysis from a user. These types of business flags will generate a To Do when they are received from the external application. To resolve the To Do a user will investigate the service point based on the type of business flag present. Generally this investigation will result in the business flag being either confirmed or rejected. To reach the conclusion of whether the business flag should be confirmed or denied the user can be provided with a set of available actions that can be performed on the business flag.

See C1-BFANL for more information.

Configuring Business Flag Integration

Oracle Utilities Meter Solution products include the following components used business flag integrations:

Copyright © 2026, Oracle and/or its affiliates.

Master Configuration

The following master configurations are used to configure the data synchronization process between MDM and external analytics applications:

Master Configuration	Description
Oracle Utilities DataRaker Integration Master Configuration	This master configuration is used to enable deep linking into Oracle Utilities Analytics Insights. Multiple URL patterns are supported with an ability to provide substitution variables to satisfy the URL parameters (e.g. "&searchTech="). This also provides a mapping between Oracle Utilities Meter Data Management service types and point types defined in external analytics applications.

Oracle Utilities Data Intelligence

Receiving business flag insights from external analytics applications such as Oracle Utilities Data Intelligence requires set up of the following:

- Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup
- Measurement Identification - Electric Configuration (D1-MsrmtIdentifyElectricConfig) master configuration
- Measurement Identification - Gas Configuration (D1-MsrmtIdentifyGasConfig) master configuration
- Measurement Identification - Water Configuration (D1-MsrmtIdentifyWaterConfig) master configuration

See Defining Standard Measurement Identifiers for more information.

Service Provider Processing Methods

The following processing methods should be associated to the service provider configured for external analytics applications:

Processing Role	Description
Business Flag Confidence Change	This role will identify the appropriate message to send when a business flag confidence has been changed (e.g. from Suspected to Confirmed). When not configured no message will be sent.
Business Flag Discarded Notification	This role will identify the appropriate message to send when a business flag has been discarded. When not configured no message will be sent.
Business Flag Field Work Details	This role will identify the appropriate message to send when a business flag has completed that initiated field work. When not configured no message will be sent.

Copyright © 2026, Oracle and/or its affiliates.

Processing Role	Description
Business Flag Added Notification	This role will identify the appropriate message to send when a business flag has been created from within Oracle Utilities Meter Data Management. When not configured no message will be sent.

Extendable Lookups

The following extendable lookups should be configured when setting up business flag integrations:

Extendable Lookup	Description
Business Flag Standard Name	If external analytics applications will be communicating business flags using a non-standard name provide an entry in the Valid External Name with the non-standard name for each standard name communicated by external analytics applications.

Meter Operations Insights Business Flags

Meter Operations Insights are analytics produced by Oracle Utilities Data Intelligence (OUDI) that evaluate meter and service-point data to identify potential operational issues. OUDI publishes the resulting insights to Meter Solution Cloud Service, Customer Cloud Service, or Meter Data Management as Business Flags with supporting context, so users can prioritize and review potential issues in their existing MDM workflow.

Meter Operations Insights sends the following insights as business flags:

- Meter Bypass
- Frequent Power Outage
- Power Factor Monitoring

The Business Flag identifies the service point and includes context that supports review of the insight.

For general information about business flags, see Business Flags in the *Application Framework Administrative User Guide*. For general integration configuration, see Business Flag Integration and Configuring Business Flag Integration.

Meter Operations Insight Context

The following optional context values can be included with Meter Operations business flags:

Insight	Context Value	Description
Meter Bypass	Usage Before	Daily average kWh observed before the suspected bypass.
Meter Bypass	Usage After	Daily average kWh observed after the suspected bypass.
Frequent Power Outage	Power Outage Count	Number of power outages associated with the insight.
Power Factor Monitoring	Power Factor	Calculated power factor.

Copyright © 2026, Oracle and/or its affiliates.

Insight	Context Value	Description
Power Factor Monitoring	Usage Loss	Aggregated kWh loss for the duration of the business flag.

Context information is displayed on the business flag when it is available. A business flag can be created with or without context values.

Configuring Meter Operations Insight Business Flags

Configure a separate Service Point Business Flag Type for each Meter Operations insight you wish to enable. Use the Service Point Business Flag (D1-SPBusinessFlag) business object unless your implementation requires monitor-driven follow-up or field work.

The following table lists the business flag types to configure. Use the standard names and any delivered type codes supplied with your release.

Insight	Business Object	Description
Meter Bypass	Service Point Business Flag (D1-SPBusinessFlag)	Review the business flag using the standard Business Flag portal.
Frequent Power Outage	Service Point Business Flag (D1-SPBusinessFlag)	Review the business flag using the standard Business Flag portal.
Power Factor Monitoring	Service Point Business Flag (D1-SPBusinessFlag)	Review the business flag using the standard Business Flag portal.

The Service Point Business Flag (D1-SPBusinessFlag) business object does not create a service issue monitor, field activity, or investigation task. If your implementation requires monitor-driven follow-up or field work, configure the applicable business flag type using the Initiate SP Monitoring Business Flag (D1-SPMonitorBusinessFlag) business object. See Service Issue Monitors and Service Order Management for configuration details.

If the external system uses a name that differs from the configured standard name, add the external name to the **Business Flag Standard Name** extendable lookup. See Configuring Business Flag Integration for more information.

Configuring Measurement Identification

Meter Operations Insights uses the measurement identification configuration in Meter Solution Cloud Service and Meter Data Management to identify the customer-specific measurement channels used by the insights. Configure the mappings for the measurements required by the insights you enable. See Defining Standard Measurement Identifiers for more information.

For Power Factor Monitoring, configure the mappings for power factor and the supporting measurements used by the implementation, including kVA when applicable. If the required power factor data is not available in the system, configure pass-through measuring processing before enabling Power Factor Monitoring. See Measuring Component Filtering for more information.

Reviewing Meter Operations Insight Business Flags (Section)

Use the **Business Flag** portal to search for and review Meter Operations insight business flags. You can search by business flag type to review insight records across service points. Review the business flag context together with the service point information and follow the investigation process defined by your implementation.

Copyright © 2026, Oracle and/or its affiliates.

A Field Activity Can Determine Business Flag Confidence

When a business flag results in a field activity being issued and a field team is sent to investigate the service point the result of that field activity investigation will drive the resolution of the business flag. This is done through the field remarks left by the field team. The field activity remarks can either be defined as confirming the business flag or rejecting it. By selecting the appropriate field activity remark the field team will conclude the investigation into the business flag and the final confidence will be updated as appropriate.

For those field activity remarks that should either confirm or reject the business flag the Business Flag Confidence Update algorithm type should be provided for the field activity remark – activation plug-in spot.

Pease refer to algorithm type Business Flag Confidence Update (C1-BSFLGCFUP) for more detail.

Business Flags Can Create Alerts

Some business flags have a high enough importance that they should be highlighted to any user that is viewing data for the service point the business flag is related to. Whether a business flag should create an alert is controlled on the business flag type. Furthermore, the appropriate algorithm must be configured in the “Installation Options – Framework” “Algorithms” page for the “Control Central Alert” system event.

The product delivers the algorithm type “Highlight Open Business Flags”

Pease refer to algorithm type *Business Flag Confidence Update (C1-BSFLGCFUP)* for more detail.

Customer Engagement

Customer engagement integration allows information about communications sent from an external system such as Opower to be stored in the system as customer contacts. This information includes who the communication was for, what it was about, a reference to the communication content (content path), and a unique identifier from the external system (external reference ID). The account and person defines who the communication was for, while the communication product and event type defines what the communication was about. The content path allows the user to view the content of the communication. A single customer contact is created for each external reference ID.

Receiving Communication

For a customer contact to be created, a one-to-one mapping between the communication product/event type combination and a notification event type lookup value must be defined in the Communication Product Event Type Mapping (C1-CustEngageValueMapping) extendable lookup. The system is delivered with mappings for the Opower communication products and event types. Additionally, the following must also be defined for the notification event type lookup value:

- Notification Event Type
- Notification Type
- Customer Contact Type with Contact Action set to 'Record Customer Engagement'.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> An error is raised if a Notification Event Type or Notification Type is not defined. If a Customer Contact Type is not defined, the communication is skipped.

The system receives communications in two ways:

- Customer Engagement (C1-CustomerEngagement) inbound web service
- On-demand retrieval of communications for an account

The inbound web service is the primary mechanism through which communication are received. It is expected to be invoked periodically to receive communications for one or more accounts.

To enable on-demand retrieval of communications for an account, set the **Enable On Demand Communication Query** parameter in the Customer Engagement Integration Configuration (C1-CustEngageIntegration) extendable lookup. When enabled, recent communications for an account are retrieved from the external system in real time when the Customer Activity History zone is loaded. The on-demand retrieval is limited to communications sent after the contact date/time of the most recent customer engagement customer contact, if one exists. Otherwise, all communications for the account are retrieved. The **On Demand Communication Query Interval (Minutes)** parameter in the same extendable lookup determines how soon after that contact date/time the on-demand retrieval can be performed again. Set this parameter to zero to perform the on-demand retrieval each time the zone loads. The external system and outbound message type used in this process are defined in the same extendable lookup.

Error Handling

If an error occurs while processing a communication, a Customer Engagement Error (C1-CustomerEngagementError) business object instance, or error record, is created to store the communication information for later remediation and reprocessing. Only one non-final error record can exist for each communication external ID. Error records can be viewed and maintained in the Inbound Sync Request portal. An error record is created in the Error state and remains there until it is reprocessed or discarded. Communication information in the error record is reprocessed the same way as information received through the inbound web service or on-demand account communication retrieval.

Viewing Communication Content

Communication content can be viewed using the **View Communication** action button in the **Customer Contact** portal. On click of the button, a real-time outbound message is sent to the external system to retrieve a pre-authenticated resource (PAR) link for the content path stored in the customer contact. The PAR link then opens in a new window to display the communication content. The validity of the PAR link is determined by the external system. The external system and outbound message type used in this process are defined in the Customer Engagement Integration Configuration (C1-CustEngageIntegration) extendable lookup.

Oracle Fusion Customer Experience Cloud Service Integrations

This section provides information about the integrations that are available with Oracle Fusion Customer Experience Cloud Services.

- Service Request to Case Integration

Copyright © 2026, Oracle and/or its affiliates.

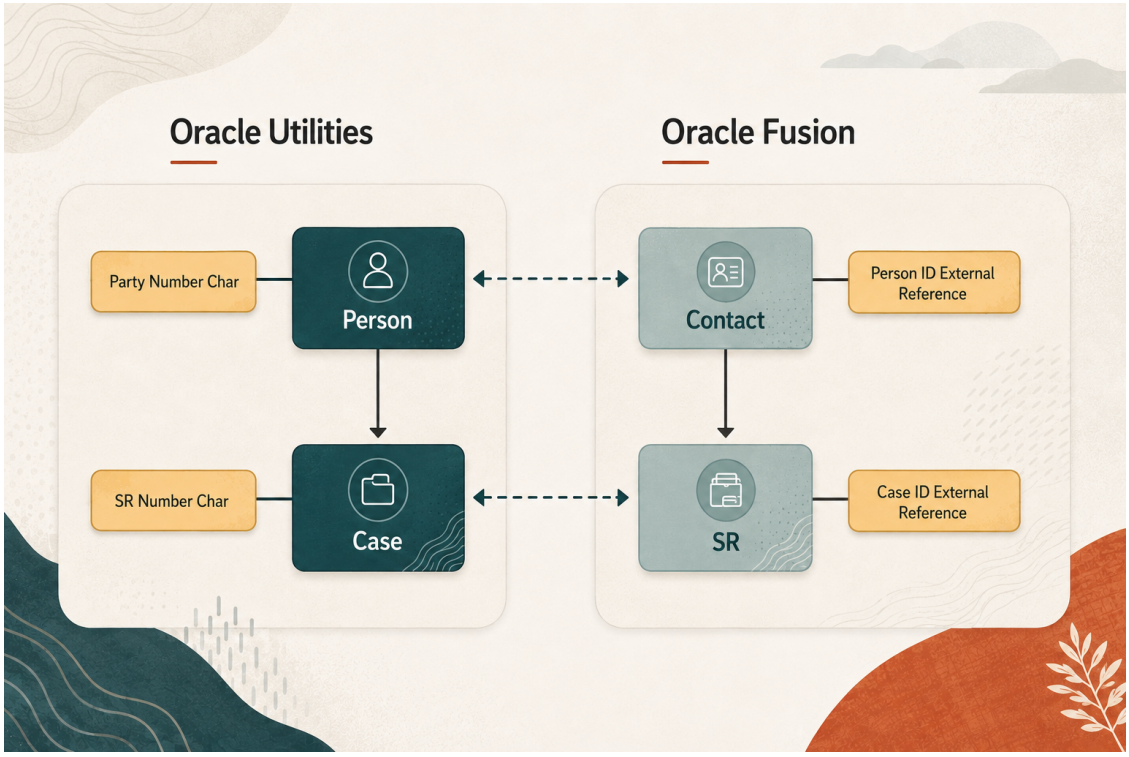
Service Request to Case Integration

Use this integration to synchronize selected Oracle Utilities cases with Oracle Fusion service requests (SRs). The integration is optional and supports these scenarios:

- Create or update a Fusion SR from a case.
- Create or update a case from a Fusion SR.
- Maintain cross-references between the case and Fusion SR.

The integration allows you to synchronize only the case types that your organization selects. The integration does not synchronize case or SR configurations, such as workflows, topics, or field setup. You can configure those items independently in each application.

This image illustrates the data flow of the integration:



Before continuing with the integration setup, be aware of the following:

- Case synchronization is optional and can be enabled only for selected case types.
- The integration uses cross-references rather than full configuration synchronization.
- Oracle Utilities-to-Fusion processing is asynchronous. Cloud deployments process messages near-real-time, while on-premises deployments use the configured monitor batch.
- This integration does not support an initial load of existing cases or SRs. It applies to new and updated records after you enable it.

Prerequisites

Confirm the following prerequisites:

Copyright © 2026, Oracle and/or its affiliates.

- Person synchronization between Oracle Utilities and Oracle Fusion is enabled and operational.
- Oracle Integration Cloud (OIC) is configured to route messages between Oracle Utilities and Oracle Fusion.
- The required case synchronization components are available in your release.
- Configuring the Service Request to Case Integration

Configuring the Service Request to Case Integration

The following sections provide details about the tasks you need to complete in your Oracle Utilities system.

Verify the SR Number Characteristic Type

Verify that the `C1-SRNO` characteristic type is valid for the Case entity.

> **Note**
>
> You need a valid case characteristic type for each case type you want to synchronize with Oracle Fusion.

Oracle Utilities uses this characteristic to store the Oracle Fusion SR number after a case is synchronized. The Oracle Fusion SR stores the case ID as its external reference.

Specify Eligible Case Types

Configure the `Case Synchronization` master configuration business object to list the case types that can be synchronized with Fusion, and specify the Oracle Fusion Service Request page URL. For example, `https://<FUSION_HOST>/fscmUI/redwood/service/ec/container/sr/details?srNumber=`

Only cases that use the specified case types are eligible for synchronization. Additionally, configure eligible case types to require a person.

Activate Data Synchronization Control

The Case entry is available in the **Data Synchronization Control** extendable lookup. Keep the entry inactive until all integration configuration and testing are complete. Activate it when you are ready to begin synchronization.

This table as an example of the values in the extendable lookup:

Setting	Value
Synchronization Name	Case
Status	Inactive
Maintenance Object	Case
External System	C1FUSION
Sequence	10
Synchronization Script	Case Data Synchronization
Near Real-Time	Enabled

Copyright © 2026, Oracle and/or its affiliates.

Configure the OIC Endpoint

Configure the message sender to use the Oracle Integration Cloud (OIC) endpoint for the Service Request to Case integration.

Confirm the endpoint-management approach for your deployment. In particular, determine whether the endpoint is supplied through a framework URL variable or through a customer-specific message-sender configuration.

For on-premises deployments, configure and schedule the required monitor batch for case synchronization requests.

Customer Profile Insights

The following section describes how customer insights from external sources can be stored in Accounts' Customer Profiles.

Refer to Understanding Customer Profiles for more information on the Customer Profile maintenance object.

- Uploading Customer Insights

Uploading Customer Insights

There are certain types of customer insights that can be uploaded from an external system. For example, the system could receive flat files containing affordability information, indicating specific Accounts that may be in the low-to-moderate (LMI) income segment. This customer insight can be loaded into the Account's Customer Profile, using the Customer Profile Business Flag Upload (`C1-CPRUP`) batch process.

Refer to Understanding Business Flags for general information on how Business Flags are processed.

How the Customer Profile Upload Works

The Customer Profile Upload is a plugin-driven upload process that reads a flat file (e.g. a CSV file) and invokes the 'File Upload' algorithm that is plugged in on the Batch Control.

The base product provides a 'File Upload' algorithm for the purposes of the LMI/affordability use case: Add LMI Customer Profile Business Flag (`C1-CPRBUSFLG`). This algorithm maps the LMI indicator's external name - i.e., `lmi_selection_flag` (referenced in the input file) - to the Standard Name that is configured in the External Insight Standard Name (F1-BusinessFlagStandardName) lookup value, `C1-LMI`. This Standard Name value maps`lmi_selection_flg` as a Valid External Name.

The Standard Name determines the Business Flag Type, which in turn, determines the Business Flag object to create.

The 'Additional Processing' algorithm plugged in on the Business Flag Type is then invoked, to add or update the Account's Customer Profile. The base product provides the algorithm, Add/Update Customer Profile Attributes (`C1CPAUCPA`), which is meant to be plugged in for a specific Customer Profile Type (defined in the algorithm parameter). The base algorithm performs the following:

- Checks if the Account already has an active Customer Profile of the configured type.
- Creates the Customer Profile when one does not exist.

Copyright © 2026, Oracle and/or its affiliates.

- Adds new attributes to (or updates existing attributes in) the Customer Profile, based on the Customer Profile Attribute Type definition. Existing attributes are updated based on the Occurrence setting:
  - If set to `Single`, the existing value is updated.
  - If set to `Single - Per Related Object`, the attribute that references the same related object (e.g., Service Agreement) is updated.
  - If set to `Multiple`, the new values are compared with the existing values. to determine what to add or delete.

Customer Profile creation and attribute changes are captured on the Customer Profile log, for audit purposes.

Customer Profile Upload Validation and Error Handling

The Customer Profile Upload performs initial validation of the file record, including confirmation that the account ID is provided and valid. Detailed validation of Customer Profile and attribute information occurs during Customer Profile Business Flag BO and Customer Profile BO processing. If an error is encountered, the Business Flag is moved to the Validation Error state and an exception log entry is created. Users can review, make corrections and retry processing. The upload record skips the record in error and proceeds to process the next records in the file.

Refer to the `C1-CustomerProfBusinessFlag` and `C1-CustomerProfile` business objects in the application, for more information on the Business Flag and Customer Profile schemas and lifecyle processing.

Setting Up the System for Customer Profile Uploads

The following are the steps to enable Customer Profile Batch Upload processing:

- Set up a Customer Profile Attribute Type for each unique attribute to capture in the Customer Profile. For example, with the LMI use case, there could be two attribute types - one for the LMI indicator (true/false switch) and another for the LMI Selection Reason (extendable lookup). Refer to Setting Up Customer Profile Attribute Types for more information.
- Set up the Customer Profile Type and link the new Customer Profile Attribute Types to it. Refer to Setting Up Customer Profile Types for more information.
- Add an Algorithm for the Add/Update Customer Profile Attributes (`C1CPAUCPA`) Algorithm Type and set its algorithm parameter to the new Customer Profile Type.
- Update the External Insight Standard Name (`F1-BusinessFlagStandardName`) Extendable Lookup, with the applicable Standard Name (lookup value) and its associated Valid External Name(s). For the LMI use case, the base product provides the `C1-LMI` lookup value. It just needs to be updated with any additional implementation-specific Valid External Name(s) that differ the `lmi_selection_flg` name that is already included.
- Set up a Customer Profile Business Flag Type using the Customer Profile Business Flag Type (`C1-CustomerProfBusFlagType`) business object:
  - Set the Standard Name. For example, in the LMI use case, set the Standard Name to the `C1-LMI` extendable lookup value or an implementation-specific Valid External Name, if any.
  - Configure attribute relationships, if any. This applies to cases where multiple attribute types are received and some attribute types provide more information related to the key attribute type. For example, with the LMI use case, the LMI indicator is defined

Copyright © 2026, Oracle and/or its affiliates.

with LMI Selection Reason as a related attribute. The LMI Selection Reason's external name (in the input file) is mapped to a corresponding Customer Profile Attribute Type.

  - Configure the 'Additional Processing' Algorithm with the new algorithm instance of Add/Update Customer Profile Attributes (`C1CPAUCPA`) algorithm type, created above.

- Set up other implementation-specific configuration. For example, for LMI Selection Reasons, add the valid values for the LMI Selection Reason extendable lookup, `C1-LMISelectionReasonLookup`.
- Set up Object Storage access. Refer to External File Storage for more information.

Customer Relationship Management Integration

Oracle Utilities Customer Care and Billing provides tools to facilitate the integration with customer relationship management applications such as Oracle Social Relationship Management (SRM).

The following sections describe this functionality.

- About Customer Relationship Requests
- Setting Up Customer Relationship Request Configuration
- Maintaining Customer Relationship Requests

About Customer Relationship Requests

A customer relationship request is used to process requests that are initiated from an external customer relationship management system.

The following sections highlight customer relationship request functionality.

- Social Relationship Management Requests
- Creating Customer Relationship Requests

Social Relationship Management Requests

Oracle Social Relationship Management (SRM) is an example of an external system that can initiate customer relationship requests.

Social network posts cover a vast range of topics, a number of which could be related to the users' concerns with their utility service. For example:

- A Facebook user could post on a community page about a power outage in a specific area.
- A Twitter user might post about specific problems with their utility service (*e.g.*, high bill complaints).

Social conversations of this nature can be monitored and managed via Oracle SRM. Oracle SRM’s Engage module allows its users to see all social conversations, identify customer-specific issues and route the messages to the appropriate systems/channels.

Requests from social relationship management systems typically include:

- Information about the original posting on the social network/community (*e.g.*, author’s identifier, link to the post, etc.).
- Information about the posting within the social relationship management system (*e.g.*, the user that sent the request to the utility, link to the post, etc.).

Copyright © 2026, Oracle and/or its affiliates.

- Additional details that will help with processing/resolution (*e.g.*, customer/account numbers, address, bill IDs, etc.).

Creating Customer Relationship Requests

Most customer relationship requests will be initiated from an external system.

The base product provides an Add Customer Relationship Request (C1–AddCustRR) inbound web service that can be used in a customer relationship management integration to create customer relationship request records. This inbound web service invokes a processing script that is configured in Customer Relationship Integration master configuration. The base product supplies an example of this processing script. It maps the inbound service's input data into the customer relationship request record. Refer to the C1-CreCustRR service script system data for processing details.

Setting Up Customer Relationship Request Configuration

The following topics highlight the general configuration steps required to use customer relationship request functionality and to integrate with a customer relationship management application such as Oracle Social Relationship Management (SRM).

- Customer Relationship Management Integration Configuration
- Defining Customer Relationship Request Types

Customer Relationship Management Integration Configuration

The integration with a customer relationship management application such as Oracle Social Relationship Management (SRM) requires the setup of master configuration data that controls the processing of your customer relationship requests.

To set up customer relationship integration master configuration, navigate using **Admin**, and then **General**, and then **Master Configuration**. You are brought to an all-in-one portal with options to add/edit the Customer Relationship Integration master configuration record. Refer to the embedded help for more details on how each section in this record is configured.

Defining Customer Relationship Request Types

A customer relationship request type defines properties that control how a customer relationship request is processed.

Refer to About Customer Relationship Requests for an overview of customer relationship request functionality.

To maintain customer relationship request types, open **Admin**, and then **Integration**, and then **Customer Relationship Request Type**, and then **Search**.

This is a standard All-In-One Portal.

The information captured on the customer relationship request type depends on the business objects supported by your implementation. Refer to the embedded help text for more information.

Maintaining Customer Relationship Requests

This section describes the functionality supported for viewing and maintaining customer relationship requests.

Copyright © 2026, Oracle and/or its affiliates.

Refer to About Customer Relationship Requests for an overview of customer relationship request functionality.

Navigate using **Menu**, and then **Integration**, and then **Customer Relationship Request**, and then **Search**. You are brought to a query portal with options for searching for customer relationship requests.

Once a customer relationship request record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Customer Relationship Request** zone provides basic information about a customer relationship request. Refer to the embedded help for more information.

The **Notes** zone allows users to capture additional notes to the customer relationship request.

RightNow Knowledge Integration

Oracle Utilities Customer Care and Billing provides an integration with Oracle RightNow Knowledge that allows users to view frequently asked questions about business processes/ functions that relate to the portal or page that they are currently viewing and/or working on.

The following sections describe this functionality.

- Displaying Frequently Asked Questions
- RightNow Knowledge Integration Configuration

Displaying Frequently Asked Questions

The Oracle RightNow Knowledge application can be used to develop content for frequently asked questions about business processes. It also functions within the Oracle Customer Care and Billing application. The RightNow KnowledgeSyndication widget defines the interface for the retrieval of this content.

The Oracle Customer Care and Billing application provides a **Frequently Asked Questions** sidebar zone that uses the RightNow KnowledgeSyndication widget to retrieve content from a RightNow Knowledge base instance. This zone is context-sensitive.

Configuration data controls how information is displayed in the **Frequently Asked Questions** sidebar zone. Refer to RightNow Knowledge Integration Configuration for more details.

RightNow Knowledge Integration Configuration

The integration with Oracle RightNow Knowledge requires the setup of master configuration data that controls how content from the RightNow Knowledge base is retrieved and presented on the **Frequently Asked Questions** sidebar zone. This configuration includes the following settings:

- The URL of the RightNow Knowledge base instance.
- The list of portals and pages for which the **Frequently Asked Questions** sidebar zone will display.
- Settings for each portal / page:
  - Search terms to pass to the RightNow KnowledgeSyndication widget. The base product supplies a number of common search term types. Implementation-specific search term types can be added.

Copyright © 2026, Oracle and/or its affiliates.

- Number of results/answers to display in the **Frequently Asked Questions** sidebar zone.

To set up the RightNow Knowledge Integration master configuration, navigate using **Admin**, and then **General**, and then **Master Configuration.** You are brought to an all-in-one portal with options to add/edit the RightNow Knowledge Integration master configuration record. Refer to the embedded help for more details on how each section in this record is configured.

External Applications

- Understanding External Applications
- Configuring External Applications
- Initiating Smart Meter Commands from an External Application

Understanding External Applications

External applications are applications and systems that are external to the Oracle Utilities meter data products, and can include customer information systems, outage management systems, or other types of applications.

External system service providers utilize processing methods to specify how the system sends and creates data used by the two applications. .

Refer to Understanding Process Methods for more information about processing methods.

External Applications Impact Data Import and Export

External Applications are configured to identify how a particular external system communicates data with Oracle Utilities Meter Data Management. This includes:

- The identifier type used to locate devices and measuring components. These are used both on import and export of data.
- The date/time format used in various data imports (i.e. whether or not the date/time format includes time zone information).

Please refer to the embedded help for more information about these fields.

Each external application can be associated to an external system which is used to define the messages that can be sent to that service provider and how each message is sent.

Configuring External Applications

This portal is used to display and maintain External Applications.

Refer to Understanding External Applications for more information.

You can access the portal from the **Admin**, then **Integration**, then **External Application**.

The following zones may appear as part of the portal's **Main** tab page:

- **External Applications List:** This zone lists all External Application records. Broadcast a record to display the details of the selected record.
- **External Application:** This zone provides information about the selected External Application.

Copyright © 2026, Oracle and/or its affiliates.

- **Processing Method List:** This zone provides the list of processing methods defined for the External Application.
- **Translation Method List:** This zone provides the list of translation methods defined for the External Application.
- **Inbound BOs Send By Service Provider:** This zone lists inbound Business Objects that are sent by this External Application. The identification is driven by the Business Object having a Business Object Option of type "Sent By Service Provider" that references the current External Application.

Initiating Smart Meter Commands from an External Application

Though often initiated as part of an orchestration activity (such as Enable Service or Exchange Meter) or from the Oracle Utilities Meter Data Management application, smart meter commands can also be initiated from external applications such as a customer information system such as Oracle Utilities Customer Care and Billing, a self-service application such as Oracle Utilities Digital Self-Service Transactions, or an outage management system such as Oracle Utilities Network Management System.

External applications initiate commands through the use of inbound web services, and outbound messages are used to send the results back to the initiating application. This section describes how smart meter commands can be initiated from external applications.

Command Processing Overview

This section provides a high level overview of how the system processes commands initiated from external applications.

For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific base package objects used. Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process. This example uses an "On-Demand Read - Interval" command.

Step	Process	Sample Object
1	External application initiates a command request via inbound web service	Inbound Web Service: D1-OnDemandReadInterval
2	Command activity is created in the Pending state.	Activity Business Object: D1-OnDemandReadInterval
3	The command activity goes through it's lifecycle, creating outbound communications as defined by the head end system, receiving the response.	
4	When the command enters the Completed state, an Enter algorithm sends the result to the initiating application.	Algorithm: Send On Demand Read to Edge Application (D1-SNDRDTEA)

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Sample Object
5	The algorithm uses a "Response - Success" processing method (based on the the Response - Success (D1SR) processing role) defined for the external application to create an outbound message (based on the outbound message type defined for the processing method).	Processing Method Business Object: How to Send Activity Related O\B Messages (D1-HowToSendActivityResponse) Outbound Message Business Object: Outbound Message for Communication BOs (D1-OutboundMessage)
6	The external system and message sender sends the outbound message to the external application.	

Inbound Web Services

This section provides information about the inbound web services available for initiating smart meter commands from an external application. The table below lists the smart meter commands available, and the specific inbound web services provided for each.

Command	Inbound Web Service
Device Status Check	Device Status Check (D1-DeviceStatusCheck)
On-Demand Read (Interval)	On Demand Read Interval (D1-OnDemandReadInterval)
On-Demand Read (Scalar)	On Demand Read Scalar (D1-OnDemandReadScalar)
Remote Connect	Remote Connect (D1-RemoteConnect)
Remote Disconnect	Remote Disconnect (D1-RemoteDisconnect)

See **Inbound Web Services** for more information about using inbound web services.

Required Data Elements

The inbound web service calls used to initiate commands must include some specific data elements in order to be successfully processed.

The table below lists the specific data elements required when initiating commands via inbound web service calls. Use the **WSDL** link to view the full web service description language definition for each command.

Data Element	Description	Used by
`requester`	The system or person submitting the command request. This should be the name of the external application.	Device Status Check On-Demand Read (Interval) On-Demand Read (Scalar) Remote Connect Remote Disconnect

Copyright © 2026, Oracle and/or its affiliates.

Data Element	Description	Used by
`utilityDeviceIdentifierNumber`	The ID of the device in the external system. This should be based on the **Utility Device ID Type** defined for the external application. For example, if the external application uses Serial Numbers for their device IDs, this would be the device's serial number.	Device Status Check On-Demand Read (Interval) On-Demand Read (Scalar) Remote Connect Remote Disconnect
`measurementDestination`	The destination for measurements returned by On-Demand read commands, from the Measurement Destination (MSRMT_DEST_FLG) lookup. Valid values include: • Calling Edge Application (D1EA) • Calling & Subscribing Edge Application (D1ES) • Measurement Subscribing Edge Application (D1SE)	On-Demand Read (Interval) On-Demand Read (Scalar)

Send to Edge Application Algorithms

As noted above, when the command enters the Completed state, an Enter algorithm sends the result to the initiating application. The table below lists the specific algorithms used with each command.

Activity Business Object	Send to Edge Application Algorithm
Device Status Check (D1-DeviceStatusCheck)	Send Device Status Check Notification to Edge Application (D1-SNDDVSTCK)
On Demand Read Interval (D1-OnDemandReadInterval)	Send On Demand Read to Edge Application (D1-SNDRDTEA)
On Demand Read Scalar (D1-OnDemandReadScalar)	Send On Demand Read to Edge Application (D1-SNDRDTEA)
Remote Connect (D1-RemoteConnect)	Send Remote Connect Notification to Edge Application (D1-SRCNTEA)
Remote Disconnect (D1-RemoteDisconnect)	Send Remote Disconnect Notification to Edge Application (D1-SRDNTEA)

Sending Messages Back to the External Application

Sending outbound communications to am external application involves configuring a number of entities. This section provides an overview of the objects required to send outbound communications to external applications.

Outbound Message Types

Outbound message types define the details of outbound messages sent to the external application. You should create an outbound message type for each different type of message to be sent to the external application.

Copyright © 2026, Oracle and/or its affiliates.

See **Outbound Messages** and **Defining Outbound Message Types** for more information about creating outbound message types.

Message Senders

Message senders define details for sending messages to external applications. You must create a message sender for each type of message to be sent to the external application.

Note: The Message Class MUST be set to **SOAPSNDR** (Sender for real-time HTTP / SOAP messages).

See **Message Senders** for more information about creating message senders.

External Systems

You must create an external system that represents the external application and defines the outbound message types that can be sent to the external application. The external system will be referenced on the external application.

You must configure an outbound message type on the external system for each type of message you will send to the external application. An external system's outbound message types are defined by the following:

- **Outbound Message Type**: The outbound message type (see above) created for the message
- **Processing Method**: The method by which the message will be sent. This should be set to "Real-time".
- **Message Sender**: The message sender (see above) created for the message.
- **Message XSL / Response XSL**: XSL files used to transform your message and the response to the message into the appropriate format used by the external application (Message XSL) and Smart Grid Gateway (Response XSL).

See **External Systems** for more information about creating external systems.

External Applications

The external applications that can initiate smart meter commands must be defined as External Applications. Examples can include:

- A customer information system (such as Oracle Utilities Customer Care and Billing)
- A self-service application (such as Oracle Utilities Digital Self-Service Transactions)

Information defined for external applications include:

- **External System**: The external system (see above) the defines the types of outbound messages to be sent to the external application.
- **Our Name/ID in Their System**: The value that the external application uses to identify our system.
- **Utility Device ID Type**: The Device ID Type that will be used when communicating with the external application and it will be the assumed Device ID Type for any device identifiers sent from the external application.
- **Utility Service Point ID Type**: The Service Point ID Type that will be used when communicating with the external application and it will be the assumed Service Point ID Type for any service point identifiers sent from the external application.

Copyright © 2026, Oracle and/or its affiliates.

- **Processing Methods**: The external application's processing methods define how data relevant to the application is sent and/or created. At least one processing method should reference the outbound message type used to send responses to the external application. External applications that initiate smart meter commands should use the following processing roles:
  - Response- Success: Used to send an acknowledgement that the command was successfully executed and the results of the command, if applicable.

Refer to **Understanding External Applications** and **Configuring External Applications** for more information about external applications.

Configuring Master Data Synchronization

Refer to Customer To Meter Master Data Synchronization for more information.

Oracle Utilities Operational Device Management

- Overview
- Configuring Master Data Synchronization

Overview

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle Utilities Operational Device Management. In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management:

- Oracle Utilities Meter Data Management is typically considered the "system of record" for service points (asset locations), contacts, and device installation information (install events)
- Oracle Utilities Operational Device Management is typically considered the "system of record" for assets (devices) and asset specification configuration (device configurations)

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions. The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management is based on data synchronization between the two systems.

Configuring Master Data Synchronization

The specific data synchronization flows supported between Oracle Utilities Meter Data Management (MDM) and Oracle Utilities Operational Device Management (ODM) include the following:

- **Asset-Device Synchronization**: As new assets and asset specification configurations are created or changed in ODM, corresponding devices and device configurations must be created or changed in MDM
- **Service Point/Contact - Asset Location/Contact**: As Service Points and/or Contacts are created or changed in MDM, corresponding Asset Locations and Contacts must be created or changed in ODM

Copyright © 2026, Oracle and/or its affiliates.

- **Install Events- Asset Location/Disposition**: As devices are installed/removed in MDM, corresponding changes to an asset's Disposition (location and status) must be made in ODM

This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured Inbound Web Services. Refer to the *Oracle Utilities Integration for Device Operations Implementation Guide* for more information about this integration and the data synchronization processes used by this integration.

Master Configuration

The following master configurations are used to configure the sync process between MDM and ODM:

Master Configuration	Name / Description
Master Data Synchronization Configuration	Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system.
Seeder Sync Request Master Configuration	Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.
ODM Integration Master Configuration	Specifies the external system used to represent Oracle Utilities Operational Management, and the URL for the Oracle Utilities Operational Management application. Also specifies the outbound message types used to send synchronization requests to Oracle Utilities Operational Management.

Inbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following inbound (to Oracle Utilities Mater Data Management) synchronization business objects:

Object Name	Business Object	Description
Device Ongoing Sync Request	D1-OngoingSyncRequestDevice	Used to synchronize devices and device configurations in Oracle Utilities Mater Data Management based on assets and asset specification configurations created/updates in Oracle Utilities Operational Device Management.

Copyright © 2026, Oracle and/or its affiliates.

Outbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following outbound (from Oracle Utilities Mater Data Management) synchronization business objects (based on the F1-SYNC REQ maintenance object)

Object Name	Business Object	Description
ODM Sync Request	D1-ODMContactSyncRequest	Used to synchronize contacts in Oracle Utilities Operational Device Management based on contacts created/updated in Oracle Utilities Meter Data Management
ODM Install Event Sync Request	D1-ODMInstallEventSyncRequest	Used to synchronize asset location/disposition in Oracle Utilities Operational Device Management based on install events created/ updated in Oracle Utilities Meter Data Management
ODM SP Sync Request	D1-ODMSPSyncRequest	Used to synchronize asset locations in Oracle Utilities Operational Device Management based on service points created/updated in Oracle Utilities Meter Data Management. This includes updating asset locations when equipment is attached to or remove from a service point.

Analytics Integrations

This section describes integrations with external analytics applications such as Oracle Utilities Data Intelligence.

- Business Flag Integration
- Meter Insights and Investigations
- Usage and Event Integration

Business Flag Integration

With the business flag integration with external analytics applications:

- Either Oracle Utilities Meter Solution Cloud Service/Meter Data Management or an external analytics application can initiate a business flag
- Oracle Utilities Meter Solution Cloud Service/Meter Data Management can notify external analytics applications of a change in status of a business flag:
  - if a business flag in error has been discarded
  - if a business flag's confidence has been updated

Copyright © 2026, Oracle and/or its affiliates.

  - with the results of field work initiated by a business flag

- Oracle Utilities Data Intelligence can also publish Meter Operations insights as service point business flags. See **Meter Operations Insights Business Flags** for insight-specific configuration and context information.

For additional details see About Business Flags in the Oracle Utilities Application Framework *Administrative User Guide*.

Meter Insights and Investigations

The Meter Insights and Investigations functionality leverages external analytics application business flag integrations to support receipt of revenue protection events (referred to as "insights") from external analytics applications for research and investigation using Oracle Utilities Meter Data Management (or Oracle Utilities Meter Solution Cloud Service). Revenue protection insights sent from external analytics applications for investigation are events that suggest potential theft or loss of revenue such as unexpected consumption from an inactive meter, evidence of meter tampering, discovery of an improperly installed meter (such as a meter installed upside down or in reverse), and so on. Insights received from external analytics applications create business flags, which in turn create related Meter Insight Investigation service tasks (referred to as “investigations”), which can then be researched and investigated using a dedicated **Meter Insights Investigations Viewer** portal.

See **Business Flags** in the *Administrative User Guide* for general information about the business flag functionality provided by the Oracle Utilities Application Framework.

See **Business Flag Integration** in the *Administrative User Guide* for more information about the Business Flag integration with external analytics applications.

See **Service Issue Monitors** in the *Business User Guide* and **Service Task Types** in the *Administrative User Guide* for general information about service issue monitors.

See **Using the Meter Insights Investigations Viewer Portal** in the *Business User Guide* for more information about viewing meter insight service issue monitors.

Meter Insights and Investigation Business Objects

Business flags and service issue monitors used by Meter Insights and Investigations functionality are based on specific business objects:

Type of Object	Business Object
Business Flag	Revenue Protection Insight (D1-RevenueProtectionInsight)
Service Issue Monitor (Service Task)	Meter Insight Investigation (D1-MeterInsightInvestigation)
Investigation Type (Service Task Type)	Meter Insight Investigation Type (D1-MtrInsightInvestigationType)

Meter Insights and Investigation Processing

The table below provides a brief description of the process by which insights are received and processed, and lists example objects used by the Oracle Utilities Meter Data Management (and Oracle Utilities Meter Solution Cloud Service). Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Example Data
1.	Oracle Utilities Analytics Insights sends events ("insights") to Meter Data Management, via the Business Flag integration, creating business flags based on the business flag type configuration.	Business Flag Business Object: Revenue Protection Insight (D1-RevenueProtectionInsight)
2.	An Enter algorithm on the "Create Service Issue Monitor" state creates a Service Issue Monitor based on the Service Task type defined on the business flag's type. The investigation is created in "Pending" status.	Enter Algorithm: Business Flag - Create Service Issue Monitor (D1-CREATESIM) Note: This algorithm is on the parent business object (D1-SPMonitorBusinessFlag). Service Issue Monitor (Service Task) Business Object: Meter Insight Investigation (D1-MeterInsightInvestigation)
3.	The investigation evaluates the service point (based on configuration of the service task type). • If there are no issues, the investigation moves into the Discarded (No Investigation Needed) state. • Otherwise, the investigation continues to the Open state.	Evaluate Algorithm: Meter Insight Investigation - Analyze Service Point For Service Issues (D1-MII-ANZ)
4.	An Enter algorithm on the Open state creates a To Do entry.	Open Enter Algorithm: Meter Insight Investigation - Create To Do Entry (D1-MIICRTODO)
5.	Once the investigation is in the Open state, a business user can use the **Meter Insights Investigation Viewer** portal to examine the issue. See "Using the Meter Insights Investigation Portal" in the *Business User Guide* for more information about using this portal. The user can add comments and update the investigation to one of the following states: • Waiting for Field Work • Issue Resolved • No Issue • Discarded When the investigation moves into the Issue Resolved, No Issue, or Discarded states, it transitions the business flag the Completed state with a Final Confidence based on the investigation’s state (see below).	

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Example Data
6.	If the service issue monitors enters the Waiting for Field Work state, a Monitor algorithm evaluates the Completion Criteria on the Service Task Type and transitions the investigation to one of the following states: • Fieldwork Error • Open • Issue Resolved • No Issue • Inconclusive When the investigation moves into the Issue Resolved, No Issue, or Inconclusive states, it transitions the business flag the Completed state with a Final Confidence based on the investigation's state (see below).	Waiting For Field Work Monitor Algorithm: Evaluate Completion Criteria (D1-MIIEVCMCR)
7.	If the service issue monitors enters the Issue Resolved state: • An Enter algorithm completes the To Do entry • An Enter algorithm transitions the Business Flag to the Completed state (Final Confidence: Confirmed)	Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE) Enter Algorithm: Transition Business Flags - Completed (D1-TRBUSFLGC)
8.	If the service issue monitors enters the No Issue state, • An Enter algorithm completes the To Do entry • Transitions the Business Flag to Completed (Final Confidence: Rejected)	Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE) Enter Algorithm: Transition Business Flags - Rejected (D1-TRBUSFLGR)
9.	If the service issue monitors enters the Inconclusive state, • An Enter algorithm completes the To Do entry • Transitions the Business Flag to Completed (Final Confidence: Suspected)	Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE) Enter Algorithm: Transition Business Flags - Suspected (D1-TRBUSFLGI)
10.	If the service issue monitors enters the Discarded state • An Enter algorithm completes the To Do entry • Transitions the Business Flag to Completed (Final Confidence: Rejected)	Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE) Enter Algorithm: Transition Business Flags - Rejected (D1-TRBUSFLGR)

- Configuring Meter Insights and Investigations

Copyright © 2026, Oracle and/or its affiliates.

Configuring Meter Insights and Investigations

Configuring the Meter Insights and Investigations functionality involves creation and configuration of the following:

- Business Flag Types
- Extendable Lookup Values
- To Do Types
- Service Tasks Types
- Master Configuration

Business Flag Types

Create a business flag type for each type of revenue protection insight that will be received from external analytics applications, including the following:

- **Business Object**: Service Point Monitor Business Flag Type (D1-SPMonitorBusinessFlagType)
- **Related Transaction BO**: Revenue Protection Insight (D1-RevenueProtectionInsight)
- **Service Issue Monitor Type**: The service task type for investigations created from business flags of this type.

> **Note**
>
> Because the **Service Issue Monitor Type** field is required, when initially creating business flag types, select a placeholder Service Issue Monitor Type to enable you to save the business flag type. After you have created the appropriate service task types, return to each business flag type and select the correct Service Issue Monitor Type.

Define other fields based on your business requirements.

Extendable Lookups

The **Meter Insights Investigation Viewer** portal uses the following profile extendable lookups when displaying consumption and insights (events):

- **360 View Event Bar Profile**: Defines events to highlight that are pertinent to investigations, including your new business flag types
- **Final Values Overlay Profile**: Defines values to graph that are pertinent to investigations

Use the **Extendable Lookup** portal to create new extendable lookup values:

1. Create a new 360 View Event Bar Profile extendable lookup value that includes your business flag types as well as other events pertinent to your investigations.
2. Create a new Final Values Overlay Profile extendable lookup value that includes the values to graph that are pertinent to your investigations.

These extendable lookup values will be referenced by your service task types.

Copyright © 2026, Oracle and/or its affiliates.

To Do Types

Use the **To Do Type** portal to create a To Do Type (or update an existing To Do Type) with the Set Investigation Priority (D1-STINVSGPR) algorithm defined for the **Calculate Priority** system event.

These To Do Types will be referenced by your service task types.

Service Task Types

Create a service task type for each type of revenue protection insight (each type of investigation) that will be received from Oracle Utilities Analytics Insights, including the following:

- **Business Object**: Meter Insight Investigation Type (D1-MtrInsightInvestigationType)
- **Related Transaction BO**: Meter Insight Investigation (D1-MeterInsightInvestigation)
- **Evaluation Criteria**: The criteria used to evaluate the service point, including the existence of business flags of specified types (the business flag types you created earlier). See the embedded help for more details.
- **Service Investigation Order**: The field activity type and Field Task Type for field activities created by the investigation.
- **Completion Criteria**: The criteria used to determine if the service investigation order (field activity) should be completed and the result to be returned to the investigation. See the embedded help for more details.
- **Consumption Data Source**: Details that specify the types of consumption data will be displayed and how it is displayed in the **Meter Insights Investigation Viewer** portal, including the 360 Event Bar Profile and Final Values Overlay Profile extendable lookup values created earlier.
- **To Do Information**: The To Do Type and To Do Role used for To Do Entries created by the investigation. This should reference the To Do Type configured earlier.

Note: After you have created the appropriate service task types, return to the corresponding business flag type and select the correct service task type from the **Service Issue Monitor Type** drop-down list.

Master Configuration

Create a Meter Insights Configuration master configuration to capture key details used by Oracle Utilities Analytics Insights when creating insights including:

- Tolerances details related to specific types of insights (Flipped Meter, Unauthorized Consumption, and Leak Detection)
- Core sources of consumption for electric, gas, and water service types (including service point types, and unit of measure/time of use/service quantity identified combinations for each)

The master configuration is synchronized with Oracle Utilities Analytics Insights, and is used by Oracle Utilities Analytics Insights when generating insights for potential investigation.

Usage and Event Integration

Oracle Utilities Smart Grid Gateway adapters can be configured to publish initial measurement and device event data for use in external analytics applications or other external systems.

Copyright © 2026, Oracle and/or its affiliates.

Published data is posted to a user-defined directory that Oracle Utilities Analytics Insights can monitor for import.

Initial measurement and device event data is published in the “native” data format (the format of the initial measurement and device event seeder business objects). This format includes normalized unit of measure, condition, and device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published via this feature is published prior to VEE processing. In addition, filtering can NOT be applied to initial measurement or device event data published via this feature.

- Configuring Analytics Usage and Event Integration

Configuring Analytics Usage and Event Integration

Oracle Utilities Smart Grid Gateway adapters use OSB and BPEL components when publishing initial measurement and device event data for integration with external analytics applications.

OSB Projects

The following “CM” Oracle Service Bus projects are used to configure the integration between SGG and external analytics applications:

Adapter	OSB Projects
Landis+Gyr	SGG-D3-USAGE-CM (used to publish usage) SGG-D3-EVENT-CM (used to publish device events)
NES	SGG-D4-USAGE-CM (used to publish usage) SGG-D4-EVENT-CM (used to publish device events)
MV90	SGG-D5-USAGE-CM (used to publish usage)
Sensus	SGG-D6-USAGE-CM (used to publish usage) SGG-D6-EVENT-CM (used to publish device events)
Silver Spring Networks	SGG-D7-SSNXML-CM (used to publish usage) SGG-D7-CSV-CM (used to publish device events)
Itron OpenWay	SGG-D8-ITRONXML-CM (used to publish usage) SGG-D8-EXCEPTION-CM (used to publish device events)
Adapter Development Kit	SGG-DG-SEEDER-CM (used to publish usage and device events)

The following components provided with these projects are used in publishing measurement data and device events to Oracle Utilities Analytics Insights:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This business service is specified in the EnvironmentSettings.xq file (see below).

Copyright © 2026, Oracle and/or its affiliates.

- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Enabling Data Publishing

Publishing initial measurement and device event data is enabled by referencing a publisher business service (DataRakerBusinessService) in the publishServices/service element in the EnvironmentSettings.xq files provided with the OSB projects listed above as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

BPEL Composites

The SGGDRIntegration BPEL composite handles publishing the data to external analytics applications or other systems.

Configuring Publishing Output

The following parameters can be used to configure details of how the data is provided to external analytics applications, including the directory where files are posted for external analytics applications to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description	Default Value
SOA_DR_INTEGRATION_DIR	Mount point/directory from which external analytics applications will consume the converted XML files.	/spl/sploutput/int

Oracle Utilities Analytics Visualization

- Timeliness and Quality Aggregations Overview
- Consumption Analytics Aggregations Overview

Timeliness and Quality Aggregations Overview

Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service can be integrated with Oracle Utilities Analytics Visualization (OUAV) to allow users to view analytics data based on the quality and timeliness of measurement data.

This section describes integration of Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service with Oracle Utilities Analytics Visualization (OUAV).

See Operational Analytics in the *Administrative User Guide* for general information about integrating with Oracle Utilities Analytics Visualization.

- Base Package Timeliness and Quality Aggregation Measuring Components and Measurement Business Objects
- Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization
- Timeliness and Quality Aggregation Processing for Use with Oracle Utilities Analytics Visualization

Base Package Timeliness and Quality Aggregation Measuring Components and Measurement Business Objects

Oracle Utilities Meter Data Management and Oracle Utilities Customer To Meter include a number of standard timeliness and quality aggregation processes delivered as part of the base product that can be used with Oracle Utilities Analytics Visualization (OUAV). These aggregate measurement data (including quantities and counts) for constituent measuring components based on the following dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

The table below lists the names (and business object codes) for these aggregators, along with the corresponding Oracle Utilities Analytics Visualization subject area.

Copyright © 2026, Oracle and/or its affiliates.

Aggregator	Description
Measurement Timeliness Quantity Aggregator Business Object: D2-MsrmtTimelinessQuantityAggr OUAV Subject Area: Meter - Aggregated Timeliness Quantity	Aggregates measurement quantities that arrived on time or are late.
Measurement Timeliness Count Aggregator Business Object: D2-MsrmtTimelinessCountAggr OUAV Subject Area: Meter - Aggregated Timeliness Count	Aggregates measurement counts that arrived on time or are late.
Measurement Quality Count Aggregator Business Object: D2-MsrmtQualityCountAggregator OUAV Subject Area: Meter - Aggregated Measurement Count	Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes.
Measurement Measured Quantity Aggregator Business Object: D2-MeasuredQuantityAggregator OUAV Subject Area: Meter - Aggregated Measurement Quantity	Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes.

Refer to the Oracle Utilities Analytics Visualization User Guide for more information about Oracle Utilities Analytics Visualization subject areas.

In addition, the base package also includes measurement business objects used to store aggregated measurement data. These include:

- Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- Quality Count Measurement (D2-QualityCountMsrmt)
- Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)

Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization

This topic outlines the configuration needed for timeliness and quality aggregations used with Oracle Utilities Analytics Visualization.

Master Configuration

Create a Timeliness Master Configuration, defining the following:

- **Heating Degree Days Factor**: A number-based factor that contains heating degree days. These are aggregated and stored on the Measurement Measured Quantity Aggregator.
- **Cooling Degree Days Factor**: A number-based factor that contains cooling degree days. These are aggregated and stored on the Measurement Measured Quantity Aggregator.
- **Number of Hours to Consider IMD as Late**: The number of hours after which an initial measurement is considered late.

Copyright © 2026, Oracle and/or its affiliates.

- **Buckets**: Value Identifiers that specify the late buckets used by the Measurement Timeliness Count and Measurement Timeliness Quantity aggregators.

Extendable Lookup: BI Aggregation — Usage Group Mapping

If your implementation makes use of dynamic usage calculation group determination on usage subscriptions (via the **Calculation Group Determination Override Algorithm** on the usage subscription type, the **Rate History** collection on the usage subscription, and the Customer Rate Schedule extendable lookup (D2-CCBRateScheduleLookup)), you will need to configure the BI Aggregation Usage Group Mapping extendable lookup (D2-UsageCalculationGroupLookup) with the usage calculation groups you wish to aggregate. If in the future you add new usage calculation groups, you will also need to add those usage calculation groups to this extendable lookup if you want them to be aggregated.

Note that an aggregation measuring component will be created for all dimension combinations and each usage calculation group in the extendable lookup (this is done to alleviate the need to execute the usage group determination algorithm for all usage subscriptions during the dimension scanning process).

> **Note**
>
> Configuring this extendable lookup is required ONLY if using dynamic usage calculation group determination.

Measuring Component Types

Create a master aggregator measuring component type and sub aggregator measuring component types for the four types of aggregators outlined above. Details for each aggregator measuring component type are listed below.

> **Note**
>
> When creating the master aggregator measuring component type, we recommend setting the **Aggregation Horizon** at 7 days or less. See Aggregation Recommendations for more information.

Measurement Measured Quantity Aggregator

- **Measuring Component Type Business Object**: Master Measurement Aggregator Type (D2-MasterMsrmtAggregatorType)
- **Measuring Component Business Object**: Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator)
- **Measurement Business Object**: Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- **Value Identifiers** (recommended):

Value Identifier Type	Description	Unit of Measure
Measurement	kWh	Kilowatt-Hours
Value 01	Regular kWh	Kilowatt-Hours
Value 02	Estimated kWh	Kilowatt-Hours

Copyright © 2026, Oracle and/or its affiliates.

Value Identifier Type	Description	Unit of Measure
Value 03	User-Edited kWh	Kilowatt-Hours
Value 06	Count	Count
Value 07	Heating Degree	Fahrenheit
Value 08	Cooling Degree	Fahrenheit
Value 09	Ave kWh	Kilowatt-Hours

- **Sub Aggregator MC Types** (see below):
  - Measurement Quality Count Aggregator Type
  - Measurement Timeliness Count Aggregator Type
  - Measurement Timeliness Quantity Aggregator Type

Measurement Quality Count Aggregator

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type (D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Quality Count Aggregator (D2-MsrmtQualityCountAggregator)
- **Measurement Business Object**: Quality Count Measurement (D2-QualityCountMsrmt)
- **Value Identifiers** (recommended):

Value Identifier Type	Description	Unit of Measure
Measurement	kWh	Kilowatt-Hours
Value 01	Regular Count	Count
Value 02	Estimated Count	Count
Value 03	User-Edited Count	Count
Value 04	N Meas/No IMD Count	Count
Value 05	No Meas/IMD Exists Count	Count
Value 06	Read Outage Count	Count
Value 07	Read Other Count	Count
Value 08	Missing Count	Count

Measurement Timeliness Count Aggregator

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type (D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Timeliness Count Aggregator (D2-MsrmtTimelinessCountAggr)
- **Measurement Business Object**: Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- **Value Identifiers** (recommended):

Value Identifier Type	Description	Unit of Measure
Measurement	kWh	Kilowatt-Hours

Copyright © 2026, Oracle and/or its affiliates.

Value Identifier Type	Description	Unit of Measure
Value 01	On Time Count	Count
Value 02	Late Count	Count
Value 03	Very Late Count	Count
Value 04	Very, Very Late Count	Count
Value 05	Missing Count	Count

Measurement Timeliness Quantity Aggregator

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type(D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Timeliness Quantity Aggregator (D2-MsrmtTimelinessQuantityAggr)
- **Measurement Business Object**: Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)
- **Value Identifiers** (recommended):

Value Identifier Type	Description	Unit of Measure
Measurement	kWh	Kilowatt-Hours
Value 01	On Time kWh	Kilowatt-Hours
Value 02	Late kWh	Kilowatt-Hours
Value 03	Very Late kWh	Kilowatt-Hours
Value 04	Very, Very Late kWh	Kilowatt-Hours

Activity Type

Create a Measurement Quantity Scanner activity type, based on the following business objects:

- **Business Object**: Measurement Quantity Scanner Type
- **Related Transaction Business Object**: Measurement Quantity Scanner

Timeliness and Quality Aggregation Processing for Use with Oracle Utilities Analytics Visualization

This topic describes the processes involved in running timeliness and quality aggregation processing for use with Oracle Utilities Analytics Visualization.

Execute Measurement Quantity Scan

Create an activity of the Measurement Quantity Scanner activity type, and process the activity (by clicking **Scan** in the **Record Action** section in the **Activity** zone).

This process uses the Measurement Quality and Timeliness Dimension Scanner batch control (D1-MQTDS) which creates measuring component aggregators that will also serve as the measuring component dimensions. This process will create a measuring component aggregator for each unique combination of Postal Code, City, Device Type, Head-End System,

Copyright © 2026, Oracle and/or its affiliates.

Usage Group, Market, Service Type, Geographic Code, Manufacturer/Model, Service Provider (where Market Relationship Type is specified on the dimension scanner activity).

> **Note**
>
> The Measurement Quality and Timeliness Dimension Scanner batch control (D1-MQTDS) supports multi-threading.

This step will create aggregator measuring components for all **Valid MC Types to Aggregate** defined on the master aggregator type (Measurement Measured Quantity Aggregator).

The created aggregator measuring component's characteristics will contain the Postal Code, City, Device Type, Head-End System, Usage Group, Market, Service Type, Geographic Code, Manufacturer/Model, Service Provider information.

Execute Aggregation

Once the measuring component aggregators have been created, run the Aggregation Monitor (D2-AGG) batch process. This will process ALL active master aggregators (together with the linked sub aggregators) that satisfy the batch filter parameters. Aggregated measurements will be created for the span of the aggregator horizon which is the current date less the horizon days in intervals defined on the master aggregation measuring component type (Measurement Measured Quantity Aggregator).

Condition Top Range and Condition Bottom Range parameters on the Aggregate Measurement Counts and Quantity (D2-AGG-MCQ) algorithm (an Enter algorithm on the Aggregate state of the Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator) business object) define condition ranges for quantity and count aggregation buckets.

This batch process should be run on a scheduled reoccurring basis.

Aggregated Data

Aggregated measurements are created in the Measurement (D1_MSRMT) and Aggregation Measurement (D1_AGG_MSRMT) tables, using the following base package measurement business objects:

- Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- Quality Count Measurement (D2-QualityCountMsrmt)
- Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)

Oracle Utilities Analytics Visualization Fact views access the Aggregation Measurement table.

Oracle Utilities Analytics Visualization Dimension views access the Measuring Component and Measuring Component Characteristics tables.

Consumption Analytics Aggregations Overview

This section outlines the configuration needed to support integration with Oracle Utilities Analytics Visualization for the following consumption-based analytics:

- Usage by Network Location
- Usage by Region

Copyright © 2026, Oracle and/or its affiliates.

- Usage by Customer Classification
- Unaccounted for Losses

These analytics are accessed in Oracle Utilities Analytics Visualization (OUAV) via the following subject areas:

Analytics	OUAV Subject Areas
Usage by Network Location Usage by Region	Meter - Distribution Nodes
Usage by Customer Classification	Meter - Customer Classification
Unaccounted for Losses	Meter - Downstream Usage Meter - Downstream Service Points Meter - Facility Usage Meter - Facilities

- Configuring Dynamic Aggregation for Consumption Analytics
- Dynamic Aggregation Processing for Consumption Analytics

Configuring Dynamic Aggregation for Consumption Analytics

This section outlines the configuration needed for consumption analytics aggregations used with Oracle Utilities Analytics Visualization.

Consumption analytics leverage data generated through dynamic aggregation, and requires configuration of the following objects:

- Data Sources
- Aggregator Measuring Component Types
- Aggregation Groups
- Measuring Component Sets

Refer to **Dynamic Aggregation** for more information about configuring dynamic aggregation.

Note: The configuration outlined below is based on data in the demonstration database. This sample configuration uses Electric service for Usage by Network Location, Region, and Customer Classification, and Gas service for Unaccounted for Losses.

Data Sources

Create the following data sources, based on the Measurements from US Service Points (D1-DataSourceMeasurements) business object:

For Usage by Network Location, Usage by Region, and Usage by Customer Classification (Electric)

Field / Section	Value
Data Source	CM-E-KWH-60-TRANS
Description	Electric kWh 60min Transformer Interval Measurements
Status	Active
Data Source Class	Measurements from US Service Points
Interval / Scalar	Interval

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Sourced Measuring Component Type	

Measuring Component Type	Value Identifier Type
Measuring Component Types for measuring components to be aggregated	Measurement

For Unaccounted for Losses (Gas)

Field / Section	Value
Data Source	CM-G-CCF-60-FACIL
Description	Gas CCF 60min Facility Interval Measurements
Status	Active
Data Source Class	Measurements from US Service Points
Interval / Scalar	Interval
Sourced Measuring Component Type	

Measuring Component Type	Value Identifier Type
Measuring Component Types for measuring components to be aggregated	Measurement

Aggregator Measuring Component Types

Create the following aggregator measuring component types, based on the Dynamic Aggregator - Type (D1-AggregationIntervalType) business object:

For Usage by Network Location, Usage by Region, and Usage by Customer Classification (Electric)

Field / Section	Value
Measuring Component Type	CM-E-I-AGG-KWH-60-TRANS
Description	Electric - Interval - Aggregation - kWh 60min Transformer Load
Measuring Component Business Object	Dynamic Aggregator - Interval (D1-AggregatorInterval)
Measurement Business Object	Aggregation Measurement (D1-AggregationMeasurement)
Measuring Component Class	Aggregator
Service Type	Electric
Allow Negative Consumption	Allowed
Consumptive/Subtractive	Consumptive
Interval Size	01:00:00
Data Source	Electric kWh 60min Transformer Interval Measurements

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Value Identifiers	

Value Identifier Type	Short Hand Description	Unit of Measure
Measurement	kWh	Kilowatt-Hours

For Unaccounted for Losses (Gas)

Field / Section	Value
Measuring Component Type	CM-G-I-AGG-CCF-60-FACIL
Description	Gas - Interval - Aggregation - CCF 60min Downstream
Measuring Component Business Object	Dynamic Aggregator - Interval (D1-AggregatorInterval)
Measurement Business Object	Aggregation Measurement (D1-AggregationMeasurement)
Measuring Component Class	Aggregator
Service Type	Gas
Allow Negative Consumption	Allowed
Consumptive/Subtractive	Consumptive
Interval Size	01:00:00
Data Source	Gas CCF 60min Facility Interval Measurements
Value Identifiers	

Value Identifier Type	Short Hand Description	Unit of Measure	SQI	Value Calculation Method
Measurement	CCF	100 Cubic Feet (Gas)		Sum
Value 01	Count	Count		Count
Value 02	Average	100 Cubic Feet (Gas)	Average	Average
Value 03	Min	100 Cubic Feet (Gas)	Minimum	Min
Value 04	Max	100 Cubic Feet (Gas)	Maximum	Max

Aggregation Groups

Create the following aggregation groups based on the Aggregation Group (D1-AggregationGroup) business object:

For Usage by Network Location and Usage by Region (Electric)

Field / Section	Value
Aggregation Group	CM-E-CONS-TRANS
Description	Electric Dynamic Consumption

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Status	Finalized
Aggregation Category	Analytics
Time Zone	US Pacific Time
Aggregation Schedule	• **Aggregation Schedule Type**: Defined Lag • **Aggregation Cut-Off Time**: 0:00:00 • **Aggregation Horizon**: 24:00:00 • **Aggregation Lag**: 48:00:00
Aggregation Measuring Component Sets	

Sequence	Measuring Component Set	Initiation Method	Measuring Component Set Class
10	Electric - Consumption - Transformer - Interval	Manual	Aggregation Foundation

For Usage by Customer Classification (Electric)

Field / Section	Value
Aggregation Group	CM-E-CONS-PROF
Description	Consumption - Profile Class
Status	Finalized
Aggregation Category	Analytics
Time Zone	
Aggregation Schedule	• **Aggregation Schedule Type**: Defined Lag • **Aggregation Cut-Off Time**: 0:00:00 • **Aggregation Horizon**: 24:00:00 • **Aggregation Lag**: 48:00:00
Aggregation Measuring Component Sets	

Sequence	Measuring Component Set	Initiation Method	Measuring Component Set Class
10	Consumption - Profile Class - Interval	Manual	Aggregation Foundation

For Unaccounted for Losses (Gas)

Field / Section	Value
Aggregation Group	CM-G-LOSS
Description	Gas Dynamic Consumption
Status	Finalized
Aggregation Category	Analytics
Time Zone	US Pacific Time

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Aggregation Schedule	• **Aggregation Schedule Type**: Defined Lag • **Aggregation Cut-Off Time**: 0:00:00 • **Aggregation Horizon**: 24:00:00 • **Aggregation Lag**: 48:00:00
Aggregation Measuring Component Sets	

Sequence	Measuring Component Set	Initiation Method	Measuring Component Set Class
10	Gas - Consumption - Downstream - Interval	Automatic	Aggregation Foundation

Measuring Component Sets

Create the following measuring component sets based on the Aggregation Foundation Set (D1-AggregationFoundationSet) business object:

For Usage by Network Location and Usage by Region (Electric)

Field / Section	Value
Measuring Component Set	CM-E-CONS-TRANS
Status	Active
Description	Electric - Consumption - Transformer - Interval
Measuring Component Type	Electric - Interval - Aggregation - kWh 60min Transformer Load
Aggregation Group	Electric Dynamic Consumption
Measuring Component Set Class	Aggregation Foundation
Allow Partial Constituent Aggregation	Yes
Extract	Do Not Extract
General Processing	
Criteria Processing	• **Non-Individually Managed Formula**: TFORM
Dimensions	

Source Type	Source Entity	Column/Type	Represent Missing Data
Table	Service Point - MDM	City	No
Table	Service Point - MDM	Postal	No
Table (Demo)	Service Point Facility - MDM	Facility ID	No
Table (Demo)	Service Point Facility - MDM	Facility Relationship Type	No

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Criteria	• **Criteria Name**: TFORM • **Source Type**: Table (Demo) • **Source Entity**: Service Point Facility - MDM • **Column/Type**: Facility Relationship Type • **Criteria Type**: Include • **Value**: Commodity Distribution

For Usage by Customer Classification (Electric)

Field / Section	Value
Measuring Component Set	CM-E-CONS-PROF
Status	Active
Description	Consumption - Profile Class - Interval
Measuring Component Type	Electric - Interval - Aggregation - kWh 60min Transformer Load
Aggregation Group	Consumption - Profile Class
Measuring Component Set Class	Aggregation Foundation
Allow Partial Constituent Aggregation	Yes
Extract	Do Not Extract
General Processing	• **Sequence**: 10 • **Initiation Method**: Manual • **Imported from External Source**: No
Criteria Processing	
Dimensions	

Source Type	Source Entity	Column/Type	Represent Missing Data
Characteristic *	Usage Subscription Characteristics*	Procurement Class*	No*
Table	Service Point - MDM	City	No
Table	Service Point - MDM	Postal	No
Table	Usage Subscription	Usage Subscription Type	No
Table (Demo)	Service Point Facility - MDM	Facility ID	No

* Optional

Criteria	

For Unaccounted for Losses (Gas)

Field / Section	Value
Measuring Component Set	CM-G-LOSS

Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field / Section</th><th>Value</th></tr>
<tr><td>Status</td><td>Active</td></tr>
<tr><td>Description</td><td>Gas - Consumption - Downstream - Interval</td></tr>
<tr><td>Measuring Component Type</td><td>Gas - Interval - Aggregation - CCF 60min Downstream</td></tr>
<tr><td>Aggregation Group</td><td>Gas Dynamic Consumption</td></tr>
<tr><td>Measuring Component Set Class</td><td>Aggregation Foundation</td></tr>
<tr><td>Allow Partial Constituent Aggregation</td><td>Yes</td></tr>
<tr><td>Extract</td><td>Do Not Extract</td></tr>
<tr><td>General Processing</td><td></td></tr>
<tr><td>Criteria Processing</td><td>Non-Individually Managed Formula: TFORM</td></tr>
<tr><td>Dimensions</td><td>
<table>
<tr><th>Source Type</th><th>Source Entity</th><th>Column/Type</th><th>Represent Missing Data</th></tr>
<tr><td>Table</td><td>Service Point - MDM</td><td>City</td><td>No</td></tr>
<tr><td>Table</td><td>Service Point - MDM</td><td>Postal</td><td>No</td></tr>
<tr><td>Table (Demo)</td><td>Service Point Facility - MDM</td><td>Facility ID</td><td>No</td></tr>
<tr><td>Table (Demo)</td><td>Service Point Facility - MDM</td><td>Facility Relationship Type</td><td>No</td></tr>
</table>
</td></tr>
<tr><td>Criteria</td><td>

Criteria Name: TFORM
Source Type: Table (Demo)
Source Entity: Service Point Facility - MDM
Column/Type: Facility Relationship Type
Criteria Type: Include
Value: Commodity Distribution

</td></tr>
</table>

Dynamic Aggregation Processing for Consumption Analytics

Processing of dynamic aggregation for consumption analytics uses the standard approach to dynamic aggregation execution. Refer to **Dynamic Aggregation** in the *Business User Guide* for more information about dynamic aggregation execution and processing.

Oracle Utilities Network Management System

- Outage Care Integration
- The Big Picture of Outage Care Integration
- Customer Information Integration
- Interfacing Outage Calls
- Outage Inquiry
- Setting Up The System To Enable Outage Integration

Copyright © 2026, Oracle and/or its affiliates.

Outage Care Integration

The following section describes functionality provided for the integration between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.

The Big Picture of Outage Care Integration

Oracle Utilities Customer Care and Billing is the central repository for customer information; for example, name, address, phone number, etc. Oracle Utilities Network Management System is the central repository for outage information; for example, outage calls, affected supply nodes, expected restoration time, etc.

In an integrated environment, each system provides information to the other system so that they can operate together seamlessly.

- The outage system uses the set of current customers to determine and manage outages to minimize their impact.
- The outage system is informed of outages captured in Oracle Utilities Customer Care and Billing.
- Oracle Utilities Customer Care and Billing uses the current status of an outage at a given premise for customer service.

Customer Information Integration

The outage system needs information about current customers to determine and manage outages to minimize their impact. The current customer information must be made available in Oracle Utilities Network Management System. This can be done via data synchronization.

Depending on the specific edge application and version you are using, there are components of the integration that use Sync Request for the change data capture step. The following points highlight how to get more information:

- Refer to your products integration to Oracle Utilities Network Management System Implementation Guide to configure the Sync Request and change data capture functionality.
- Review the Sync Request Business Objects provided by your product for NMS integration.
- Refer to Data Synchronization for a high level understanding of the process.

NOT_SUPPORTED

Refer to The Big Picture of Sync Requests for more information about synchronizing data.

Interfacing Outage Calls

The following points describe the integration:

Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Customer Care and Billing is able to record trouble calls for a particular service point that exists in the system, as well as for an unknown service point, i.e. a fuzzy call. For a fuzzy call, the caller must provide either a street intersection, or a street segment.
- When an outage call is created and sent to the external system, an algorithm on the outage call business object is responsible for creating an outbound message that's sent to the external system. This is a real-time synchronous interface between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.
- Oracle Utilities Network Management System processes the Calls table and creates Incidents.

NOT_SUPPORTED

Refer to the *Oracle Utilities Customer Care and Billing - Network Management System Integration Implementation Guide* for information about outage call integration.

Outage Inquiry

Oracle Utilities Customer Care and Billing provides query transactions that can be used to make real-time synchronous calls to NMS and inquire on one of the following:

- Job History for a particular customer, service point, location or call identifier
- Call History for a particular customer, service point, location or call identifier
- Planned Outage Jobs for a particular service point

NOT_SUPPORTED

Refer to the *Oracle Utilities Customer Care and Billing - Network Management System Integration Implementation Guide* for information about outage query integration.

The product also supports the option of storing key Outage Event information in snapshot database tables for the purpose of reporting on active, planned and recently resolved outages. This information is made available by Oracle Utilities Network Management System as part of the integration flow that notifies impacted customers of outage event updates. The outage notification integration flow can be enabled to also share records for the latest updates of an outage event. Having outage data readily available within the application allows for more robust inquiries and insight analysis that does not overload the outage management system with real-time calls.

NOT_SUPPORTED

Refer to the *Oracle Utilities Notifications Hub Configuration Guide* for information about enabling outage data snapshot integration. When enabled, outage data is interfaced in using the **Outage Event Notifications** (C1-OutageEventNotifications) web service.

Copyright © 2026, Oracle and/or its affiliates.

You may review outage snapshot data in the following portals:

- Outage Event.
- Customer Outage Information.

Setting Up The System To Enable Outage Integration

The following section provides an overview of how to enable the integration between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System.

- External System Setup
- Define Characteristic Types
- Schema Constants - Feature Configuration
- NMS Outage Integration - Master Configuration
- Outage Notification Preferences

External System Setup

An external system must be setup in order integrate Oracle Utilities Network Management System. Define message senders for the following Outbound Message Types and associate them to the external system:

- NMS Outage Call (C1-NMSOUTCAL)
- NMS Job History Query (C1-NMSJOBHST)
- NMS Call History Query (C1-NMSCALHST)
- NMS Planned Outages Query (C1-NMSPLNOUT)

For synchronizing outage related notification preferences to NMS, define a message sender for the Outbound Message to Notify Edge Applications (C1-NTFOUTMSG) Outbound Message Type and associate it to the external system.

Once the external system is defined, configure the so the system knows which external system to use for outage queries.

Define Characteristic Types

The following characteristic types must be defined to facilitate Outage integration:

NOT_SUPPORTED

Refer to Defining Characteristic Types for information on how to set up characteristic types.

Outage Group Code Characteristic Type

These characteristics are used to describe the outage problem.

Copyright © 2026, Oracle and/or its affiliates.

- Create at least one pre-defined characteristic type
- For each characteristic type, define its list of valid values
- Include Service Task Type in the characteristic entity collection

> **Note**
>
> **Integration with Outage Management**. The outage codes must be defined in both Oracle Utilities Customer Care and Billing and the outage management system. Refer to your Oracle Utilities Network Management System documentation for information about defining the outage codes there.

Critical Customer Characteristic Type

These characteristics are used to associate a service point with one or more critical customer identifiers (e.g., medical, key account, government, etc.).

- Create a pre-defined characteristic type
- Define one or more critical customer predefined values
- Include Service Point in the characteristic entity collection
- Associate the characteristic and pre-defined values to the Critical Customer Types Characteristic Mapping section of the NMS Sync Integration Mater Configuration.

> **Note**
>
> **Integration with Outage Management**. The outage codes must be defined in both Oracle Utilities Customer Care and Billing and the outage management system. Refer to your Oracle Utilities Network Management System documentation for information about defining the outage codes there.

Contact Name Characteristic Type

This is used to link the contact name of the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Contact Name Characteristic Type feature option

Contact Number Characteristic Type

This is used to link the contact number of the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Contact Number Characteristic Type feature option

Call Identifier Characteristic Type

Copyright © 2026, Oracle and/or its affiliates.

This is used to link the call identifier supplied by the caller associated with an outage call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Identifier Characteristic Type feature option

Street Name Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Street Name Characteristic Type feature option

Cross Street Name Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Cross Street Name Characteristic Type feature option

Block Number Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call Block Number Characteristic Type feature option

City Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection
- Specify this characteristic type on theSchema Constants Feature Configuration using the Outage Call City Characteristic Type feature option

State Characteristic Type

This is used to link location information supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call.

- Create an ad-hoc characteristic type
- Include Service Task Type in the characteristic entity collection

Copyright © 2026, Oracle and/or its affiliates.

- Specify this characteristic type on the Schema Constants Feature Configuration using the Outage Call State Characteristic Type feature option

Schema Constants - Feature Configuration

Create a feature configuration with the type Schema Constants. Populate entries for all the options listed below.

> **Note**
>
> **Only one**. The system expects only one Schema Constants feature configuration to be defined.

Configure the options for your interaction with the outage system.

> **Note**
>
> Your implementation may define additional options types. You do this by adding new lookup values to the lookup field F1CN_OPT_TYP_FLG.

Option	Description
Outage Call Contact Name Characteristic Type	This is used to link the contact name of the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Contact Number Characteristic Type	This is used to link the contact number of the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Identifier Characteristic Type	This is used to link the call identifier supplied by the caller associated with an outage call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Street Name Characteristic Type	This is used to link the street name supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.

Copyright © 2026, Oracle and/or its affiliates.

Outage Call Cross Street Name Characteristic Type	This is used to link the cross street name supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call Block Number Characteristic Type	This is used to link the block number supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call City Characteristic Type	This is used to link the city supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.
Outage Call State Characteristic Type	This is used to link the state supplied by the caller associated with a fuzzy trouble call as a characteristic on the outage call. Refer to Defining Service Order Options for more information. Only one value is allowed for this option.

NMS Outage Integration - Master Configuration

The integration with Oracle Utilities Network Management System (NMS) requires the setup of master configuration data that controls the processing of master data synchronization and outage management operations.

Navigate to **Admin**, and then **General**, and then **Master Configuration**

Set up the following master configurations:

- **NMS Outage Integration Master Config**: This master configuration relates to the outage call and outage query integration with NMS.
- **NMS Sync Integration Master Config**: This master configuration relates to the service point based master data synchronization to NMS.

Refer to the embedded help for more details on how each section is configured.

Outage Notification Preferences

The system supports the ability to synchronize notification preferences that are owned by other edge applications. For example, during an outage event, Oracle Utilities Network Management System (NMS) is the system responsible for sending notifications to impacted customers that are subscribed or opted-in to receive notifications for these events. To support this functionality, notification preference statuses for outage related notification types that are updated under a customers account must be synchronized to NMS.

To configure your system to synchronize outage related notification preferences to NMS, perform the following steps:

Copyright © 2026, Oracle and/or its affiliates.

- Configure the outage notification types that are owned by NMS (e.g. Planned Outage, Unplanned Outage, Outage Started, Estimated Restoration Type, Outage Restored). Notification types can be configured as template based subscription, or template based parent/individual push. Refer to Setting Up Notification Preference Options for more information.
- Configure the **Notification Preferences** master configuration. Under the master configuration's Edge Application Notification section, add the notification types along with the edge application (Network Management System) and External System/Outbound Message Type for each entry. Refer to the embedded help for more details.

Once configured, the **Notification Preference Sync Initial Load (C1–NCPIL)** batch process is used to create initial synchronization requests. This batch control creates initial synchronization requests for Notification Contact Preference data (Maintenance Object C1–NT-PREF). The maintenance object and sync business object (C1-NtfContactPreferenceSync) are plugged in as parameters to the batch job.

For ongoing synchronization requests, the **Sync Request Monitor batch control (F1-SYNRQ)** is responsible for monitoring synchronization requests in the Pending state and transitions them to the Determine if Sync Needed state. Use the following parameters when running the F1-SYNRQ batch job:

- Maintenance Object: F1-SYNC REQ
-
- Restrict By Business Object: C1-NtfContactPreferenceSync
- Restrict By Status Code: PENDING

NOT_SUPPORTED

Refer to the *Oracle Utilities Customer Care and Billing - Network Management System Integration Implementation Guide* for information about outage notification integration.

Oracle Utilities Customer Program Cloud Service

The Oracle Utilities Customer Cloud Service Integration to Oracle Utilities Customer Program Cloud Service provides the following functionality:

- Initial and incremental extract of customer data including persons, service points, and service agreements from Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter or Oracle Utilities Customer Care and Billing).
- Initial and incremental import of customer data into Oracle Utilities Customer Program Management Cloud Service.

This section describes the customer data extract functionality within Oracle Utilities Customer Care and Billing (or Oracle Utilities Customer to Meter or Oracle Utilities Customer Care and Billing) used with this integration.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This integration is only used if Oracle Utilities Customer Cloud Service and Oracle Utilities Customer Program Management Cloud Service are deployed in separate environments.

- Customer Data Extract

Customer Data Extract

Customer information including persons, service points and service agreements in Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter or Oracle Utilities Customer Care and Billing) can be extracted and sent to Oracle Utilities Customer Program Management Cloud Service (or Oracle Utilities Customer Program Management) for use in enrolling customers in demand response programs.

- An initial extract is used to duplicate customer information data as of a given date in Oracle Utilities Customer Cloud Service in Oracle Utilities Customer Program Management Cloud Service. This is typically done during initial set up of the Digital Asset Cloud Service implementation.
- Following the initial extract, audit algorithms track new data and changes to existing data and create data synchronization requests, which can be used to create incremental extract files via batch processing.

Extracted data is sent to Customer Program Management in a delimited file, which is uploaded into the system via batch processing and data synchronization. See **Importing Customer-Related Data from a Customer Information System** in the *Oracle Utilities Customer Program Management Cloud Service User Guides* for more information about the import process.

Extract Process Overview

This section provide an overview of the extract process and outlines the various system objects used for each type of data (contacts, service points, and service agreements). The overall process is as follows:

1. An initial "extract" batch process is used to extract customer data into files that can be imported into Digital Asset Cloud Service.
2. Audit algorithms on the Contact, Service Point, and Service Agreement maintenance objects create synchronization requests.
3. Sync requests are processed using the specific batch processes (a separate batch control is provided for contacts, service points, and service agreements) to create incremental extracts.

Initial Load vs. Incremental Downloads

Customer data can be extracted in one of two modes: Initial or Incremental.

Initial extracts are used when setting up the system to export contact, service point, and service agreement data from Oracle Utilities Customer Cloud Service.

Incremental extracts are used to update customer data in Customer Program Management Cloud Service as customer data changes in Oracle Utilities Customer Cloud Service.

Copyright © 2026, Oracle and/or its affiliates.

Initial Extract Batch Processes

Extract files are created using object-specific "extract" batch processes. See Extract Process Object Summary below for the specific "extract" batch process used for each type of customer data. These are used only for the initial extract.

These processes use the following parameters:

- **File Name**: The name of the extract file. File names can use wildcards (Required)
- **Flat Path**: The file location where the extract file is created. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* for more information. (Required)
- **File Format**: The file format used for extract file. This should be FIXD. Each extract type has its own format.
- **Low SP Id Override Value**: A service point ID used to define the low value in a a range of IDs to limit the data extracted.
- **High SP Id Override Value**: A service point ID used to define the high value in a a range of IDs to limit the data extracted.

Use the **Batch Control** portal to learn more about these and other parameters used by these batch processes.

Audit Algorithms and Sync Request Business Objects

Sync requests are created by audit algorithms, and are based on data specific sync request business objects. See Extract Process Object Summary below for the specific audit algorithms and "Sync Request" business objects used for each type of customer data.

The table below outlines the stages of the sync request lifecycle. Use the **Business Object** portal to learn more about these business objects and the details of each lifecycle state.

State	Notes
Pending	Sync requests begin in this state. An Enter algorithms creates the initial snapshot of the Customer Program Management Cloud Service (CPMCS) data (contact, service point, or service agreement). The Sync Request Monitor (F1-SYNRQ) batch process is used to transition sync requests in this state to the Determine if Sync Needed state.
Determine if Sync Needed	An Enter algorithms creates the final snapshot of the CPMCS data. An Enter algorithms compares the initial and final snapshots. If there is no difference, the sync request moves to the Discarded state. Otherwise, the request moves to the Send Request state,
Discarded	Sync requests for which there is no different between the initial snapshot and final snapshot are sent to this state.
Send Request	An Enter algorithm creates a General Process record for the sync request. The request then moves to the Awaiting Acknowledgement state.

Copyright © 2026, Oracle and/or its affiliates.

State	Notes
Awaiting Acknowledgement	Requests remain in this state until the system receives an acknowledgement that CPMCS has received the data contained in the request.
Synchronized	Sync requests enter this state following the Awaiting Acknowledgement state when the system receives an acknowledgement that CPMCS has received the data contained in the request. If an error occurs, the request moves to the Synchronized with Error state.
Error	Sync requests enter this state if an error occurs while in the Awaiting Acknowledgement state. An Enter algorithm creates a To Do entry.
Canceled	Sync requests enter this state if canceled in either the Error or Synchronized with Error states.
Synchronized with Error	Sync requests enter this state if an error occurs while in the Synchronized state . An Enter algorithm creates a To Do entry.

Incremental Extract Batch Processes

Incremental extract files are created from General Process records using "Incremental Extract" batch processes. See below for the specific "Incremental Extract" batch processes used for each type of customer data. These should be run as needed for incremental updates.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Extract Process Object Summary

The table below lists the batch controls, algorithms, and business objects used by each type of customer data.

Object Type	Initial Extract Batch Process	Audit Algorithms	Sync Request Business Object	Incremental Extract Batch Process
Person	Migrate Person to DACS Contact (C1-DCTIL)	Sync Person as DACS Contact (C1_DAPERSYNC)	Sync CCB Person to DACS Contact (C1-DACSContactSyncRequest)	Contact Incremental Sync Batch Control (C1-DCTEX)
Service Point	Migrate CCB SP to DACS MSP (C1-DSPIL)	Sync Service Point as DACS Metered Service Point (C1_DASPSYNC)	Sync CCB SP to DACS Metered Service Point (C1-DACSMSPSyncRequest)	SP Incremental Sync Batch Control (C1-DSPEX)
Service Agreement	Migrates SA to DACS Metered Service (C1-DMSIL)	Sync Service Agreement as DACS Metered Service (C1_DASASYNC)	Sync CCB SA to DACS Metered Service (C1-DACSMSSyncRequest)	MS Incremental Sync Batch Control (C1-DMSEX)

Copyright © 2026, Oracle and/or its affiliates.

Acknowledgement Messages

When Customer Program Management Cloud Service processes incremental updates, it can send an acknowledgement back to Customer Cloud Service.

Refer to the Oracle Utilities Customer Cloud Service Integration to Oracle Utilities Customer Program Cloud Service documentation for more information about configuring Customer Program Management Cloud Service to send acknowledgements.

- Extracting Persons
- Extracting Service Points
- Extracting Service Agreements

Extracting Persons

This topic provides additional details related to extracting Persons from Oracle Utilities Customer Cloud Service for import into Oracle Utilities Customer Program Management Cloud Service.

> **Note**
>
> Persons are converted to Contacts in Oracle Utilities Customer Program Management Cloud Service.

Initial Extract Batch Process

Initial extract files for persons are created using the Migrate Person to DACS Contact (C1-DCTIL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run only for the initial extract.

Maintenance Object Audit Algorithm

The Sync Person as DACS Contact (C1_DAPERSYNC) audit algorithm tracks new person data and changes to existing person data and creates data synchronization requests, which are used to create incremental extract files.

Sync Request Business Object

Person sync requests are based on the Sync CCB Person to DACS Contact (C1-DACSContactSyncRequest) business object.

Incremental Extract Batch Process

Incremental person extract files are created from sync requests and General Process records using the Contact Incremental Sync Batch Control (C1-DCTEX) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for incremental updates.

Person (Contact) Extract File Format

Person (Contact) extract files use the following tab-separated values format:

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
External Reference Id	Optional	The contact's reference ID in the CIS application
Change Date Time	Optional	The date and time on which the contact was changed. **Applicable only for Incremental Uploads** Will be used to sequence processing in case there are multiple changes to the same contact.
Initial Load Sync Flag	Required	**A value that indicates the type of upload. Valid values are:** • **F1IL - Initial Load** • **F1IC - Incremental**
External Id	Required	The contact's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Contact External ID.
Contact Type	Required	The type of contact. When importing data from Customer to Meter or Customer Cloud Service this will be either Person or Business. If the provided contact type has not been defined as a Program Subscription Contact Type in Digital Asset Management / Digital Asset Cloud Service an extendable lookup is used to map this value to the Contact Type used in Digital Asset Management / Digital Asset Cloud Service.
Name	Required	The contact's name. For format for this is <FIRST_NAME> <LAST_NAME> (note the space between them.
Home Phone	Optional	A home phone number associated with person-type contacts.
Mobile Phone	Optional	A mobile phone number associated with person-type contacts.
Phone	Optional	A phone number associated with business-type contacts.
Email	Optional	An email address for the contact.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Life Support Sensitive Load	Required	A flag that indicates if the contact requires a Life Support Sensitive Load. Valid values are "true" and "false" An extendable lookup is used to map this value to the Life Support Sensitive Load flag used in Digital Asset Management / Digital Asset Cloud Service
Life Support Sensitive Load Description	Optional	An optional description for Life Support Sensitive Load.
Is Erased	Optional	A flag that indicates if the contact has been obfuscated (if supported) in the CIS application. Valid values are Y/N (Default) **Applicable only to Incremental Uploads** This should be set to "N" if Object Erasure is not supported in the CIS application.

Sample Record

Below is a sample file:

```
EXTREFCON11    2023-06-21T13:15:00-08:00    F1IC    EXTCONTACT1    P    John
Smith    (510)777-8888    (510)777-9090    (510)556-4556
johnsmith@gmail.com    false    None    N
```

Extracting Service Points

This topic provides additional details related to extracting service points from Oracle Utilities Customer Cloud Service for import into Oracle Utilities Customer Program Management Cloud Service.

Initial Extract Batch Process

Initial extract files for service points are created using the Migrate CCB SP to DACS MSP (C1-DSPIL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run only for the initial extract.

Maintenance Object Audit Algorithm

The Sync Service Point as DACS Metered Service Point (C1_DASPSYNC) audit algorithm tracks new service point data and changes to existing service point data and creates data synchronization requests, which are used to create incremental extract files.

Sync Request Business Object

Service point sync requests are based on the Sync CCB SP to DACS Metered Service Point (C1-DACSMSPSyncRequest) business object.

Copyright © 2026, Oracle and/or its affiliates.

Incremental Extract Batch Process

Incremental service point extract files are created from sync requests and General Process records using the SP Incremental Sync Batch Control (C1-DSPEX) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for incremental updates.

Service Point Extract File Format

Service point extract files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Reference Id	Optional	The metered service point's reference ID in the CIS application.
Change Date Time	Optional	The date and time on which the metered service point was changed. **Applicable only for Incremental Uploads** Will be used to sequence processing in case there are multiple changes to the same metered service point.
Initial Load Sync Flag	Required	**A value that indicates the type of upload. Valid values are:** • **F1IL - Initial Load** • **F1IC - Incremental**
External Id	Required	The metered service point's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Service Point External ID.
Metered SP Type	Required	The type of metered service point. When importing data from Customer to Meter or Customer Cloud Service this comes from the Service Point Type. If the provided metered service point type has not been defined as a Metered Service Point Type in Customer Program Management / Customer Program Management Cloud Service an extendable lookup is used to map this value to the Metered Service Point Type used in Customer Program Management / Customer Program Management Cloud Service.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
SP Source Status	Required	The metered service point's source status. Valid values are values for the Source Status (SP_SRC_STAT_FLG) lookup, and include: • **D1CN (Connected)** • **D1DC (Disconnected)**
Division	Optional	The metered service point's division. If the provided division has not been defined as a Division in Customer Program Management / Customer Program Management Cloud Service an extendable lookup is used to map this value to the Division used in Customer Program Management / Customer Program Management Cloud Service.
External Premise Id	Optional	The ID of the premise associated with the metered service point in the CIS application.
Time Zone	Required	The time zone in which the metered service point is located.
Country	Optional	The country in which the metered service point is located.
Postal	Optional	The metered service point's postal code.
Address 1	Optional	The metered service point's address.
Address 2	Optional	Additional details regarding the metered service point's address (if applicable).
Address 3	Optional	Additional details regarding the metered service point's address (if applicable).
Address 4	Optional	Additional details regarding the metered service point's address (if applicable).
City	Optional	The city in which the metered service point is located.
County	Optional	The county in which the metered service point is located.
State	Optional	The state in which the metered service point is located.
Longitude	Optional	The metered service point's longitude.
Latitude	Optional	The metered service point's latitude.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Life Support Sensitive Load Flag	Required	A flag that indicates if the metered service point requires a Life Support Sensitive Load. Valid values are "true" and "false" An extendable lookup is used to map this value to the Life Support Sensitive Load flag used in Customer Program Management / Customer Program Management Cloud Service
Life Support Sensitive Load Description	Optional	An optional description for Life Support Sensitive Load.
Status	Required	The current state of the metered service point. Valid values are "ACTIVE" and "INACTIVE".
Is OK to Enter?	Optional	
SP Warning	Optional	
SP Instructions	Optional	
SP Instructions Details	Optional	
Is Key	Optional	
Key Id	Optional	
Device Location	Optional	
Device Location Details	Optional	

Sample Record

Below is a sample file:

```
EXTREFMSP11    2023-06-21T14:01:00-08:00    F1IC    EXTMSP1    ZZ-EXTMSPTYPE
D1CN    ZZEDV    EXTPREM1    USPACIFIC    USA    94705    7610 Block Derby St
Addr2    Addr3    Addr4    Berkley    Alameda    CA    -81.697494
41.501120    false        ACTIVE
```

Extracting Service Agreements

This topic provides additional details related to extracting service agreements from Oracle Utilities Customer Cloud Service for import into Oracle Utilities Customer Program Management Cloud Service.

Initial Extract Batch Process

Initial extract files for service agreements are created using the Migrates SA to DACS Metered Service (C1-DMSIL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run only for the initial extract.

Maintenance Object Audit Algorithm

The Sync Service Agreement as DACS Metered Service (C1_DASASYNC) audit algorithm tracks new service agreement data and changes to existing service agreement data and creates data synchronization requests, which are used to create incremental extract files.

Copyright © 2026, Oracle and/or its affiliates.

Sync Request Business Object

Service agreement sync requests are based on the Sync CCB SA to DACS Metered Service (C1-DACSMSSyncRequest) business object.

Incremental Extract Batch Process

Incremental service agreements extract files are created from sync requests and General Process records using the MS Incremental Sync Batch Control (C1-DMSEX) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for incremental updates.

Service Agreement Extract File Format

Service agreement extract files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Reference Id	Optional	The metered service's reference ID in the CIS application.
Change Date Time	Optional	The date and time on which the metered service was changed. **Applicable only for Incremental Uploads** Will be used to sequence processing in case there are multiple changes to the same metered service point.
Initial Load Sync Flag	Required	**A value that indicates the type of upload. Valid values are:** • **F1IL - Initial Load** • **F1IC - Incremental**
External Id	Required	The metered service's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Service Agreement External ID.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Metered Service Type	Required	The type of metered service. When importing data from Customer to Meter or Customer Cloud Service this comes from the Service Agreement Type. If the provided metered service point type has not been defined as a Metered Service Type in Customer Program Management / Customer Program Management Cloud Service an extendable lookup is used to map this value to the Metered Service Type used in Customer Program Management / Customer Program Management Cloud Service.
External Contact ID	Optional	The ID of the contact associated with the metered service in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Contact External ID.
External Account ID	Optional	An ID used to identify the account associated with the metered service record in the CIS application.
Start Date Time	Required	The date and time on which the metered service record started.
End Date Time	Optional	The date and time on which the metered service record ended or will end (if applicable).
Division	Optional	The metered service's division. If the provided division has not been defined as a Division in Customer Program Management / Customer Program Management Cloud Service an extendable lookup is used to map this value to the Division used in Customer Program Management / Customer Program Management Cloud Service.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
SA Status	Optional	The metered service's status. If the provided status has not been defined as a Division in Customer Program Management / Customer Program Management Cloud Service an extendable lookup is used to map this value to a Status defined in Customer Program Management / Customer Program Management Cloud Service.
Rate Schedule List	Optional	A concatenated list of rate schedules and effective dates associated with the metered service. **Each row in this list is composed of the following:** • **Rate Schedule** • **Effective Date** Each element in the row will be delimited by comma (,). Each row in the list will be delimited by semi-colon (;). **For example:** **ERES-1,2022-01-01ERES-2,2021-01-01**
Metered SP List	Optional	A concatenated list of metered service points associated with the metered service. **Each row in this list is composed of the following:** • **External Metered SP Id (Required)** • **Start Date Time (Required)** • **End Date Time (Optional)** Each element in the row will be delimited by comma (,). Each row in the list will be delimited by semi-colon (;). **Example:** **6800401571,2022-12-13T14:30:00-05:00,1703304223,2022-12-13T14:30:00-05:00,**

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Is Erased	Optional	A flag that indicates if the customer has been obfuscated (if supported) in the CIS application. Valid values are Y/N (Default) **Applicable only to Incremental Uploads** This should be set to "N" if Object Erasure is not supported in the CIS application.

Sample Record

Below is a sample file:

```
EXTREFMS11    2023-06-15T13:02:00-08:00    F1IC    EXTMS1    CA!ZZMTRDST
EXTCONTACT1    EXTACCT1    2022-10-15T09:23:00-08:00
2023-06-14T13:45:00-08:00    ZZEDV    20    ERESRBOH,2022-11-09;
EXTMSP1,2022-11-09T13:14:55-08:00,2023-04-14T08:25:44-08:00;    Y
```

Asset Management System Integration

The following section describes functionality provided for the integration with an asset management system, such as Oracle Utilities Work and Asset Management (OUWAM).

- Interfacing Service Calls
- Setting Up The System To Enable Asset Management System Integration

Interfacing Service Calls

Oracle Utilities Customer Care and Billing is able to record service calls for a particular service point in the system, as well as for an unknown service point, i.e. a fuzzy call. Service call information is sent to the asset management system that will process any work that may be required due to the service call.

When recording a service call, the system creates a service call process flow and assigns it a service call communication type. The process flow type and service call communication type are determined from the action method with Action Method Role of **Service Call**. See Action Method for more information about setting up action methods. Service call communication type defines the rules and controls for creating service call communication. It also defines the service call communication business object.

The system is delivered with the following objects that are used to record a service call:

- Process Flow Type **C1-SERVICECALL**
- Service Call Communication Type business object **C1-ServiceCallCommTaskType**
- Service Call Communication business object **C1-ServiceCallCommunication**

Refer to Service Calls for more details.

Copyright © 2026, Oracle and/or its affiliates.

Setting Up The System To Enable Asset Management System Integration

The following section describes how to enable the integration between Oracle Utilities Customer Care and Billing and your asset management system.

- External System Setup
- WAM Integration Master Configuration

External System Setup

An external system must be setup to integrate with Oracle Utilities Work and Asset Management (WAM).

For sending service call information to WAM, define message senders for the following outbound message types:

- **New Service Call Communication Request Outbound Message (C1-SVCCALNEW)**. This is used to send new service call information.
- **Update Service Call Communication Request Outbound Message (C1-SVCCALUPD)**. This is used to send service call comments and service call cancel request.

For retrieving service call information from WAM, define message senders for the following outbound message types:

- **Service Call Search (C1-SVCCLSRCH)**. This is used by the service call search panel to retrieve service calls from the external system.
- **Service Call Details (C1-SVCCALDTL)**. This is used in the service call details pop up window to retrieve details of a service call from external system.
- **Retrieve Work Documents (C1-RETWRKDOC)**. This is used in the service call details pop up window to retrieve work documents related to a service call.

Once the external system is defined, configure the WAM Integration Master Configuration so the system knows which external system and outbound message types to use.

WAM Integration Master Configuration

The integration with Oracle Utilities Work and Asset Management (WAM) requires the setup of master configuration data that controls the processing of service call information.

To maintain master configuration, open **Admin**, and then **General**, and then **Master Configuration**

Set up the **CC&B WAM Integration** master configuration, which defines the external system and outbound message types to send and receive service call information. It also defines the URL of the WAM environment. This is used by the service call details pop-up window to render the service call hyperlink. The hyperlink is available if the user has access to the **Go To WAM (C1-GOTOWAM)** application service. Note that the login user to WAM should have access to the service call transaction. This authorization needs to be done in WAM.

Oracle Customer Experience for Utilities

This section provides configuration and implementation information used when Customer Cloud Service is used with Customer Experience for Utilities (CX4U). This includes:

Copyright © 2026, Oracle and/or its affiliates.

- Person Export: Batch controls and master configuration used to support export of person data for use with Customer Experience for Utilities.
- Person Export for Customer Experience for Utilities

Person Export for Customer Experience for Utilities

When integrating with Customer Experience for Utilities, all person records for individuals and business must be exported from Customer Cloud Service so they can be imported into, and synchronized between, all related Oracle Fusion systems. This involves defining mapping person and business values to corresponding values in the Oracle Fusion systems via a master configuration, and a pair of batch processes to export and synchronize the data.

Master Configuration

The CX4U Person Sync Mapping master configuration maps Customer Cloud Service person and business related values to their corresponding Fusion values for the purpose of an initial data export via a batch process. This includes the following:

- Phone Country Code
- Communication Types
- Person Characteristics
- Business Characteristics

On this master configuration, "internal" values are those used in Customer Cloud Service, and "external" values are those used in related Oracle Fusion systems.

Batch Processes

Two batch processes are used to export and synchronize person data.

- The **CX4U Export Person Information** (C1-PEREX) batch process exports person records into multiple files that can be used to perform an initial import of person information into Oracle Fusion to support Oracle Customer Experience for Utilities.
- After the initial import of records into Oracle Fusion, the **CX4U Update Person External ID** (C1-PERSY) batch process updates the External ID characteristics for Person records that exist in Oracle Fusion to reference their Oracle Fusion party number.
  Note: The **CX4U Update Person External ID** (C1-PERSY) batch process uses the CX4U Data Synchronization (CX4U) External System and the FA Person Synchronization (C1FsnPerSync) Message Sender to communicate with the Fusion systems to retrieve the Fusion party number. Refer to the External Systems and Message Sender sections for more information about these components.

Refer to the **Detailed Descriptions** of the CX4U Export Person Information (C1-PEREX) and CX4U Update Person External ID (C1-PERSY) batch controls for more information about parameters used with these batch processes.

Oracle Utilities Data Intelligence

- Defining Standard Measurement Identifiers

Defining Standard Measurement Identifiers

Copyright © 2026, Oracle and/or its affiliates.

Oracle Utilities Data Intelligence (OUDI) can perform data aggregation and enrichment on Advanced Metering Infrastructure (AMI) data extracted from Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service. Data is extracted using Measurement Data Extract.

Enabling the extract process requires defining the data to extract which involves defining the "standard measurement identifiers" used by Data Intelligence in Meter Solution Cloud Service and Customer Cloud Service, and mapping those identifiers to specific combinations of Units of Measure (UOM), Time of Use (TOU), and Service Quantity Identifiers (SQI) for specific service types.

Extendable Lookups

The Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup is used to define standard measurement identifiers. Values for this extendable lookup include the following:

- **Standard Measurement Identifier**: A code for the identifier
- **Standard Service Type**: The basic or standard service type for the identifier, from the Standard Service Type (STD_SVC_TYPE_FLG) lookup.
- **Status**: The current status of the identifier
- **Description**: The label used for the identifier in the Meter Solution Cloud Service and Customer Cloud Service user interface.

As an example, the standard measurement identifier for consumed electric kilowatt hours is defined as follows:

- **Standard Measurement Identifier**: E-KWH-CONS
- **Standard Service Type**: Electric
- **Status**: Active
- **Description**: Electric kWh - Consumed

Master Configurations

A set of "Measurement Identification Configuration" master configurations are used to map relationships between "standard measurement identifiers" defined in the Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup and specific UOM/TOU/SQIs combinations per service type. There are three of these master configurations, one for each of the Standard Service Types used in the extendable lookup. These are:

- Measurement Identification - Electric Configuration (D1-MsrmtIdentifyElectricConfig)
- Measurement Identification - Gas Configuration (D1-MsrmtIdentifyGasConfig)
- Measurement Identification - Water Configuration (D1-MsrmtIdentifyWaterConfig)

Mappings in these master configurations are defined by the following:

- **Service Type**: The service type associated with a channel (see Service Types for more information)
- **Channel Identification**: Details regarding how to identify data for a channel, including
  - **Standard Measurement Identifier**: One or more standard measurement identifiers (from the Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup) for the channel

Copyright © 2026, Oracle and/or its affiliates.

- **Color Option**: The color used to display measurement data for this channel (from the Color Option (F1-Color) extendable lookup)
- **Measurement Value Identification**: One more UOM/TOU/SQI combinations that defines what the channel measures
  - **Unit of Measure**: The unit of measure for the combination, such as kWh, kW, and so on
  - **Time of Use**: The time of use period for the combination, such as On-Peak, Off-Peak, and so on
  - **Service Quantity Identifier**: The service quantity identifier for the combination, such as "Generated", "Consumed", and so on

For example, the following configuration could be used to map three standard measurement identifiers (Electric kWh - Consumed, Electric kWh - Generated, and Electric kW) to the "Electric Services" service type and define the UOM/TOU/SQI combinations for each.

- **Service Type**: Electric Service
- **Channel Identification**:
  - **Standard Measurement Identifier**: Electric kWh - Consumed
  - **Color Option**: Green 010
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
    - **Service Quantity Identifier**: Consumed
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
- – **Standard Measurement Identifier**: Electric kWh - Generated
  - **Color Option**: Blue 010
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
    - **Service Quantity Identifier**: Generated
- – **Standard Measurement Identifier**: Electric Kilowatt
  - **Color Option**: Orange 010
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt

Message Formats

Oracle Utilities Meter Data Management supports three core transactional data imports: Initial Measurement Data, Device Event Data, and Usage Transactions. To see the appropriate format for each of these imports please see the following:

- **Initial Measurement Data**: Please see the IMD Seeder (D1-IMDSeeder) business object for the appropriate format
- **Device Events**: Please see the Device Event Seeder (D1-DeviceEventSeeder) business object for the appropriate format

Copyright © 2026, Oracle and/or its affiliates.

- **Usage Transactions**: Please see the Usage Transaction Seeder (D2-UsgTranSeeder) business object for the appropriate format*

Data Import - Message Driven Bean Configuration

- Overview
- JMS Configuration
- Message Driven Bean Configuration
- Notification Queue Configuration

Overview

This section describes the steps for configuring the Message Driven Bean (MDB) feature of Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway to listen to inbound JMS messages. This feature is used when importing IMDs and device events from head-end systems or when receiving Usage Transactions from CIS.

JMS Configuration

JMS configuration involves setting up JMS queues which will received inbound usage readings and device events. The JMS queues need to be created first on the application server where the OSB component is deployed. This server is referred to as remote server in the sections below. In the following section the JMS queue on the remote server is assumed to be created with the name **DestinationQueueWatch-CM**.

Note: The JMS changes described in the following sections are not persistent during patches or upgrades. They will need to be re-created after applying any patches or upgrades to Oracle Utilities Smart Grid Gateway. It is recommended to keep a backup of the `$SPLEBASE/splapp/config.xml` file.

Create a new JMS Module

Log in to the Oracle Utilities Smart Grid Gateway Weblogic console and create a JMS Module with an appropriate name. Specify the following values for this JMS module:

- **Name**: the name of JMS module. For example, JMSModule-CM
- **Target**: the name of the target server where the Oracle Utilities Smart Grid Gateway application is running. This should be specified as myserver.

Create a Foreign JMS Server

Create a Foreign JMS server under the JMS module created in the above step. Specify the following values for this foreign JMS server:

- **Name**: Name of the foreign server. For example, JMSFAServer-CM
- **Target**: This should be specified as myserver
- **JNDI Initial Context Factory**: This should be specified as weblogic.jndi.WLInitialContextFactory
- **JNDI Connection URL**: The URL of the server where OSB is deployed. For example: t3://osbserver:7001

Copyright © 2026, Oracle and/or its affiliates.

- **JNDI Properties Credential**: Password for the OSB server user
- **JNDI Properties**: The java.naming.security.principal additional property should be specified and set to the OSB server user. For example, java.naming.security.principal=weblogic

Create a Foreign Destination

Create a Foreign destination for each remote queue. Specify the following values for this foreign destination:

- **Name**: Name of foreign destination. For instance, DestinationQueue-CM
- **Local JNDI Name**: Local JNDI name for the foreign JMS queue. For example, ForeignDestinationQueue-CM
- **Remote JNDI Name**: JNDI name of the queue on the remote server. For example, DestinationQueueWatch-CM

Create a Remote Connection Factory

Create a remote connection factory for the foreign JMS server. Specify the following values for this remote connection factory:

- **Name**: Name of remote connection factory. For example, DestinationQueueConnectionFactory-CM
- **Local JNDI Name**: Local JNDI name for the Remote Connection Factory. For example, ForegnDestinationQueueConnectionFactory-CM
- **Remote JNDI Name**: JNDI name of the JMS Connection Factory on the remote server. For example, weblogic.jms.XAConnectionFactory

Message Driven Bean Configuration

Configuration of message driven beans (MDB) involved modifying the **ejb-jar.xml** and **weblogic-ejb-jar.xml** configuration files delivered with Oracle Utilities Smart Grid Gateway. It is recommended that instead of modifying these files directly you create "Customer Modification" (CM) versions of these files to make changes to these configuration files. This ensures that your modifications are not overwritten by future application patches.

The following section describes the changes required in the CM files for configuring the MDBs to read from the foreign JMS queues set up in the steps above. This requires creating the following files under $SPLEBASE/templates:

- cm_ejb-jar.xml.wls.jms_1.include
- cm_ejb-jar.xml.wls.jms_2.include
- cm_weblogic-ejb-jar.xml.jms.include.

Note: After making these changes the initialSetup script needs to be run and Oracle Utilities Smart Grid Gateway application needs to be redeployed. However the initialSetup script will overwrite the JMS configuration changes made in the steps above. So it is recommended to keep a backup of the $SPLEBASE/splapp/config.xml file before running this script.

Copyright © 2026, Oracle and/or its affiliates.

Changes to cm_ejb-jar.xml.wls.jms_1.include

The following is an an example of the `cm_ejb-jar.xml.wls.jms_1.include` file:

```
<message-driven>
   <description>MDB for DestinationQueue-CM</description>
   <display-name>DestinationQueueWatcher-CM</display-name>

      <ejb-name>DestinationQueueWatch-CM</ejb-name>

   <ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
   <messaging-type>javax.jms.MessageListener</messaging-type>
   <transaction-type>Bean</transaction-type>
   <message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
```

The values specified in the above file include the following:

- **ejb-name**: This is the name of the MDB.

Changes to cm_ejb-jar.xml.wls.jms_2.include

The following is an example of the `cm_ejb-jar.xml.wls.jms_2.include` file:

```
   <assembly-descriptor>
   <security-role>
   <role-name>cisusers</role-name>
   </security-role>
   <container-transaction>
   <method>

      <ejb-name>DestinationQueueWatch-CM</ejb-name>

   <method-name>onMessage</method-name>
   </method>
   <trans-attribute>NotSupported</trans-attribute>
   </container-transaction>
   </assembly-descriptor>
```

The values specified in the above file include the following:

- **ejb-name**: This is the name of the MDB

Changes to cm_ejb-jar.xml.wls.jms_2.include

The following is an example of the `cm_weblogic-ejb-jar.xml.jms.include` file:

```
   <weblogic-enterprise-bean>

    <ejb-name>DestinationQueueWatch-CM</ejb-name>

   <message-driven-descriptor>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>

 <destination-jndi-name>ForeignDestinationQueue-CM</destination-jndi-name>

 <connection-factory-jndi-name>ForeignConnectionFactory-CM</connection-
factory-jndi-name>

</message-driven-descriptor>
</weblogic-enterprise-bean>
```

The values specified in the above file include the following:

- **ejb-name**: This should be the name of the MDB as specified in ejb-jar.xml
- **destination-jndi-name**: This should be the JNDI name of the foreign destination as provided in JMS module ' Foreign server ' Foreign destination ' Local JNDI name.
- **connection-factory-jndi-name**: This should be the JNDI name of the connection factory as provided in JMS module ' Foreign server ' Remote Connection Factory ' Local JNDI name.

Notification Queue Configuration

Payload statistics and payload summary records must be submitted sequentially in order for them to be processed correctly. To prevent them from being processed at the same time, you should set the number of notification queue polling threads to 1. Follow these steps to configure the number of notification queue threads:

1. Log in to the WebLogic Server Administration Console.
2. Under Helpful Tools, click **Configure Applications.**
3. Click on **Webservices**.
4. Click on the NotificationQueue link. for the EJB that you want to configure.
5. Go to the Configuration tab.
6. In the Change Center, click **Lock & Edit.**
7. Specify the new value of polling threads in **Max Beans in Free Pool.**
8. Click **Save.**
9. Click **Release Configuration.**
10. Restart the OUAF WebLogic instance.

Creating Application Links in External Applications

When integrating Oracle Utilities applications with other external applications, you can create links from external applications to specific portals and pages within the Oracle Utilities applications (also known as "deep linking"). These links typically leverage Business Process Assistant (BPA) scripts that direct the user to the appropriate target portal or page for a specific entity such as an account or service point. Refer to Creating Application Links in External

Copyright © 2026, Oracle and/or its affiliates.

Applications in the *Application Framework Administrative User Guide* for more information about creating links.

Note: When creating application links, the use of the following parameters will help speed up launching the target page

- **initNav=false**: skips launching the default home page for the user
- **minSidebar=true** minimizes the sidebar for the user

Go to Portal Script

The Framework provides a generic script (F1-GoToPrtl) that can be used to navigate to the maintenance portal for a specified maintenance object and entity. Refer to Navigating to a Given Record's Maintenance Portal in the *Application Framework Administrative User Guide* for more information about using this script.

Go To CCB Script

Oracle Utilities Customer Care and Billing includes the Go To CCB script (C1-GoToCCB) that can be used to navigate directly to a specific page on Control Central. This script uses the following syntax:

```
script=C1-GoToCCB&FLAG=<FLAG_VALUE>&PKVALUE1=<ID>
```

Where:

- **<FLAG_VALUE>**: A value from the C1_TARGET_PORTAL_FLG lookup that specifies which Control Central page to open.
- **<ID>**: The primary key value for the entity (such as a Account, Person, or Premise) you wish to view.

Examples:

Script	Flag Value	ID	Example
C1-GoToCCB	FLAG=C1CC (Control Central Account)	Account ID	script=C1-GoToCCB&FLAG=C1CC&PKVALUE1={your Account ID}
C1-GoToCCB	FLAG=C1CP (Control Central Person)	Person ID	script=C1-GoToCCB&FLAG=C1CP&PKVALUE1={your Person ID}
C1-GoToCCB	FLAG=C1CR (Control Central Premise)	Premise ID	script=C1-GoToCCB&FLAG=C1CR&PKVALUE1={your Premise ID}

Full URL Example:

```
https://<ENV_BASE_URL>/ouaf/cis.jsp/?minSidebar=true&initNav=false&script=C1-GoToCCB&FLAG=C1CC&PKVALUE1={your Account ID}
```


Copyright © 2026, Oracle and/or its affiliates.

Go To MDM Script

Oracle Utilities Meter Data Management includes the Go To MDM script (D1-GoToMDM) that can be used to navigate directly to a specific 360 Degree View tab. This script uses the following syntax:

```
script=D1-GoToMDM&FLAG=<FLAG_VALUE>&PKVALUE1=<ID>
```

where:

- **<FLAG_VALUE>**: A value from the DRILL_ENTITY_FLG lookup that specifies which 360 Degree View page to open.
- **<ID>**: The primary key value for the entity (such as a Device, Measuring Component, or Service Point) you wish to view.

Examples:

Script	Flag Value	ID	Example
D1-GotoMDM	FLAG=D1DV (360 View for Device)	Device ID	script=D1-GotoMDM&FLAG=D1DV&PKVALUE1={your Device ID}
D1-GotoMDM	FLAG=D1MC (360 View for Measuring Component)	Measuring Component ID	script=D1-GotoMDM&FLAG=D1MC&PKVALUE1={your Measuring Component ID}
D1-GotoMDM	FLAG=D1SP (360 View for Service Point)	Service Point ID	script=D1-GotoMDM&FLAG=D1SP&PKVALUE1={your Service Point ID}

Full URL Example:

```
https://<ENV_BASE_URL>/ouaf/cis.jsp/?minSidebar=true&initNav=false&script=D1-GotoMDM&FLAG=D1DV&PKVALUE1={your Device ID}
```

Defining Communications Options

- Device Event Types
- Activity Types
- Communication Types
- Service Task Types

Device Event Types

- Understanding Device Event Types

Copyright © 2026, Oracle and/or its affiliates.

- Configuring Device Event Types

Understanding Device Event Types

Device event types define properties common to specific types of events.

Device event types represent different types of events that can take place relative to a device. Examples of device events include power outages, power restoration, tampering alerts, and other events.

Device event types can be defined by the following attributes:

- **Standard Event Name:** the "standard" name of the event type. Device vendors may have their own specific names for device events. Only a single active device event type may be mapped to a given standard name at any time.
- **Device Event Category:** a category (defined as an Extendable Lookup) used to group device event types.
- **Reporting Category:** a category used to group device event types for reporting purposes.
- **Activity Type:** the activity type for activities created for device events of this type.

Device Event Mapping

The first step to device event type configuration is defining the list of standard event names that will be processed by the system. This is done by populating the Standard Event Name extendable lookup (D1-StdEventNameLookup). More information about this extendable lookup can be found in About Device Events.

With the fully defined list of Standard Event Names each head-end specific event name (which is also called an external event name) that the system will receive from a given head-end should be mapped to a standard event name. This mapping is configured using a device event mapping extendable lookup. Each head-end system should have its own extendable lookup to define event name mapping in order to prevent possible conflicts between mappings. Each of these business objects should be defined as a child of the parent business object Device Event Mapping (D1-DeviceEventMappingLookup).

For example, head-end systems A and B might both use the same event name, such as the code "1", but this event might need to be mapped to "outage" for head-end system A but "tamper" for head-end system B.

The device event mapping extendable lookup business object is configurable. Each Oracle Utilities Smart Grid Gateway adapter includes a device event mapping lookup business object for the supported head-end system.

Lastly, each device event type is associated to a standard name. This means that as each device event that is received will go through the following mapping steps:

1. The head end specific event name is mapped to a standard event name using the head end specific device event mapping extendable lookup
2. The standard event name is mapped to a device event type
3. The device event type is used to select the appropriate device event business object

Device Event - Additional Processing

Certain device event business objects can be used to update master data as a result of device events received into the system. For example, when a utility's field worker arms a meter, the

Copyright © 2026, Oracle and/or its affiliates.

resulting device event can trigger an update to the device's arming status in the Oracle Utilities application. This processing is initiated by the Execute - Additional Processing system event

algorithm during the Additional Processing status of the business object's lifecycle.

The following base package business objects are configured to execute additional processing:

If a standard device event requires additional processing, the algorithms that execute the processing should be specified in the Additional Processing Algorithms list on the Standard Event Name extendable lookup for the event. The base package includes the additional processing algorithm Arm Meter (D1-ARM-METER).

Only a subset of base package device event business objects are configured to execute additional processing

Description	Business Object Name
Device Event Communication Response	D1-DeviceEventComResp
Standard Device Event	D1-StandardDeviceEvent

If additional processing is required for any business objects not listed above it can easily be added by configuring the algorithm Executer - Additional Processing System Event (D1-EXTADDPRC) in the Additional Processing status.

Reader Remarks

Reader remark types define properties common to specific types of reader remarks.

Reader remarks are a type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters. Reader remark types represent the different types of remarks that meter readers can record. Examples of reader remark types include evidence of tampering, broken seals, damaged meter, dog on premises, and other notices.

When creating a new device type there are the following options:

Name	Details	Business Object
Reader Remark Type	Provides the configuration for a reader remark type	D1-ReaderRemarkType

Field Activity Remarks

Field activity remark types define properties common to specific types of field activity remarks.

Field activity remarks are a type of device event used to capture and/or record specific events or circumstances encountered when a field worker is performing field work at a service point. Field activity remarks represent can represent situations that are found as well as actions that were taken in the field.

When creating a new device type there are the following options:

Name	Details	Business Object
Device Field Remark Type	Provides the configuration for a field activity remark type	D1-DeviceFieldRemarkType

Copyright © 2026, Oracle and/or its affiliates.

Configuring Device Event Types

This portal is used to display and maintain a Device Event Type.

Refer to Understanding Device Event Types for more information.

You can access the portal by selecting **Admin**, then **Communication**, then **Device Event Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Event Type List:** This zone lists all Device Event Type records. Broadcast a record to display the details of the selected record.
- **Device Event Type:** This zone provides information about the selected Device Event Type.

Activity Types

- Understanding Activity Types
- Configuring Activity Types
- Configuring Service Order Management Activity Types

Understanding Activity Types

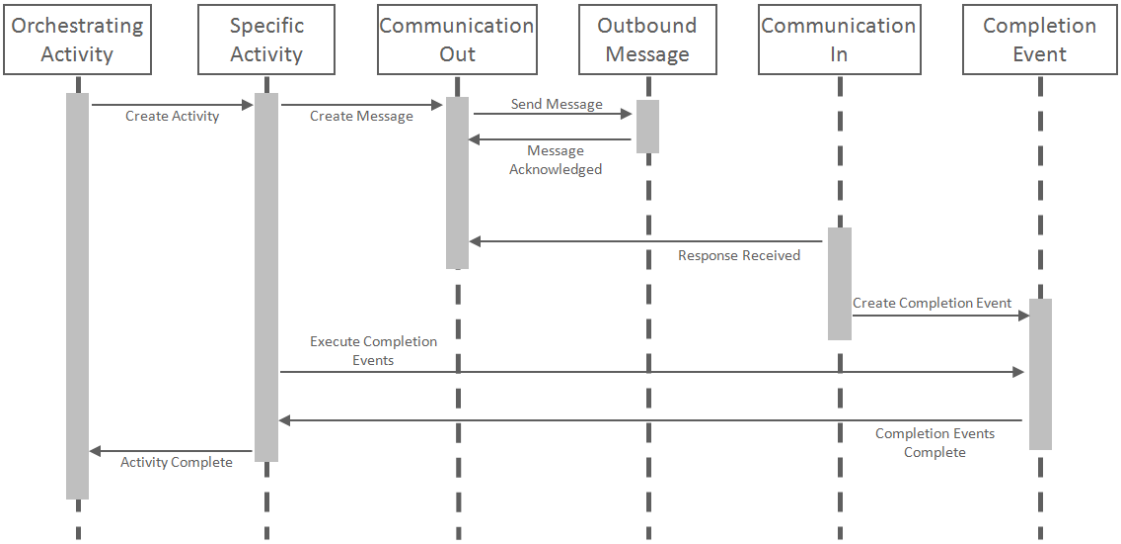
Activities are records of a communication or event related to a device, measuring component, service point or other entity in the system. Examples of activities include smart meter command request, field activities, meter read downloads (for manually read meters) or the combined event of a "last gasp" and "power up" message sent by devices when they detects they experience an outage.

Each type of activity is assigned to an activity type category, please refer to About Activities for more information about the activity type categories supported by base product.

Activities Orchestrate Communication

Certain types of activities such as request orchestrators, field activities, and command requests coordinate a large number of child transactions that represent the communication to and from an external application. The following diagram depicts a two-way communication with an external system:

Copyright © 2026, Oracle and/or its affiliates.

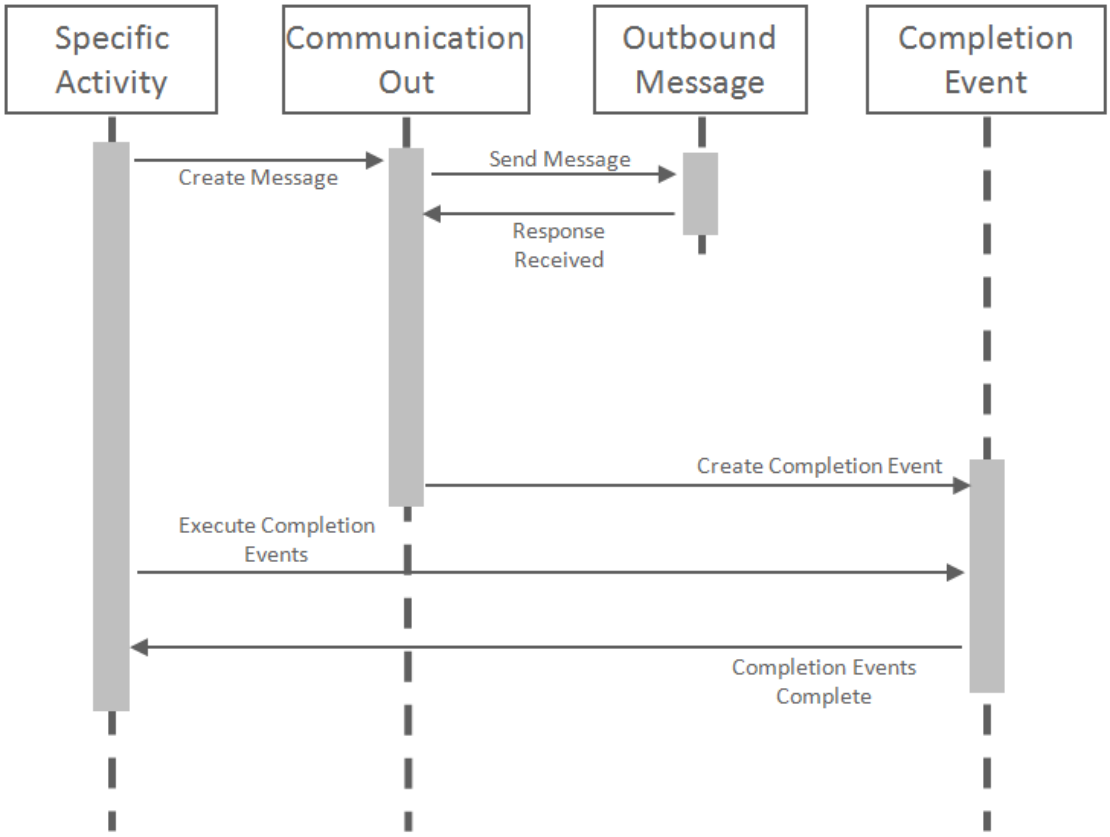


Each object in the sequence diagram has a distinct set of duties within the context of the communication:

- Orchestrating Activity: controls the overall intent of the communication. For example it may "Start Service" which would include initiating one-to-many specific activities to install the meter and begin the flow of the metered commodity to the service point.
- Specific Activity: can be initiated from an orchestrating activity or directly. These activities represent a single task to be carried out such as installing a meter or a remote disconnect smart meter command.
- Communication Out: orchestrates the communication to the external application and provides robust handling for any errors that might occur during that communication.
- Outbound Message: represents the message payload sent to the external system and the synchronous response.
- Communication In: orchestrates the handling of an asynchronous or unsolicited response from an external system.
- Completion Event: carries out the results of the communication. For example, in the case of a remote connect it would create the appropriate on off history entry for the device's installation event.

The following diagram involves many of the same objects but instead represents a one-way communication with an external system. All objects maintain the same duties described above with the addition of the communication out handling the synchronous response which contains the result of the message.

Copyright © 2026, Oracle and/or its affiliates.



Important Activity Type System Events

The activity type supports several business object algorithm system events that relate to calculating the consumption for measuring components of that type:

- **Customer-Device Compatibility**: receives information about the activity being analyzed and returns an indication of whether the customer and device are compatible. This system event is primarily used in service order management. See algorithm type Ensure Device-Usage Calculation Group Compatibility (D2-ENSDVCCMP) as an example.
- **Override Device / Task**: receives information about the activity such as whether the device is installed at a service point, whether the service point is connected, where it is disconnected (if applicable), and the installation event status override. Based on those inputs it should determine the output device configuration type as well as the field activity business object and field task type. This system event is primarily used in service order management. See algorithm type Evaluate Smart Meter Opt-Out for Device Installation (D2-EVSMOPDV) as an example.

Service Order Activity and Service Order Field Activity Types and Divisions

Service order activity types and service order field activity types can specify a Division, to represent the specific operating company associated with activities of that type. If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", users can only access service order activities or service order field activities with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the**Miscellaneous** tab of the **User** portal.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In Oracle Utilities Customer to Meter, the **Control By CIS Division** setting on the **Main** tab on the **Installation Options** portal is used to restrict data access based on divisions.

See Understanding Divisions for more information about divisions.

Activity Type Maintenance Category

Activity types can be assigned an Activity Type Maintenance Category to specify that they should be maintained using the SOM Activity Type portal rather than the standard Activity Type portal. This option is used with the following categories of activity types:

- Field Activity Types
- Non-Dispatchable Activity Types
- Orchestration Maintenance Types
- Request Orchestration Types

Assigning activity types to the "Service Order Management Activity Type" Activity Type Maintenance Category is done via the Activity Type Maintenance Category business object option on the appropriate activity type business objects.

Configuring Activity Types

You use the **Activity Type** portal is to view and maintain activity types.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured activity types as well activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the

Copyright © 2026, Oracle and/or its affiliates.

algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

Configuring Service Order Management Activity Types

You use the **SOM Activity Type** portal to view and maintain activity types whose business object has the Activity Type Maintenance Category business object option set to D1SO (Service Order Management Activity Type).

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order management activity types as well activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type

Copyright © 2026, Oracle and/or its affiliates.

- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

Communication Types

- Understanding Communication Types
- Configuring Communication Types

Understanding Communication Types

Communication types define properties common to a specific type of communication. This single admin object covers both communication in and communication out types.

Communication types include types of communications between an application and an external system such as a head-end system or the work management system. These are outbound communications such as smart meter command requests or field activities as well as inbound communications such as a message response that indicates the results of a request.

Communication types typically provide configuration around exception handling:

- **To Do Type and Role**: identifies the To Do to be created when an error is encountered
- **Retry Frequency**: defines how often the communication should be retried. As many errors can be due to connectivity the ability to retry provides automatic resolution.

Copyright © 2026, Oracle and/or its affiliates.

- **Maximum Retries**: provides an upper limit to the number of times a communication can be re-attempted
- Each communication type can have its own unique set of fields, please refer to the embedded help for that communication type for more details on those fields.

Configuring Communication Types

This portal is used to display and maintain a Communication Type.

Refer to Understanding Communication Types for more information.

You can access the portal by selecting **Admin**, then **Communication**, then **Communication Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Communication Type List:** This zone works differently than the typical zone that list types in that it displays both those communication types that have been configured as well as those communication types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Communication Type:** This zone provides information about the selected Communication Type

Service Task Types

- Understanding Service Task Types
- Configuring Service Task Types

Understanding Service Task Types

Service tasks types define properties common to specific types of service tasks.

Service task types represent types of tasks that can be performed by users of other Oracle Utilities applications, such Oracle Utilities Customer Self Service or Oracle Utilities Network Management System. Examples of service tasks include self service meter reads, in which users enter their own meter reads via the Customer Self Service application, and service issue monitor types used when determining if service investigation is needed for a service point.

When creating a new record, the following options are available:

Name	Details	Business Object
Service Issue Monitor Type	The service issue monitor type defines the parameters for when a service investigative order should be created. It can also optionally define the completion parameters for that indicate the investigative order was successful in its purpose.	D1-ServiceIssueMonitorType
Self-Service Rate Compare Scenario Request Type	A simple description of a rate compare scenario request type.	D2-RateCompareScenarioRqstType

Copyright © 2026, Oracle and/or its affiliates.

Self-Service Meter Read Task Type	Provides the default settings for meter reads created through self service.	D2-SSMeterReadTaskType

Configuring Service Task Types

This portal is used to display and maintain a Service Task Type.

Refer to Understanding Service Task Types for more information.

You can access the portal by selecting **Admin**, then **Customer**, then **Service Task Type**

The following zones may appear as part of the portal's **Main** tab page:

- **Service Task Type List:** This zone lists all Service Task Type records. Broadcast a record to display the details of the selected record.
- **Service Task Type:** This zone provides information about the selected Service Task Type.

Data Extracts

This section describes functionality that support extract and export of data for use in external applications and systems.

- Configurable Consumption Extracts
- Maintenance Object Data Export
- Service Point Consumption Extracts
- Consumption Extract Type
- Analytics Configuration
- Migrating Data Extract Processing from Legacy Measurement Processing to Direct Measurement Processing

Configurable Consumption Extracts

Configurable Consumption Extracts are measurement data extracts for a set of usage subscriptions optionally related to a specific market participant.

Configuration options for these extracts include:

- Type of Market Participant, based on Service Point Market Participant or Usage Subscription Market Participant tables and market relationship
- Usage Subscriptions Types associated with the market participant (typically a customer information system receiving bill determinants)
- Unique combinations of unit of measure (UOM), time of use (TOU), and service quantity identifier (SQI) associated with the usage subscriptions
- Specific identifiers used to identify the usage subscriptions, service points, and devices associated with extracted measurement data
- Date and time parameters and frequency of extract

Copyright © 2026, Oracle and/or its affiliates.

- Extract file naming and compression options

Consumption extracts are exported to a file in a specified file location, either on an application server (in an on-premises implementation) or an Oracle Cloud Infrastructure Object Storage location.

Consumption extracts are created via Consumption Extract Requests. See Consumption Extract Requests in the *Business User Guide*.

This section describes the configuration needed for these types of consumption extracts.

Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

Consumption Extract Types

Configuration for consumption extracts are defined via a Consumption Extract Type. See Consumption Extract Type in the *Administrative User Guide* for more information about creating these records.

Configurable consumption extract types should be created using the Extract US Consumption to File (D1-ExtractUSConsumptionToFile) business object. Refer to the embedded help on this business object for more details about the parameters defined for consumption extract types.

Extract US Consumption to File consumption extract types define the following parameters:

- The market participants for which measurement data will be extracted. You define market participants based on their type (Service Point Market Participant or Usage Subscription Market Participant) and market relationship type (retailer, distributor, supplier, and so on), and a specific market participant. Note: If Market Participant Specific is set to "No", the market participant used is based on the Usage Recipient on the Usage Subscription or its Usage Subscription Type.
- The consumption type (interval or scalar)
- One or more Usage Subscription Types and a related unique UOM/TOU/SQI combination
- The specific ID types to include in the extract file for the Usage Subscription, Service Point, child Service Point (if applicable), and Device
- Execution parameters, including:
  - The date and time through which data has been extracted for this consumption extract type
  - A Request Type. Request types used with consumption requests should be created using the Consumption Extract Request Type (D2-IntervalDataExtRepoType) business object, and should reference the Consumption Extract (D2-IntervalDataExtRepository) business object in the **Request Business Object** field. (see The Big Picture of Requests in the *Application Framework Administrative User Guide*)
  - The frequency of extracts for this type (adhoc or daily)
  - Batch controls used for the extract process (see **Batch Controls** below)
- File parameters, including:
  - The file location where extract files will be saved (see External File Storage in the *Application Framework Administrative User Guide*)

Copyright © 2026, Oracle and/or its affiliates.

- An optional child folder in the file location
- File naming options (including an optional prefix and date/time stamp)
- Whether or not to compress the extract file (file compression is performed using gzip)

Batch Controls

The data extract process and creation of export files is performed by the following batch controls:

- **Extract Current US Intervals to File** (D1-ECUSI): Used for daily extracts of interval consumption. This process requires that Interval Snapshot Mapping data be generated for any of the interval sizes being extracted. This aides in the efficiency of transposing the interval data into daily interval columns. (see Interval Snapshot Mapping for more information).
- **Extract Replacement US Intervals to File** (D1-ERUSI): Used for extracts of replacement historical interval consumption. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table (see **Initial Measurement Subsequent Correction Change Data Capture** below).
- **Extract Current US Scalar Reads to File** (D1-ECUSS): Used for daily extracts of scalar consumption.
- **Extract Replacement US Scalar Reads to File** (D1-ERUSI): Used for extracts of replacement scalar interval consumption. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table (see **Initial Measurement Subsequent Correction Change Data Capture** below).

These batch controls are run for a specific Consumption Extract Type and date range. Batch processing is threaded by on Usage Subscription ID.

Refer to Consumption Extract Requests - Batch Controls in the **Defining DataConnect Options** section of the **Integrations** chapter for general information about consumption extract batch processing,

Initial Measurement Subsequent Correction Change Data Capture

The following algorithms maintain measuring component date/times based on the latest measurement date time for the measuring component. These algorithms also check for a Consumption Extract Type that applies to the current initial measurement's measuring component and identifies whether the initial measurement is for a date/time that has already been extracted.

Algorithm	Description	Consumption Type
D1-UPD-DTMC	Update Latest Measurement Date/Time on MC with Consumption Sync	Interval
D1-UDTSCMCRE	Update Latest Measurement Date/Time on Scalar MC with Consumption Sync	Scalar
D1-UPD-DTMCF	Update Latest Date/Time on MC from a Future Measurement	Interval
D1-UPDDTSCMC	Update Latest Date/Time on Scalar Measuring Component	Scalar

If an initial measurement is for a date/time that has already been extracted, these algorithms create Change Data Capture records in the General Process table that capture the batch code, batch run number, and the initial measurement's ID. The "Replacement" batch processes run off of these records.

Copyright © 2026, Oracle and/or its affiliates.

Note that the following parameter must be set to “true” on the above algorithms to enable historical change data capture:

- Capture CET Historical Measurement Changes: true
- Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

Data Extract File Format - Interval

The interval extract process produces JavaScript Object Notation (JSON) files with the extract data described in the following table.

Note that interval extract files can store up to a maximum of 300 intervals. This means that export files can store approximately:

- 12.5 days worth of hourly data
- 3 days worth of 15 minute data
- 1 day worth of 5 minute data

Field	Sample Value	Comments
usId	995647003500	Usage Subscription ID defined by the Consumption Extract Type
usType	E-RES	Usage Subscription type
pSpId	458081	Parent Service Point ID, Null when there is no parent service point.
spId	458081_001	Child Service Point ID This is the Service Point ID defined by the Consumption Extract Type
dvcId	SN823287793219	Device ID defined by the Consumption Extract Type
uomTouSqi	KWH//	UOM / TOU / SQI, separated by forward slashes ("/")
tz	US/Eastern	Time zone of the service point. This is the time zone name from the CI_TIME_ZONE table not the time zone code.
intPerDay	288	Number of Intervals for the day, will be 1 hour less or 1 hour more of intervals for DST transitions
intSize	300	Length of each interval in seconds
mktPart	DIRECT_ENERGY	Market Participant ID of the market participant (if applicable)
stDttm	2020-01-01T00:00:00-05:00	Start of the day, similar to initial measurement, this represents the start of the first interval.
q1	1.325	Measurement Value for the first interval, the measurement date/time for this interval would be the stDttm plus the intSize
c1	301000	Measurement Condition for the first interval. Null when "Regular" (501000)

Copyright © 2026, Oracle and/or its affiliates.

Field	Sample Value	Comments
q2	1.230	
c2	301000	
...	q/c 3 through 299 here	
q300	.9134	
c300	30100	

Example:

```
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "intPerDay":"288", "intSize":"300",
"mktPart":"DIRECT_ENERGY",
 "stDttm":"2020-01-01T00:00:00-05:00", "q1":"1.325", "c1":"301000",
"q2":"1.325",
 "c2":"301000" "q3":"1.325", "c3":"301000",
<intervening intervals>, "q300":"1.325", "c300":"301000"}
```

Data Extract File Format - Scalar

The scalar extract process produces JavaScript Object Notation (JSON) files with the extract data described in the following table.

Name	Sample Value	Comments
c	301000	Measurement Condition for the scalar reading Null when "Regular" (501000)
dvcId	SN823287793219	Device ID defined by the Consumption Extract Type
enDttm	2020-01-02T00:00:00-05:00	End date/time of the reading
mktPart	DIRECT_ENERGY	Market Participant ID of the market participant (if applicable)
pSpId	458081	
q	1.325	Measurement Value for scalar reading
r	678645	Dial reading of the meter
spId	458081_001	Service Point ID defined by the Consumption Extract Type
stDttm	2020-01-01T00:00:00-05:00	Start date/time of the reading (the End Date/Time of the previous reading)
tz	US/Eastern	Time zone of the service point. This is the time zone name from the CI_TIME_ZONE table not the time zone code.
uomTouSqi	KWH//	UOM / TOU / SQI, separated by forward slashes ("/")
usId	995647003500	Usage Subscription ID defined by the Consumption Extract Type

Copyright © 2026, Oracle and/or its affiliates.

Name	Sample Value	Comments
usType	E-RES	Usage subscription type

Example (3 records):

```
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
```

Extending the Extract Processing

Some implementations may wish to extend and/or customize the extract process based on specific requirements.

Extending List of IDs To Use

Consumption Extract Types define specific types of IDs to use for Usage Subscriptions, Service Points, child Service Points (if applicable), and Devices. If an implementation wishes to extend the types of IDs available, they need to add appropriate values to the lookup flags used by each of ID type. The table below lists the lookup codes used by the "IDs to Use" fields.

ID Type	Lookup Flag
Usage Subscription ID	Usage Subscription Identifier Type (US_ID_TYPE_FLG)
Service Point ID (used for both parent and child service points)	Service Point Identifier Type (SP_ID_TYPE_FLG)
Device ID	Device Identifier Type (DVC_ID_TYPE_FLG)

Processing Records and Creating Output Files

The extract process leverages the generic Plug-in Driven Background Processes functionality but works in a slightly different manner. Instead of using the Select Records algorithm to retrieve the IDs of the individual records to be processed, that algorithm returns a range of

Copyright © 2026, Oracle and/or its affiliates.

Usage Subscription IDs to be extracted and the Process Records algorithm performs the detailed query to extract the range of Usage Subscription IDs.

This provides two benefits:

1. There is no limitation on the length of the query in the Process Records algorithm.
2. Retrieving data for many Usage Subscriptions at a single time is significantly more efficient.

This means that any required customization can be achieved by creating a new custom (CM) batch control and duplicating the Process Records Groovy script. Changes can be made within the custom Groovy script to the extraction SQL to retrieve additional information or provide additional selection criteria. If any changes are needed to the file format that can be done within the record processing logic that populates the file.

> **Note**
>
> The Process Records scripts include a number of steps each containing specific processing logic and SQL based on the configuration of the Market Participants on the Consumption Extract Type as well as whether or not the specified UOMs measure peak quantities or not. Be sure to edit each of these steps as appropriate when creating customized versions of these scripts.

Maintenance Object Data Export

This section describes the process by which data can be exported from maintenance objects used with Oracle Utilities Customer Cloud Service and Oracle Utilities Meter Solution Cloud Service for use in external applications and systems.

This section provides details regarding how the data export process is used with specific maintenance objects. See Data Export in the *Application Framework Administrative User Guide* for a general description of the data export process.

- Enabling Maintenance Objects for Generalized Data Export
- Measurement Data Export
- Initial Measurement Data Header Export
- Aggregated Measurement Data Export

Enabling Maintenance Objects for Generalized Data Export

Export of data from most maintenance objects is supported via "Generalized" export. See Generalized Data Export in the *Application Framework Administrative User Guide* for more information about generalized data export.

Data Export Control Records for Maintenance Objects

Data export from a maintenance object requires a Data Export Control record configured for that maintenance object. See **Enabling Data Export** in Generalized Data Export in the *Application Framework Administrative User Guide* for more information.

Copyright © 2026, Oracle and/or its affiliates.

Change Data Capture for Measuring Components and Usage Subscriptions

For most maintenance objects, changes are tracked via the Capture Change for Ongoing Export audit algorithm (F1-MO-REGCHN).

Measuring Components and Usage Subscriptions use a "Most Recent" date field to track related transactions (Initial Measurements and Measurements in the case of Measuring Components and Usage Transactions in the case of Usage Subscriptions). These "Most Recent" fields are updated with each new initial measurement or usage transaction created by the system.

In order to prevent the Change Data Capture process from capturing changes based on these frequent transactions (daily for measurements and monthly for usage transactions), the Capture Change for Ongoing Export audit algorithm has been disabled on the Measuring Component and Usage Subscription maintenance objects.

Enabling change data capture and generalized data export for these maintenance objects requires adding the “Generalized Data Export BO Change Data Capture” (D1-BO-REGCHN) Audit algorithm to the appropriate Measuring Component and Usage Subscription business objects. This algorithm creates records in the Data Export Update table used by the generalized export process.

Measurement Data Export

Export of data from the Measurement maintenance object is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export.

The Measurement maintenance object is configured to support data export via the following options:

Option Type	Option Value
Data Export Class	F1SE (Specialized)
Export Batch Control	D1-SMDEC (Specialized Measurement Data Extract Current)

Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Measurement maintenance object:

- **Initial Extracts** gather an initial extract of measurement data, controlled by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the measurement data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the **Measurement Date/Time** and **Last Update Timestamp** fields on the Measurement table, as well as the **Export Through Date/Time** field on the Data Export Control record (see below) to include only changes since the last extract.

Copyright © 2026, Oracle and/or its affiliates.

- **Daily Historical Extracts** gather measurement data for measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Measurement table, this extract is based on Change Data Capture. During initial measurement data processing if an initial measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.
  Enabling historical change data capture requires configuration of the following algorithm parameters:
  - Capture Historical Measurement Changes: true
  - Historical Measurement Data Extract Batch Control: D1-SMDEH (Specialized Measurement Data Extract Historical)
  - Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

  These parameters should be configured as described on the following algorithms:
  - Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
  - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
  - Update Latest Date/Time on MC from a Future Measurement (D1-UPD-DTMCF)
  - Update Latest Date/Time on Scalar Measuring Component (D1-UPDDTSCMC)
  - Re-derive other values (D1-REDEROVAL)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

Data Export Control Records

Data export from the Measurement maintenance object requires a Data Export Control record. This record designates that the Measurement maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized Measurement Initial Extract** (D1-SMDIE): Used for initial extract of measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the measurement data prior to using the current extract (D1-SMDEC) and the historical extract (D1-SMDEH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all measurements with a measurement date/time that is greater than the input start date/time and less than or equal to the input end date/time.

Copyright © 2026, Oracle and/or its affiliates.

- **Specialized Measurement Data Extract Current** (D1-SMDEC): Used for daily extracts of current (new) measurement data. This process extracts data that has changed since the last extract process has run (based on the **Export Through Date/Time** field on the Data Export Control record. This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized Measurement Data Extract Historical** (D1-SMDEH): Used for daily extracts of historical measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table.

Batch processing is threaded by on Measuring Component ID, and creates one or more files based on the number of threads processed and batch control parameters.

File Creation and Naming

The extract process uses two controls to determine the number of files to be created:

1. Each thread will create at least 1 file (more threads result in smaller files).
2. The **File Size Limit** parameter can be used to specify set the number of measurement rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve measurement data for a set of Measuring Component IDs, but beyond that there will be no control for how the data is loaded into the file.

File Naming

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- MSRMT_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<thread file indicator>_<process date time>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file
- <total thread count> is the total number of threads
- <thread file indicator>: is an identified for the first and last files created by a thread, S for first file E for last file
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_S_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_20191104173233.csv.gz
- Last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_E_20191104173233.csv.gz

Copyright © 2026, Oracle and/or its affiliates.

File Format

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
MEASR_COMP_ID	CHAR(12)	000004769213	Primary Key of Channel from Meter (aka Device)	X	X	X
MSRMT_DTTM	DATE	2019-01-01T00:00:00Z	End of the time period the measurement covers. For example, if this were an hourly interval it would cover 2018-12-31T23:00:00Z to 2019-01-01T00:00:00Z.	X	X	X
PREV_MSRMT_DTTM	DATE	2018-12-31T00:00:00Z	For scalar measurements this identifies the start of the period being measured. Note: there should be another measurement for the same MC with this date as the MSRMT_DTTM.		X	

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
MSRMT_COND_FLG	CHAR(6)	501000	Defines the quality of the measurement (e.g. estimated, regular, super). Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup.	X	X	X
MSRMT_USE_FLG	CHAR(4)	D101	Indicates whether the measurement is usable. Valid values defined on CI_LOOKUP.		X	
USER_EDITED_FLG	CHAR(4)	D1UE	Indicates whether the user has edited the measurement. Valid values defined on CI_LOOKUP.	X	X	X
READING_VAL	NUMBER(16,6)	750000000	This is the Dial Reading which is used to calculate the consumed amount.		X	X
READING_COND_FLG	CHAR(6)	501000	Defines the quality of the Daily Reading (e.g. estimated, regular, super). Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup.			X

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
COMBINED_MULTIPLIER	NUMBER(18,6)	1	Contains the product of the Measuring Component Multiplier and the Installation Constant. The MSRMT_VAL will have been multiplied by this number.	X	X	X
MSRMT_VAL	NUMBER(16,6)	1.23	This is the consumption reported from the meter after going through VEE (aka billing ready).	X	X	X
MSRMT_VAL1	NUMBER(16,6)		This and the subsequent measurement values (MSRMT_VAL 1 through 10) are derived values that are computed from the core MSMRT_VAL. Examples of this are UOM conversion (CCF to THERM) or loss adjustment.	X	X	X
MSRMT_VAL2	NUMBER(16,6)			X	X	X
MSRMT_VAL3	NUMBER(16,6)			X	X	X
MSRMT_VAL4	NUMBER(16,6)			X	X	X
MSRMT_VAL5	NUMBER(16,6)			X	X	X
MSRMT_VAL6	NUMBER(16,6)			X	X	X
MSRMT_VAL7	NUMBER(16,6)			X	X	X
MSRMT_VAL8	NUMBER(16,6)			X	X	X
MSRMT_VAL9	NUMBER(16,6)			X	X	X
MSRMT_VAL10	NUMBER(16,6)			X	X	X
CRE_DTTM	DATE	2019-10-31T11:09:31Z	When the record was created.	X	X	X

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
LAST_UPDATE_DTTM	DATE	2019-11-09T16:35:02Z	When the record was last updated.	X	X	X

Sample Export File

The following is a sample export file for interval data spanning a six hour window. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
000004769213,2019-01-01T08:00:00Z,,501000,,,,1,1.3715,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T09:00:00Z,,501000,,,,1,1.882,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T10:00:00Z,,501000,,,,1,1.838,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T11:00:00Z,,501000,,,,1,1.8585,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T12:00:00Z,,501000,,,,1,1.836,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T13:00:00Z,,501000,,,,1,1.83,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:39:09Z,2019-08-09T20:39:09Z
000671510494,2019-01-01T08:00:00Z,,501000,,,,1,0.486,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T09:00:00Z,,501000,,,,1,0.4965,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T10:00:00Z,,501000,,,,1,0.45,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T11:00:00Z,,501000,,,,1,0.4725,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T12:00:00Z,,501000,,,,1,0.4575,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T13:00:00Z,,501000,,,,1,0.447,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
020643301975,2019-01-01T08:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T09:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T10:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T11:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T12:00:00Z,,501000,,,,1,0.0165,0,0,0,0,0,0,0,0,0,0,201
9-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T13:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
```

Initial Measurement Data Header Export

Copyright © 2026, Oracle and/or its affiliates.

Export of header data from the Initial Measurement Data maintenance object is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export.

The Initial Measurement Data maintenance object is configured to support data export via the following options:

Option Type	Option Value
Data Export Class	F1SE (Specialized)
Export Batch Control	D1-SIMDC (Specialized IMD Header Extract Current)

> **Note**
>
> When Initial Measurement Data Header Export is used with Direct Measurement Processing, exports contain only IMD header information for exceptions, including Informational Exception Headers created due to "Informational" exceptions.

Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Initial Measurement Data maintenance object:

- **Initial Extracts** gather an initial extract of initial measurement data headers, controlled by providing a start and end date time of the initial measurement data headers that should be extracted. This process should be used to perform the initial extract of the data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the **Last Update Timestamp** field on the Initial Measurement Data table, as well as the **Export Through Date/Time** field on the Data Export Control record (see below) to include only changes since the last extract.
- **Daily Historical Extracts** gather data for initial measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Initial Measurement Data table, this extract is based on Change Data Capture. During initial measurement data processing if an initial measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.
  Enabling historical change data capture requires configuration of the following algorithm parameters:
  - Capture Historical Measurement Changes: true
  - Historical IMD Extract Batch Control: D1-SMDEH (Specialized IMD Header Extract - Historical)
  - Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

  These parameters should be configured as described on the following algorithms:

Copyright © 2026, Oracle and/or its affiliates.

- Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
- Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
- Update Latest Date/Time on MC from a Future Measurement (D1-UPD-DTMCF)
- Update Latest Date/Time on Scalar Measuring Component (D1-UPDDTSCMC)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

Data Export Control Records for Maintenance Objects

Data export from the Initial Measurement Data maintenance object requires a Data Export Control record. This record designates that the Initial Measurement Data maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized IMD Header Initial Extract** (D1-SIMDI): Used for initial extract of initial measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the IMD headers that should be extracted. This process should be used to perform the initial extract of the IMD header data prior to using the current extract (D1-SIMDC) and the historical extract (D1-SIMDH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all IMD headers with a IMD to date/time that is greater than the input start date/time and less than or equal to the input end date/time.
- **Specialized IMD Header Data Extract - Current** (D1-SIMDC): Used for daily extracts of current (new) initial measurement data. This process extracts data that has changed since the last extract process has run (based on **the Export Through Date/Time** field on the Data Export Control record). This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized IMD Header Extract - Historical** (D1-SIMDH): Used for daily extracts of historical initial measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table.

Batch processing is threaded by on Initial Measurement Data ID, and creates one or more files based on the number of threads processed and batch control parameters.

File Creation and Naming

The extract process uses two controls to determine the number of files to be created:

1. Each thread will create at least 1 file (more threads result in smaller files)

Copyright © 2026, Oracle and/or its affiliates.

2. The **File Size Limit** parameter can be used to specify set the number of initial measurement data rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve initial measurement data for a set of Initial Measurement Data IDs, but beyond that there will be no control for how the data is loaded into the file.

File Naming

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- IMD_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<file number>_<last file indicator>_<process date>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file
- <total thread count> is the total number of threads
- <file number> is the number of the file
- <last file indicator>: identifies the last files created by a thread (E)
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads:
  IMD_EXPORT_SOME_PREFIX_135_3_10_1_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads:
  IMD_EXPORT_SOME_PREFIX_135_3_10_2_20191104173233.csv.gz
- Last file of 3rd out of 10 threads:
  IMD_EXPORT_SOME_PREFIX_135_3_10_E_20191104173233.csv.gz

File Format

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
INIT_MSRMT_DATA_ID	CHAR(12)	52482245116604	Primary Key of the initial measurement	X	X	X
MEASR_COMP_ID	CHAR(12)	000004769213	Primary Key of the measuring component for the initial measurement	X	X	X
IMD_EXT_ID	CHAR(12)	external_file	External ID for the initial measurement. Usually the name of the file that the initial measurement data was received in.	X	X	X
BUS_OBJ_CD	CHAR(12)	D1-ManualIMDInterval	The business object that governs the initial measurement	X	X	X
BO_STATUS_CD	VARCHAR(30)	COMPLETED	The status of the initial measurement	X	X	X
BO_STATUS_REASON_CD	VARCHAR(30)	FORCE COMPLETE	A reason code for the status. Generally populated when there was a VEE exception that was overridden and the initial measurement was completed without fixing the error.	X	X	X

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
D1_FROM_DTTM	DATE/TIME	2019-01-01T00:00:00Z	Start of the time period the initial measurement covers. For subtractive scalar data this represents the prior measurement that will be used as the start reading when calculating consumption. For interval data this represents the start of the first interval for the initial measurement. Note: no measurements will be created for this date/time as MDM represents intervals as the end of the interval. So the first interval of the initial measurement will be this date/time plus the interval size (e.g. if the interval size is 15 minutes then the first interval would be 15 minutes after this date/time).	X	X	X

Copyright © 2026, Oracle and/or its affiliates.

				Applicability		
Field	**Data Type**	**Sample Content**	**Description**	**Interval**	**Scalar**	**Subtractive Interval**
D1_TO_DTTM	DATE/TIME	2019-01-02T00:00:00Z	End of the time period the initial measurement covers. For scalar data this date/time will match the measurement. For interval data this will match the date/time of the last interval measurement.	X	X	X
CRE_DTTM	DATE/TIME	2019-01-03T00:00:00Z	Date and time when the initial measurement was initially created	X	X	X
STATUS_UPD_DTTM	DATE/TIME	2019-01-03T00:00:00Z	Date and time when the initial measurement's status was last updated	X	X	X
LAST_UPDATE_DTTM	DATE/TIME	2019-01-03T00:00:00Z	Date and time when the initial measurement was last updated. Note: updates can happen outside of a status change so this can be more recent than the STATUS_UPD_DTTM.	X	X	X

Sample Export File

The following is a sample export file containing initial measurement header records. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
55394295586154,553275345711,external file1,D1-
ManualIMDScalar,ERROR,,2020-01-02T08:00:00Z,2022-01-01T08:00:00Z,2020-08-17T18
:47:41Z,2020-08-17T18:47:41Z,2020-08-17T18:48:54Z
```


Copyright © 2026, Oracle and/or its affiliates.

```
55392270285377,553886220134,external file2,D1-
ManualIMDInterval,ERROR,,2022-01-01T08:00:00Z,2022-01-02T08:00:00Z,2020-08-15T
04:18:43Z,2020-08-15T04:25:57Z,2020-08-15T04:25:57Z
55383485376359,553886220134,external file3,D1-
ManualIMDInterval,ERROR,,2022-01-01T08:00:00Z,2022-01-02T08:00:00Z,2020-08-15T
04:47:36Z,2020-08-15T04:55:03Z,2020-08-15T04:55:03Z
```

Aggregated Measurement Data Export

Export of data from the Aggregated Measurement maintenance object (D1-AGGMSRMT) is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export. The Aggregated Measurement maintenance object is configured to support data export via the following options:

Option Type	Option Value
Data Export Class	F1SE (Specialized)
Export Batch Control	D1-SAMDC (Specialized Agg. Measurement Extract Cur.)

Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Aggregated Measurement maintenance object:

- **Initial Extracts** gather an initial extract of aggregated measurement data, controlled by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the aggregated measurement data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather aggregated measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the Measurement Date/Time and Last Update Timestamp fields on the Aggregated Measurement table, as well as the Export Through Date/Time field on the Data Export Control record (see below) to include only changes since the last extract.
- **Daily Historical Extracts** gather aggregated measurement data for measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Aggregated Measurement table, this extract is based on Change Data Capture. During initial measurement data processing if an aggregated measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.

Enabling historical change data capture for aggregated measurement data requires configuration of the following algorithm parameters:

- Capture Historical Aggregation Changes: true
- Aggregation Historical Batch Control: D1-SAMDH (Specialized Aggregated Measurement Data Extract Historical)

Copyright © 2026, Oracle and/or its affiliates.

- Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

These parameters should be configured as described on the following algorithms:

- Create General Process Record for Historical Dynamic Aggregation (D1CGPHISAGG)
- Create General Process Record for Historical Static Aggregation (D1CGPHISA)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

Data Export Control Records

Data export from the Aggregated Measurement maintenance object requires a Data Export Control record. This record designates that the Aggregated Measurement maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized Aggregate Msrmt. Initial** (D1-SAMDI): Used for initial extract of aggregated measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of aggregated measurement data prior to using the current extract (D1-SAMDC) and the historical extract (D1-SAMDH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all aggregated measurements with a measurement date/time that is greater than the input start date/time and less than or equal to the input end date/time.
- **Specialized Agg. Measurement Extract Cur.** (D1-SAMDC): Used for daily extracts of current (new) aggregated measurement data. This process extracts data that has changed since the last extract process has run (based on the **Export Through Date/Time** field on the Data Export Control record. This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized Aggregated Measurement Data Extract Historical** (D1-SAMDH): Used for daily extracts of historical aggregated measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table

Batch processing is threaded by on Measuring Component ID, and creates one or more files based on the number of threads processed and batch control parameters.

File Creation and Naming

The extract process uses two controls to determine the number of files to be created:

1. Each thread will create at least 1 file (more threads result in smaller files).

Copyright © 2026, Oracle and/or its affiliates.

2. The **File Size Limit** parameter can be used to specify set the number of aggregated measurement rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve aggregated measurement data for a set of Measuring Component IDs, but beyond that there will be no control for how the data is loaded into the file.

File Naming

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- MSRMT_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<thread file indicator>_<process date time>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file
- <total thread count> is the total number of threads
- <thread file indicator>: is an identified for the first and last files created by a thread, S for first file E for last file
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_S_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_20191104173233.csv.gz
- Last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_E_20191104173233.csv.gz

File Format

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").

Field	Data Type	Sample Content	Description
MEASR_COMP_ID	CHAR(12)	000004769213	Primary Key of Channel from Meter (aka Device)

Copyright © 2026, Oracle and/or its affiliates.

Field	Data Type	Sample Content	Description
MSRMT_DTTM	DATE	2019-01-01T00:00:00Z	End of the time period the measurement covers. For example, if this were an hourly interval it would cover 2018-12-31T23:00:00Z to 2019-01-01T00:00:00Z.
PREV_MSRMT_DTTM	DATE	2018-12-31T00:00:00Z	For scalar measurements this identifies the start of the period being measured. Note: there should be another measurement for the same MC with this date as the MSRMT_DTTM.
MSRMT_COND_FLG	CHAR(6)	501000	Defines the quality of the measurement (e.g. estimated, regular, super). Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup.
MSRMT_USE_FLG	CHAR(4)	D101	Indicates whether the measurement is usable. Valid values defined on CI_LOOKUP.
USER_EDITED_FLG	CHAR(4)	D1UE	Indicates whether the user has edited the measurement. Valid values defined on CI_LOOKUP.
READING_VAL			Values for this field are not included in the aggregated measurement export.
READING_COND_FLG			Values for this field are not included in the aggregated measurement export.
COMBINED_MULTIPLIER			Values for this field are not included in the aggregated measurement export.
MSRMT_VAL	NUMBER(16,6)	1.23	This is the consumption reported from the meter after going through VEE (aka billing ready).

Copyright © 2026, Oracle and/or its affiliates.

Field	Data Type	Sample Content	Description
MSRMT_VAL1	NUMBER(16,6)		This and the subsequent measurement values (MSRMT_VAL 1 through 10) are derived values that are computed from the core MSMRT_VAL. Examples of this are UOM conversion (CCF to THERM) or loss adjustment.
MSRMT_VAL2	NUMBER(16,6)		
MSRMT_VAL3	NUMBER(16,6)		
MSRMT_VAL4	NUMBER(16,6)		
MSRMT_VAL5	NUMBER(16,6)		
MSRMT_VAL6	NUMBER(16,6)		
MSRMT_VAL7	NUMBER(16,6)		
MSRMT_VAL8	NUMBER(16,6)		
MSRMT_VAL9	NUMBER(16,6)		
MSRMT_VAL10	NUMBER(16,6)		
CRE_DTTM	DATE	2019-10-31T11:09:31Z	When the record was created.
LAST_UPDATE_DTTM	DATE	2019-11-09T16:35:02Z	When the record was last updated.

Sample File

The following is a sample export file for interval data spanning a six hour window. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
000004769213,2019-01-01T08:00:00Z,,501000,,,,,1,1.3715,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T09:00:00Z,,501000,,,,,1,1.882,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T10:00:00Z,,501000,,,,,1,1.838,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T11:00:00Z,,501000,,,,,1,1.8585,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T12:00:00Z,,501000,,,,,1,1.836,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T13:00:00Z,,501000,,,,,1,1.83,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:39:09Z,2019-08-09T20:39:09Z
000671510494,2019-01-01T08:00:00Z,,501000,,,,,1,0.486,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T09:00:00Z,,501000,,,,,1,0.4965,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T10:00:00Z,,501000,,,,,1,0.45,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T11:00:00Z,,501000,,,,,1,0.4725,0,0,0,0,0,0,0,0,0,0,201
```


Copyright © 2026, Oracle and/or its affiliates.

```
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T12:00:00Z,,501000,,,,1,0.4575,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T13:00:00Z,,501000,,,,1,0.447,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
020643301975,2019-01-01T08:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T09:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T10:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T11:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T12:00:00Z,,501000,,,,1,0.0165,0,0,0,0,0,0,0,0,0,0,201
9-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T13:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
```

Service Point Consumption Extracts

Service Point Consumption extracts are measurement data extracts for a set of service points assigned to an entity group.

Configuration options for these extracts include:

- Type of measurement data to extract (interval or scalar)
- Unique combinations of unit of measure (UOM), time of use (TOU), and service quantity identifier (SQI) associated with the usage subscriptions
- Date and time parameters and frequency of extract
- Extract file naming and location

Service Point Consumption extracts are created via Requests. Service Point Consumption extracts are exported to a file in a specified file location, either on an application server (in an on-premises implementation) or an Oracle Cloud Infrastructure Object Storage location. External URL attachments linking to extract files are created on the Request for each extract.

This section describes the configuration needed for these types of extracts.

> **Note**
>
> Service Point Consumption Extracts are only available in cloud (SaaS) implementations.

Service Point Consumption Extract Entity Groups

Entity groups used with Service Point Consumption Extracts must be created using the Service Point Measurement Extract Group (D1-SPMeasurementExtractGroup) business object. These entity groups will have a Group Category of "SP Measurement Extract".

Refer to Understanding Entity Groups for more information.

Copyright © 2026, Oracle and/or its affiliates.

Service Point Consumption Extract Request Types and Requests

Configuration for service point consumption extracts are defined via Requests. See Understanding Requests in the *Application Framework Administrative User Guide* for more information about creating these records.

Service point consumption extract request types should be created using the SP Measurement Extract Request Type (D1-SPMsrmtExtractRequestType) business object, and define the following parameters:

- The Device Identifier Type (such as Asset Number, Badge Number, and so on) to include in the extract file
- File parameters, including:
  - **File Name**: String used as the prefix for extract file names
  - **File Location**: The location where extract files will be saved (see External File Storage in the *Application Framework Administrative User Guide*)
  - **Child Folder**: An optional child folder in the file location
  - **Record Limit**: A limit on the number of records to include in extract files

Service point consumption extract requests should be created using the SP Measurement Extract Request (D1-SPMsrmtExtractRequest) business object, and define the following parameters:

- **Start/End Date/Time**: The date and time range for the extract
- **Entity Group**: The entity to which the service points for which consumption is extracted are assigned
- **UOM/TOU/SQI**: The type of measurement data extracted, filtered based on the Service Type for the selected Entity Group
- **Device Identifier Type**: The device ID type to include in the extract file (such as Asset Number, Badge Number, and so on)
- **Consumption Type**: The type of consumption (Interval or Scalar) for extract
- **Target Interval Size**: The interval size for the extract, applicable only if Consumption Type is Interval. Note that the interval size must be larger than the SPI of the measurement data
- **Batch Number**: The batch number for the batch process used to perform the extract. This is populated by the batch process that performs the extract.

Extract files for Service Point Consumption Requests are created as external URL attachments on the Request that processed the extract.

Interval Snapshot Mapping

Service point consumption extracts require interval snapshot mapping data be defined for the interval size and date range of the extract. See About Interval Snapshot Mapping and Maintaining Interval Snapshot Mapping for more information.

Request Processing

Creating a request starts batch job to extract the usage to Oracle Object Storage or a file directory in a CSV format. This CSV format can easily be imported into Excel or other software.

When a request is created, it starts in the Pending state, and an Enter algorithm creates a deferred request and the SP Measurement Extract Request Monitor (D1-SPMRQ) batch

Copyright © 2026, Oracle and/or its affiliates.

process moves the request into the Extract state, where the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm performs the extract process.

The extract process creates an "External URL" attachment in which the attachment contains a URL to the extract file in the location specified in the Request Type. Attachments created for a request can be accessed via the **Request Attachments** zone in the **Request** portal.

File Naming and Formats (SECTION)

Extract file names use the following format:

```
<file location><file name>-<request id>-<start dttm>-<end dttm>_<file number>
```

where:

- **<file location>** is the **File Location** specified in the Request Type
- **<file name>** is the **File Name** specified in the Request Type
- **<request id>** is the ID of the Request
- **<start dttm>** is the start date and time for the request, displayed as YYYYMMDDHH24MISS
- **<end dttm>** is the end date and time for the request, displayed as YYYYMMDDHH24MISS
- **<file number>** is the file number for the extract file. This is a 4-digit number with padded leading zeros. Multiple files are created if a **Record Limit** is specified on the Request Type and the records extracted exceeds the limit.

For example if the **File Location** was set to "file-storage://CM-CloudStorage/sp-measurement-extracts/" and the **File Name** was set to "sp-kWh-extract" with an extract date/time of 2026-01-01 00:00:00 to 2026-02-01 00:00:00 then the first file would be named:

Scalar Record Format

```
file-storage://CM-CloudStorage/sp-measurement-extracts/sp-kWh-
extract-10392739850273-20260101000000-20260102000000_0001.csv.gz
```

Scalar records will only have one measurement per row. The **Start Date/Time** isn't the start of the day but rather the start of the measurement. Measurements are extracted based on when they end, so the first measurement chronologically for each measuring component may cover a period that falls before the extract time period. Extract data comes from the following tables and columns:

Header Field	Table	Column	Notes
Start Date/Time	D1_MSRMT	PREV_MSRMT_DTTM	Converted from the time zone defined in Installation Options to the service point time zone
Customer Name	D1_CONTACT_NAME	NAME_VALUE	This is for the "Main" usage subscription Contact and for the Primary name
Address	D1_SP	ADDRESS1 + ADDRESS2 + ADDRESS3 + ADDRESS4	ADDRESS2 through 4 will be trimmed

Copyright © 2026, Oracle and/or its affiliates.

Header Field	Table	Column	Notes
Service Point	D1_SP	D1_SP_ID	
Local Time Zone	CI_TIME_ZONE	F1_TIMEZONE_NAME	The time zone from the service point
Measuring Component	D1_MEASR_COMP	MEASR_COMP_ID	
Meter Number	D1_DVC_IDENTIFIER	ID_VALUE	Based on the device identifier type to use (such as badge number, serial number, and so on) from the Request Type
How to Use Flag	D1_MEASR_COMP	MEASR_COMP_USAGE_FLG	
UOM	D1_MC_TYPE_VALUE	D1_UOM_CD	Unit of Measure (Description)
TOU	D1_MC_TYPE_VALUE	D1_TOU_CD	Time of Use (Description)
SQI	D1_MC_TYPE_VALUE	D1_SQI_CD	Service Quantity Identifier (Description)
Date/Time	D1_MSRMT	MSRMT_DTTM	Converted from the time zone defined in Installation Options to the service point time zone
Measurement	D1_MSRMT	MSRMT_VAL	The appropriate Measurement Value field based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI. This is the main measurement Measurement Value the majority of the time but for Gas or Water, derived values from the other measurement value fields may be requested.
Condition	D1_MSRMT	MSRMT_COND_FLG	

Interval Record Format

Interval records will have multiple measurements per row. Extract data comes from the following tables and columns:

Header Field	Table	Column	Notes
Start Date/Time	D1_INT_SNAP_MAP	D1_LOCAL_DT	Converted from the time zone defined in Installation Options to the service point time zone

Copyright © 2026, Oracle and/or its affiliates.

Header Field	Table	Column	Notes
Customer Name	D1_CONTACT_NAME	NAME_VALUE	This is for the "Main" usage subscription Contact and for the Primary name
Address	D1_SP	ADDRESS1 + ADDRESS2 + ADDRESS3 + ADDRESS4	ADDRESS2 through 4 will be trimmed
Service Point	D1_SP	D1_SP_ID	
Local Time Zone	CI_TIME_ZONE	F1_TIMEZONE_NAME	The time zone from the service point
Measuring Component	D1_MEASR_COMP	MEASR_COMP_ID	
Meter Number	D1_DVC_IDENTIFIER	ID_VALUE	Based on the device identifier type to use (such as badge number, serial number, and so on) from the Request Type
How to Use Flag	D1_MEASR_COMP	MEASR_COMP_USAGE_FLG	
UOM	D1_MC_TYPE_VALUE	D1_UOM_CD	Unit of Measure (Description)
TOU	D1_MC_TYPE_VALUE	D1_TOU_CD	Time of Use (Description)
SQI	D1_MC_TYPE_VALUE	D1_SQI_CD	Service Quantity Identifier (Description)
Total	D1_MSRMT	MSRMT_VAL<N>	The sum of the measurements for the Start Date/Time for the Measuring Component. The appropriate Measurement field will be used based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI.
Time<N>	D1_MSRMT	MSRMT_DTTM	This is date and time of each interval, calculated as follows: Start Date/Time (D1_LOCAL_DT) + (interval number * SPI). SPI is based on the **Target Interval Size** if it was provided in the Request or the **Interval Size** of the Measuring Component Type if not.

Copyright © 2026, Oracle and/or its affiliates.

Header Field	Table	Column	Notes
Measurement<N>	D1_MSRMT	MSRMT_VAL<N>	The appropriate Measurement Value field based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI. This is the main measurement Measurement Value the majority of the time but for Gas or Water, derived values from the other measurement value fields may be requested.
Condition<N>	D1_MSRMT	MSRMT_COND_FLG	This will either be the the Minimum (MIN) or Maximum (MAX) Measurement Condition Flag based on the configuration of the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm. The default is to use the MIN. Note: MIN or MAX is only important for Measuring Components that have a lower SPI than the **Target Interval Size**. If no **Target Interval Size** is provided then data is extracted based on the **Interval Size** of the of the Measuring Component Type.

Interval data is values are returned as a single string of comma-separated values that are sub delimited by tildas (~). For example:

```
3.732~1~501000,5.804~2~501000,6.432~3~501000,5.572~4~501000,7.384~5~501000,7.3
58~6~501000,4.67772~7~501000,7.8132~8~501000,7.83564~9~501000,6.14856~10~50100
0,8.0988~11~501000,8.16816~12~501000,4.63186~13~501000,7.7366~14~501000,7.7588
2~15~501000,6.08828~16~501000,8.0194~17~501000,8.08808~18~501000,3.808~19~5010
00,4.448~20~501000,7.69~21~501000,6.94~22~501000,7.964~23~501000,7.316~24~5010
00
```

Customizing Extract Output

The Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm uses the Extract Measurements for SP Entity Group Library (D1EMSPEGLib) script to return queries, format headers, and format records. This script implements three high level types of methods:

Copyright © 2026, Oracle and/or its affiliates.

1. Method to return the query that will extract the measurement data and master data details
    a. This takes in the same fields as are available in the plug-in driven script process data plug-in spot
    b. This returns a prepared statement
    c. There is one script for Interval and one script for Scalar since the queries are very different
2. Method to populate a header row for the CSV file
    a. For scalar this is always the same fields since we only ever extract one row at a time
    b. For interval this creates the measurement fields according to the maximum SPI or the target SPI and whether or not the extract period covers a DST transition
3. Method to process the returned data into the formatted row
    a. This takes in the results from executing the queries
    b. This formats the data into what will be streamed into the file and return that as a string

Use the **Script** portal to view this script in more detail.

To customize the data extracted via this process (for example to add an additional column), a custom (CM) version of the Groovy Library script can created. This involves the following steps:

1. Duplicate the Extract Measurements for SP Entity Group Library (D1EMSPEGLib) script.
2. Adjust the generate query method(s) to retrieve the additional information.
3. Adjust the generate header method(s) to return the new field for the header.
4. Adjust the result formatting method(s) to place the new data into the result string according to their requirements.
5. Add this new CM script as the value for the **Override Extract Groovy Library** parameter on the the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm.

Consumption Extract Type

- Understanding Consumption Extract Type
- Configuring Consumption Extract Type

Understanding Consumption Extract Type

The Consumption Extract Type controls which service point's measurements are extracted for Oracle Utilities DataConnect, what type of measurement is extracted, and how the measurements are grouped into TOU periods (if preferred).

The Consumption Extract Type also controls the request type used for creating Consumption Extract Requests of this type, how frequent automated requests are created for incremental extract, the batch jobs that are triggered for extracting data and the algorithms that extract, format, and write the data that is extracted.

Refer to the Defining DataConnect Options section for more details on how this object is specifically put into use.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Consumption Extract Type

This portal is used to display and maintain a Consumption Extract Type.

Refer to Understanding Consumption Extract Types for more information.

You can access the portal by selecting **Admin**, then **Integration**, then **Consumption Extract Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Consumption Extract Type List**: displays all of the Consumption Extract Types so you can choose the one you want to display in more detail
- **Consumption Extract Type**: shows the specific configuration for the selected Consumption Extract Type

Analytics Configuration

This portal provides a bird's eye view of the configuration information for Oracle Utilities Analytics. It provides links and guidelines for the areas that need configuration to successfully run the extract transfer and load processes from Oracle Utilities Analytics.

Refer to the *Oracle Utilities Analytics Administration Guide* for more information.

You can access the portal from the **Admin**, and then **Analytics Configuration**, and then **Customer Analytics Configuration**

The following zones may appear as part of the portal's **Main** tab page

- **Bucket Configuration List.** This zone lists bucket configurations to set up.
- **BI-Related Business Objects Information.** This zone lists additional control entities to set up.

The following zones may appear as part of the portal's **OWB-Based ETL** tab page

- **Outbound Sync BOs and Algorithm List.** This zone lists business objects and algorithms involved in the extract process.
- **BI-Oriented Extendable Lookup.** This zone lists extendable lookups to set.
- **External Data Source Indicators List.** This zone lists the external identifiers of the various sources for your analytics data.

Note: Service Point business objects can make use of the following System Events:

- **Service Point Snapshot**: This system event defines the algorithm used to create a snapshot of the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Snapshot."
- **Usage Snapshot**: This system event defines the algorithm used to create a usage snapshot of the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Usage Snapshot."
- **Unreported Usage Analysis Snapshot**: This system event defines the algorithm used to create a snapshot of the Service Point's consumption since the last usage transaction for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Unreported Usage Analysis Snapshot."

Copyright © 2026, Oracle and/or its affiliates.

- **SP VEE Exception Snapshot**: This system event defines the algorithm used to create a snapshot of VEE exceptions for the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - VEE Exception Snapshot."

Migrating Data Extract Processing from Legacy Measurement Processing to Direct Measurement Processing

When processing data extracts using Direct Measurement Processing, change data capture information is recorded in the Measurement Extract Request table instead of the General Process table. This requires that the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes". This applies to all types of data extracts, including:

- Configurable Consumption Extracts
- Measurement Data Extracts
- Initial Measurement Data Header Extracts
- Aggregated Measurement Data Extracts
- DataConnect Consumption Extracts (see Oracle Utilities Dataconnect / Opower for more information about DataConnect extracts)

For customers who had been using Legacy Measurement Processing and processing data extracts (in which change data capture information is recorded in the Genera Process table), migrating to using Direct Measurement Processing will involve transitioning out of Legacy-based extract processing and into Direct Measurement Processing extract processing.

There are two approaches to this transition, one involves continuing to use the current batch controls, the other involves creating new batch controls.

Using Existing Batch Controls

To use existing batch controls, do the following:

1. Pause all Initial Measurement Data Processing.
2. Run each of the Historical extract processes to process all remaining records in the General Process table.
3. Enable use of the Measurement Extract Request table by setting the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes".
4. Flush the batch server.
5. Resume Initial Measurement Data processing.

Creating New Batch Controls

To create new batch controls, do the following:

1. Duplicate all of the Historical extract batch controls
2. Update the Initial Measurement Data algorithms and Consumption Extract Types to reference the new batch controls.
3. Flush the batch server.
4. Process one final run using the original Historical extract batch controls.

Copyright © 2026, Oracle and/or its affiliates.

5. Enable use of the Measurement Extract Request table by setting the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes".
6. Start using the new batch controls for all extract processing.

Aggregation

- Standard Aggregation
- Dynamic Aggregation

Standard Aggregation

- Configuring an Out-of-the-box Aggregation
- Understanding an Example Out-of-the-box Aggregation
- Creating a New Custom Aggregation

Configuring an Out-of-the-box Aggregation

Aggregation calculations should be run on an as needed basis. This can include running the following batch processes:

- Scanning for new aggregation dimension (D1-ADS): This process is applicable if the system is configured to use aggregation dimension scanners to detect new aggregation dimensions (such as a service point referencing a new transformer for which an aggregator measuring component doesn't currently exist)
- Performing aggregation calculations (D2-AGG): this batch handles the summarization of measurement values in order to create aggregated measurements.

Note that aggregation calculations should precede usage transaction processing if aggregated values serves as input to the calculation of bill determinants.

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more information on this functionality. Refer to Understanding Measuring Component Types for information on Measuring Component Types provided for aggregation.

Understanding an Example Out-of-the-box Aggregation

The Oracle Utilities Meter Data Management base package includes an aggregation that aggregates measurement quantities for constituent measuring components based on postal code and service type dimensions. The table below outlines the types of objects used in this aggregation, based on the steps outlined above), and the specific objects for each type.

Dependency Order	Object Type	Base Package Example
1	Aggregator Measuring Component Business Object	D2-Aggregator (Aggregator - Postal and Service Type)

Copyright © 2026, Oracle and/or its affiliates.

2	Aggregator Measuring Component UI Maps	Display: D2-AggMCDisp (Service Type and Postal Aggregator-Display) Maintenance: D2-AggMCMaint (Service Type and Postal Aggregator-Maintenance)
3	Aggregator Business Object Info Algorithm	D2-AMC-INFO (Service Type and Postal Aggregator - Information)
4	Find Constituent Measuring Components Algorithm	D2-DET-CMC (Find Constituent Measuring Components Based on Service Type and Postal)
5	Measuring Component Type	D2-AggregatorType (Aggregator Type)
6	Query Zone for Consumption Statistics Portal	D2-AGGMCQRY (Aggregator Search)
7	Dimension Scanner Activity Business Object	D2-ActivityAggDimScanner (Aggregator Creator - Postal / Service Type)
8	Dimension Scanner Activity UI Maps	Display: D2-AggDimScannerActDisp (Aggregator DS Activity-Display) Maintenance: D2-AggDimScannerActMaint (Aggregator DS Activity-Maintenance)
9	Dimension Scanner Activity Business Object Info Algorithm	D2-ADS-INFO (Aggregator Dimension Scanner Information)
10	Enter Algorithm for Scan State	D2-CRE-AGGMC (Aggregator MC Creation for Post Code and Service Type)
11	Activity Type	D2-ActivityTypeAggDimScanner (Aggregator Dimension Scanner Activity Type)

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

Creating a New Custom Aggregation

This section describes the overall process for creating a new custom Aggregation.

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

Execute the following steps:

1. Create a business object for the aggregator measuring component. This will flatten the dimensional value(s) into searchable characteristics. Whether this business object is a

Copyright © 2026, Oracle and/or its affiliates.

parent or a child of another aggregator business object depends on when periodic aggregation should occur:

a. If you want the periodic aggregation to occur when another aggregation occurs, it can be a child business object (meaning that it inherits the lifecycle (and therefore the deferred monitor) of the parent)

b. If you want to schedule its periodic aggregation independently from other aggregation business objects, this must NOT be a child business object as it will require its own deferred monitor (and deferred monitors can only be defined on parent business objects)

2. Create UI maps for the aggregator business object as follows:

   a. One to display the aggregator measuring component (Display)

   b. One to allow user to change / add a new one (Maintenance)

> **Note**
>
> A newer alternative to creating UI Maps would be to use UI Hints directly within the Business Object.

3. Create an info plug-in for the aggregator business object that concatenates together its dimension types and values.

4. Create a "Find Constituent Measuring Components" algorithm and plug it on the aggregator business object. This will be passed the aggregator measuring component and the from and to date/times. It will insert the constituent measuring component IDs and the respective from / to date-time of each onto a temporary table.

5. Create a measuring component type instance and reference the new aggregator measuring component business object (as well as the types of constituent measuring component types that should be aggregated).

6. Create a query zone for Consumption Statistics search to allow users to find the aggregator measuring component.

Optional steps:

1. Create a business object for the dimension scanner activity. This should be a child business object of the base package dimension scanner business object.

2. Create UI maps for the activity business object, as follows:

   a. One to display the dimension scanner activity (Display).

   b. One to allow users to change/add a new one (Maintenance).

> **Note**
>
> A newer alternative to creating UI Maps would be to use UI Hints directly within the Business Object.

3. Create an info plug-in that will describe what it scans.

4. Create an Enter algorithm on the Scan state that finds distinct combinations of the dimensional values and creates new aggregator measuring components when new ones are detected.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can reuse the base package deferred monitor **Aggregation Dimension Scanner Monitor** (D1-ADS) batch control .

Dynamic Aggregation

- Dynamic Aggregation Configuration Overview
- Aggregation Master Configuration
- Data Sources
- Dynamic Aggregation Measuring Component Types
- Aggregation Groups
- Measuring Component Sets
- Configuring Dynamic Aggregation Step-by-Step
- Extending Dynamic Aggregation

Dynamic Aggregation Configuration Overview

Standard aggregation processing uses specific algorithms configured to work with a specified set of dimensions. For example, the base package Service Type and Postal aggregation uses specific algorithms for dimension scanning and finding constituent measuring components.

In contrast, dynamic aggregation uses dynamic queries for the dimension scanning, find constituents, and aggregation processes. These dynamic queries are based on configuration of administrative data, including:

- **Data Sources** define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or other sources of data.
- **Aggregation Measuring Component Types** define the most important properties of aggregation measuring components used to store aggregated data.
- **Aggregation Groups** define the ordering of a series of related aggregations and the schedule of aggregation.
- **Measuring Component Sets** define the dimensions and criteria by which aggregation will be performed.

Aggregation Master Configuration

The Aggregation Master Configuration defines common parameters used with aggregation processing, including:

- Batch Controls used by the dimension scan and aggregation process, for both scheduled and adhoc execution.
- Details for how the system compares interval and billing data for the same set of customers in order to use only one source or other.

Copyright © 2026, Oracle and/or its affiliates.

Data Sources

- Understanding Data Sources
- Configuring Data Sources
- Configuring Data Source Template SQL

Understanding Data Sources

Data Sources define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or measuring component sets. Data sources also define configurations used when generating dynamic queries used by aggregation processing.

Data Source Classes

Data sources are defined by classes, each of which is based on a specific business object, and determines the type of configuration used by aggregation processing. More specifically, the data types configured on the data source are used when generating dynamic queries used by aggregation processing.

The table below lists the data source classes and the business object and configuration options for each. Consult the embedded help for more information about the specific configuration options used by each data source class.

Data Source Class	Usage	Business Object	Configuration Options
Measurements from US Service Points	Used when aggregating data from measuring components linked to service points via a usage subscription.	D1-DataSourceMeasurements	Measuring Component Types
Badged Items	Used when aggregating data from badged items.	D1-BadgedItemsDataSource	Item Types
Billed Service Quantities	Used when aggregating data based on billed services quantities.	D1-DataSourceBilledSerQuantity	Usage Subscription Types
Measuring Component Sets	Used when aggregating data from measuring component sets. Data sources of this class are used with composite aggregation.	D1-DataSourceMCSet	Usage Subscription Types
Unbadged Items	Used when aggregating data from unbadged items.	D1-DataSourceUnbadgedItems	Item Types

Copyright © 2026, Oracle and/or its affiliates.

Data Source Class	Usage	Business Object	Configuration Options
Measurements from US Direct Links	Used when aggregating data from measuring components directly linked to a usage subscription.	D1-DataSourceMeasurementDirect	Measuring Component Types
Service Point Quantities	Used when aggregating data based on service point quantities.	D1-DataSourceSPQuantities	Service Point Quantity Types

Data Source Template SQL

Data sources also provide template structured query language (SQL) used when generating dynamic queries used by aggregation processing. This template SQL is used as the basis for dynamic queries, and is extended by additional SQL generated from the data source configuration as well as configuration defined for measuring component sets, measuring components, and measuring component types that reference the data source. Template SQL is provided via the Data Source SQL (D1-DataSourceSQLLookup) extendable lookup. Values for this extendable lookup are used as the **Option Value** for the following business object options on the data source business object.

- **Dimension Scanner SQL**: The template SQL used when dimension scanning. This SQL is extended by the criteria, dimensions, and individually managed items configured on the measuring component sets being processed.
- **Find Constituents SQL**: The template SQL used when finding constituents. This SQL is extended by attributes defined on the aggregator measuring components being processed.
- **Aggregation SQL**: The template SQL used when performing aggregation. This SQL is extended by the value identifiers defined on the measuring component type of the aggregator measuring components being processed.

Configuring Data Sources

This portal is used to display and maintain data sources.

Refer to Understanding Data Sources for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, then **Data Source**.

The following zones may appear as part of the portal's **Main** tab page:

- **Data Source List**: This zone lists all data source records. Broadcast a record to display the details of the selected record.
- **Aggregation Data Source**: This zone provides information about the selected data source.

Configuring Data Source Template SQL

Template SQL used when generating dynamic queries used in aggregation processing is defined in the Data Source SQL (D1-DataSourceSQLLookup) extendable lookup. The base

Copyright © 2026, Oracle and/or its affiliates.

package data source business objects reference appropriate base package values from this extendable lookup.

For example, the **Aggregation SQL** option on the Measurements from US Service Points data source business object (D1-DataSourceMeasurements) references the "D1_MSRMT_FROM_US_SP_MCTYPE_SQL" value from the Data Source SQL extendable lookup.

If an implementation requires template SQL that differs from the base package, custom extendable lookup values can be created and referenced on the data source business object.

When creating custom template SQL, start from an existing extendable lookup value. Select the value that most closely matches your requirements and use the **Duplicate** function to create a copy, and edit the SQL as needed.

Use the following procedure to add a custom extendable lookup value to a data source business object.

1. Add a business object option of the appropriate type to the data source business object. For instance, if creating custom SQL for aggregation, add an **Aggregation SQL** option.
2. Enter a **Sequence** higher than the base package option.
3. Specify the custom extendable lookup value in the **Option Value** field.

Dynamic Aggregation Measuring Component Types

- Understanding Dynamic Aggregation Measuring Component Types
- Configuring Dynamic Aggregation Measuring Component Types

Understanding Dynamic Aggregation Measuring Component Types

Like other measuring component types, dynamic aggregation measuring component types define the most important properties of a measuring component. In this case, they define the properties of aggregator measuring components.

Measuring component types used by dynamic aggregation use many of the same attributes as other types of measuring components, but some attributes are used differently. Some attributes used specifically by dynamic aggregation include:

Measuring Component Business Object: This should be an aggregator measuring component business object, such as one of the following:

- Aggregator Interval (D1-AggregatorInterval)
- Aggregator Scalar (D1-AggregatorScalar)
- Aggregator Customer Quantities (D1-AggregatorCustomerQuantity)
- Aggregator Items Measuring Component (D1-AggregatorItems)

Measurement Business Object: Aggregated measurements can be based on either the Measurement (D1-Measurement) business object or the Aggregation Measurement (D1-AggregationMeasurement) business object (based on the Aggregation Measurement maintenance object).

Data Source: The source of the data that will be aggregated and the specific type of data that should be aggregated. Note that the **Service Type** of the data source, as determined by its

Copyright © 2026, Oracle and/or its affiliates.

associated objects (e.g. Measuring Component Type, Device Type, Service Point Type), must match the **Service Type** for this measuring component type.

Anchor Time: A time used to define the start and end time of the business day for purposes of aggregation processing for aggregation measuring components of this type. In gas settlement implementations, this is used to define the start and end time for "Gas Day" calculations.

Value Identifiers: These store the values of UOM, TOU, and SQI that identify the measurements to be aggregated. For each value identifier either the **Value Calculation Method** or **Value Derivation Algorithm** must be defined (not both). These define how measurement values are calculated or derived during aggregation processing.

Refer to Understanding Measuring Component Types for more information about measuring component types.

Configuring Dynamic Aggregation Measuring Component Types

The **Measuring Component Type** portal is used to display and maintain dynamic aggregation measuring components.

Refer to Understanding Dynamic Aggregation Measuring Component Types for more information about dynamic aggregation measuring components.

Configuring Measuring Component Types

Aggregation Groups

- Understanding Aggregation Groups
- Configuring Aggregation Groups

Understanding Aggregation Groups

Aggregation Groups define the ordering of a series of related aggregations based on a set of configured Measuring Component Sets. Aggregation groups define the following attributes for aggregations performed for its Measuring Component Sets:

- The **Time Zone** for aggregated data
- **Aggregation Scheduling** can be based on either a Defined Lag, or a Processing Timetable.
  - **Defined Lag** scheduling uses the following parameters:
    - **Aggregation Cut Off Time**: The end time for aggregation calculations performed for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This is used to ensure a consistent end time for aggregation periods. This is especially useful when aggregating other aggregations.
    - **Aggregation Lag**: The number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be

Copyright © 2026, Oracle and/or its affiliates.

for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.

    * **Aggregation Horizon**: The number of days in the aggregation period for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Grou. This reflects the time period during which there's a potential change in measurement data for one or more of the measuring components associated with Aggregation Measuring Components. This together with the Aggregation Lag is used to determine the start and end dates of an aggregation period.

  – **Processing Timetable** scheduling uses a Processing Timetable Type to drive aggregation schedule.

Aggregation groups can be used to perform aggregations for Analytical or Settlement purposes (defined by the Aggregation Category).

- Aggregation Group Runs

Aggregation Group Runs

Aggregation group runs represent individual aggregation executions for an individual aggregation group. Aggregation group runs are used to capture the relevant details for an aggregation and track the status of the execution of the aggregation for each of the Measuring Component Sets for the aggregation group.

Aggregation group runs for a specific aggregation group are listed in the **Aggregation Group Run** zone on the **Aggregation Group** portal. This zone displays the following for each aggregation group run:

- Run Number
- Run Type (Scheduled vs. Ad Hoc)
- Status (Pending, In Progress, Completed, or Error)
- Aggregation Horizon Start
- Aggregation Horizon End
- Details

Details for a specific aggregation group run can be viewed by broadcasting the run in the **Aggregation Group Run** zone. The following is displayed for the selected run in the **Aggregation Group Run Details** zone:

- Sequence
- Measuring Component Set
- Initiation Method
- Batch Run

Aggregation Group Runs and Anchor Time

Aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Aggregation Groups

This portal is used to display and maintain aggregation groups.

Refer to Understanding Aggregation Groups for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, **Aggregation Group**.

The following zones may appear as part of the portal's **Main** tab page:

- **Aggregation Group List**: This zone lists all aggregation group records. Broadcast a record to display the details of the selected record.
- **Aggregation Group**: This zone provides information about a selected aggregation group. The **Record Actions** section of this zone includes the following options:
  - Click **Delete Measurements** to clear the measurements of all measuring component sets of the aggregation group for a specified time range. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "Measurement Only" (MSRMTONLY) Purge Option.
  - Click **Delete All MC Set Data** to clear all data for all measuring component sets of the aggregation group. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "All" (ALL) Purge Option.
  - Click **Delete All Aggregation Runs** to clears all aggregation group runs of the aggregation group. This runs the Clean Up Aggregation Group Runs (D1-AGGRC) batch process.

  See **Batch Processing** below for more information about the batch processes used by these options.
- **Aggregation Measuring Component Sets**: This zone lists measuring component sets associated with the selected aggregation group. Click **Resequence** to change the order of the measuring component sets in the aggregation group. Click **Add** to add a new measuring component set to the aggregation group.
  - Click the icon in the **Delete Measurements** column to delete the measurements of the measuring component set for a specified time range. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "Measurement Only" (MSRMTONLY) Purge Option for the measuring component.
  - Click the icon in the **Delete MC Set Data** column to delete all data of the measuring component set. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "All" (ALL) Purge Option for the measuring component.

  See **Batch Processing** below for more information about the batch processes used by these options.
- **Aggregation Group Run**: This zone lists individual aggregation runs performed for the selected aggregation group. Broadcast a record to display the details of the selected record. This zone displays up to 10 records. If there are more than 10 records, use the **Previous** and **Next** buttons to navigate between pages of records. Click **Add** to execute an ad hoc aggregation group run.
  - Click the icon in the **Delete** column to delete the aggregation group run. This runs the Clean Up Aggregation Group Runs (D1-AGGRC) batch process for the aggregation group run. Not applicable for aggregation group runs currently "In Progress". See **Batch Processing** below for more information about the batch processes used by this option.

Copyright © 2026, Oracle and/or its affiliates.

  - Click the icon in **Cancel** column to cancel an "In Progress" aggregation group run. Applicable only for aggregation group runs currently "In Progress".

- **Aggregation Group Run Details**: This zone displays details for a selected aggregation group run.
  - Click **Clear Batch Runs** to clear any batch runs that have been run. This clears dimension scan batch run and aggregation batch run details.
  - Click the icon in the **Restart** column to restart the aggregation group run. Available only for aggregation group runs in "Error" (see the **Aggregation Batch Run** column)

The following zones may appear as part of the portal's **Settlement Batch Run Statistics** tab page:

- **Settlement Batch Run Statistics**: This zone displays details regarding batch processes run for aggregation groups.

Batch Processing

Several of the options available for maintaining aggregation group runs are based on specific batch processes. This section provides additional details regarding these.

Clean up Aggregation Group (D1-CLAGP)

The Clean up Aggregation Group (D1-CLAGP) batch process can be used to purge aggregation group data and can perform either of the following:

- Clean up the aggregation group's measuring component set(s)' aggregator measuring components and their aggregated measurements
- Clean up a specific measuring component set's aggregator measuring components and their aggregated measurements

Parameters used by this batch process include the following:

- **Restrict by Aggregation Group** (restrictToAggregationGroup): (Required) The code for the aggregation group for which data is purged. This limits the clean up to aggregators linked to this aggregation group's measuring component sets.
- **Restrict by MC Set** (restrictToMCSet): The code for the measuring component set for which data is purged. This limits the clean up to the specific aggregation group's measuring component set. If not provided, all the aggregation group's measuring component sets will be included in the purge process.
- **Purge Option** (purgeOption): (Required) Specifies the mode in which data is purged. Options include:
  - MSRMTONLY: - Purge measurement data only
  - ALL: Purge all measurement data, any settlement items related to the measuring component, any attribute data snapshots related to the settlement item, attribute value related to the Settlement Item, any related measuring component participants (aggregators)
- **Start Date/Time** (startDateTime): The start date and time for the period to be deleted when **Purge Option** is MSRMT_ONLY. This must be provided in the format YYYY-MM-DD-HH.MM.SS.
- **End Date/Time** (endDateTime): The end date and time for the period to be deleted when **Purge Option** is MSRMT_ONLY. This must be provided in the format YYYY-MM-DD-HH.MM.SS.

Clean Up Aggregation Group Runs (D1-AGGRC)

Copyright © 2026, Oracle and/or its affiliates.

The Clean Up Aggregation Group Runs (D1-AGGRC) batch process can be used to purge either a single aggregation group run or all aggregation group runs of an aggregation group. Note this process does NOT purge the corresponding data such as measurements, settlement items, and so. Purging data can be performed using the Clean up Aggregation Group (D1-CLAGP) batch process.

Parameters used by this batch process include the following:

- **Restrict by Aggregation Group** (restrictToAggregationGroup): (Required) The code for the aggregation group for which aggregation group runs are pursed. This limits the clean up to aggregation group runs linked to this aggregation group
- **Restrict by Aggregation Group Run** (restrictToAggregationGroupRun): The code for a specific aggregation group run that will be cleaned up.

Measuring Component Sets

- Understanding Measuring Component Sets
- Configuring Measuring Component Sets
- Configuring Aggregation Criteria Source Types

Understanding Measuring Component Sets

Measuring component sets are used to define the dimensions and criteria by which aggregation will be performed.

A measuring component set's **Measuring Component Type** defines the Measuring Component and Measurement Type to be used in capturing aggregation results, as well as the aggregation functions that will be used on the source transnational data and the Data Source that will identify where that transnational data will be retrieved from.

Measuring component sets are associated with an **Aggregation Group**, which controls when and how aggregation is processed. This includes the sequence in which the aggregation group's measuring component sets are processed, as well as the method by which aggregation processing is initiated for each (this information can be found in the **General Processing** section).

A measuring component set's **Measuring Component Set Class** defines the way in which the measuring component set is used in the aggregation process. Measuring component set classes are based on the measuring component set's business object, and can include:

- **Aggregation Foundation** measuring component sets are used to aggregate transactional data, such as measurements from service points linked via usage subscriptions or billed service quantities.
- **Aggregation Composite** measuring component sets are used to aggregate data from a set of other aggregations (which can be based on either foundation or other composite measuring component sets).
- **Adhoc** measuring component sets are used to aggregate data based on an adhoc, user-defined set of measuring components.

See **Foundation and Composite Aggregation** in Dynamic Aggregation Processing for more information.

Copyright © 2026, Oracle and/or its affiliates.

Dimensions, Criteria, and Individually Managed Items

The **Dimensions** and **Criteria** defined for the measuring component set are used by the dimension scanning process to identify individual aggregation measuring components that should be created. Measuring components are also created for each **Individually Managed Item** for which data matches the item's configuration.

- **Dimensions** identify the attributes by which data should be aggregated. Dimensions can include characteristics, identifiers, market participants, and values from specific tables and columns. For example, aggregating data by loss profile code, service point type and service provider would require defining dimensions for a Loss Profile Code characteristic, the Service Point Type column from the Service Point table, and the Service Provider column from the Device table.
- **Criteria** define the ways in which the dimensions are evaluated through a series of inclusions, exclusions, or both. For example, to aggregate data for the "AGG-E-COM" and "AGG-E-RES" service point types, but NOT for devices from the "ELEC" service provider, criteria could be defined to include both of service point type values and exclude the "ELEC" service provider.
- The criteria provided for **Individually Managed Items** is used to identify specific customers that should be aggregated in isolation from all other data that matches the **Dimensions** and **Criteria** defined for the measuring component set.

More specifically, the dimension scanning template SQL (derived from the data source) is extended by the **Dimensions**, **Criteria**, and **Individually Managed Items** configured on the measuring component set being processed.

Consult the embedded help for more information about the specific configuration options used by **Dimensions**, **Criteria**, and **Individually Managed Items**. The Source Type and Source Entity values are defined in the Aggregation Criteria Source Type (D1-CriteriaSourceTypeLookup) extendable lookup. Refer to Configuring Aggregation Criteria Source Types for more information about this extendable lookup.

The manner in which the criteria provided will be evaluated is defined in formulas configured in the **Criteria Processing** section. Consult the embedded help for more information about defining criteria processing formulas.

Load Distribution

The **Load Distribution** parameters can be used to distribute the load of the transactional data derived from the associated Data Source when generating Measuring Component Set participants. Load distribution is achieved in the system by adding another dimensional attributes on the aggregator measuring component based on a defined number of digits from either the service point ID or usage subscription ID. **Load Distribution** parameters include:

- **Distribute Load**: A flag that indicates ("Yes" or "No") if if the measuring component set's transactional data should be further distributed via an additional dimensional attribute.
- **Number of Digits to Use**: The number of digits from either the service point ID or usage subscription ID to use as an additional dimensional attribute when distributing the load. For example, if **Number of Digits to Use** is 2, the last 2 digits of either the service point ID or usage subscription ID will be analyzed to determine if the transactional data should be a constituent of the aggregator measuring component.
- **Load Distribution Managed By**: Specifies which identifier, either "Service Point" or "Usage Subscription", whose last x digits will be placed as an additional dimensional attribute on the aggregator measuring component.

Copyright © 2026, Oracle and/or its affiliates.

- **Load Distribution Formula**: Identifies if further criteria are needed when applying load distribution. For example, if "postal code" is a dimensional attribute, and we only want to distribute the load for a specific postal code where population is denser, a Criteria can be specified and this formula can be used to identify how that Criteria will be evaluated. This formula is applicable only when the **Distribute Load** parameter is set to "Yes".

Configuring Measuring Component Sets

This portal is used to display and maintain measuring component sets.

Refer to Understanding Measuring Component Sets for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, **Measuring Component Set**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Set**: This zone displays details of a selected measuring component set.
- **Dimensions**: This zone lists the dimensions used to identify the attributes by which the aggregation should be analyzed. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.
- **Criteria**: This zone lists the criteria that limits the data to be aggregated through a series of inclusions, exclusions, or both. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.
- **Individually Managed Items**: This zone lists the criteria used to identify specific customers that should be aggregated in isolation from all other data that matches the dimensions and criteria defined for the measuring component set. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.
- **Measuring Component Set Participants**: This zone displays the measuring components that are in the Measuring Component Set Participant table for the measuring component set. For each measuring component, this zone displays the most recent and last measurement date/times.

Configuring Aggregation Criteria Source Types

The options in the **Source Type** and **Source Entity** drop-down lists used when configuring Dimensions, Criteria, and Individually Managed Items are defined in the Aggregation Criteria Source Type (D1-CriteriaSourceTypeLookup) extendable lookup. The measuring component set business objects reference this extendable lookup.

Each of the base package Source Type values (Characteristic, Identifier, Market Participant, Table, and Entity Group) are records in this extendable lookup. The Source Entity values for each of these are defined in the corresponding record.

For example, the “Table” Source Type defines the base package “Table” Source Entity values (Device, Device Event, Measuring Component, Service Point, Usage Subscription, and Usage), along with criteria path details for each (based on data source class).

If an implementation requires Source Type or Source Entity values that differ from the base package, custom extendable lookup values can be created.

Copyright © 2026, Oracle and/or its affiliates.

When for creating custom values for this extendable lookup, start from an existing extendable lookup value. Select the value that most closely matches your requirements and use the **Duplicate** function to create a copy, and edit as needed.

Configuring Dynamic Aggregation Step-by-Step

This section outlines the steps involved when configuring dynamic aggregation.

Create Device and Service Point Administrative Data

Many of the objects used in dynamic aggregation reference other device- and service point-related administration data, such as measuring component types, service point types, usage subscription types, etc. Make sure this data is in place before configuring entities used with dynamic aggregation.

Create Data Sources

Create data sources for each source of data from which data will be aggregated. Note that implementations can have multiple data sources of each class with different data type configuration (Measuring Component Types, Item Types, Usage Subscription Types, etc.). For example, an implementation with 15 measuring component types might want to aggregate data differently based on subsets of measuring component types. In this case, each subset of measuring component types could be defined on a different data source.

If applicable, create custom Template SQL extendable lookup values and add them to Data Source business objects as appropriate.

Create Aggregation Measuring Component Types

Create the aggregation measuring component types that will be used to define the attributes of the measuring components that will be created during aggregation processing.

Aggregation measuring component types reference an appropriate Data Source. Continuing the example above, each subset of measuring component types (each defined on a separate data source) would require a separate aggregation measuring component type.

Create Aggregation Groups

Create the aggregation groups that will be used to control aggregation processing, including aggregation groups for both foundation and composite aggregation processing.

Note: When initially creating aggregation groups, define only the basic data for each (name, category, and optional time zone). Add measuring component sets to aggregation groups when creating the latter in the next step.

Create Measuring Component Sets

Create the measuring component sets that will define the “buckets” by which the aggregated data will be grouped.

Measuring component sets reference aggregation measuring component types and aggregation groups.

Create custom Source Type records (if applicable).

Copyright © 2026, Oracle and/or its affiliates.

Revise Aggregation Groups (if needed)

After creating measuring component sets, adjust the sequence of each within its aggregation group as appropriate. For example, this would be necessary if measuring component sets are created and added to aggregation groups in an order that differs from their intended sequence.

Extending Dynamic Aggregation

The dynamic aggregation functionality provides extensive flexibility through configuration of data sources, measuring component types, and measuring component sets to handle complex aggregation requirements. However, some implementations may require functionality and configuration options beyond that provided in the base package. For instance, an implementation may need to aggregate data of a source and type not covered by the base package data sources or use different dimensions and criteria or queries for aggregation processing.

In circumstances like these, the dynamic aggregation functionality can be extended through use of the Oracle Utilities Application Framework's Configuration Tools (see Configuration Tools) and other options. This section outlines some of the possible approaches to extending this functionality.

Data Source Business Objects: If an implementation requires aggregation of a data type of source beyond those provided with the base package data source business objects, custom business objects can be created to support this requirement. Key considerations when creating custom data source business objects include:

- Configuration options such as "Sourced Measuring Component Types" or "Sourced Usage Subscription Types"
- Values for the "Aggregation SQL", "Dimension Scanner SQL", and "Find Constituents SQL" business object options from the Data Source SQL extendable lookup.
- Algorithms for the "Generate SQL For Data Source" system event that leverage the configuration options defined for the data source

Data Source SQL Extendable Lookup Values: If an implementation requires template SQL that differs from the base package, custom values for the Data Source SQL extendable lookup can be created and referenced on the data source business object. When creating custom template SQL, start from an existing extendable lookup value. Select the value that most closely matches the requirements and use the **Duplicate** function to create a copy, and edit the SQL as needed.

Aggregation Measuring Component Type Business Objects: If an implementation requires capturing data for measuring component types not provided in the base package, the base package aggregation measuring component type business objects can be extended as appropriate. Key considerations when creating custom measuring component type business objects include:

- Algorithms for the "Generate SQL For MC Type" system event that leverages the Value Identifiers and other information defined for the measuring component type

Aggregation Measuring Component Business Objects: If an implementation requires capturing data for measuring components not provided in the base package, the base package aggregation measuring component business objects can be extended as appropriate. Key considerations when creating custom measuring component business objects include:

- Algorithms for the "Generate SQL For Aggregator MC" system event that leverages the Value Identifiers and other information defined for the measuring component type

Copyright © 2026, Oracle and/or its affiliates.

- Algorithms for the “Find Constituents” system event that uses the “Generate SQL for Aggregator MC” algorithm, data source configuration, and other configuration to find constituent data for aggregation
- Algorithms for the business object’s “Active” state, including data aggregation and value derivation

Measuring Component Set Business Objects: If an implementation requires capturing data for measuring component sets not provided in the base package, the base package aggregation measuring component set business objects can be extended as appropriate. Key considerations when creating custom measuring component set business objects include:

- Algorithms for the “Generate SQL For MC Set” system event
- Algorithms for the business object’s lifecycle states, as appropriate

Aggregation Criteria Source Type Extendable Lookup Values: If an implementation requires Source Type or Source Entity values for measuring component sets that differ from the base package, custom extendable lookup values can be created. When creating custom values for this extendable lookup, start from an existing extendable lookup value. Select the value that most closely matches the requirements and use the **Duplicate** function to create a copy, and edit as needed.

Consumption Synchronization

- Configuring Consumption Synchronization

Configuring Consumption Synchronization

> **Note**
>
> Refer to Introduction to Consumption Sync for additional functional information about how consumption synchronization works.

Keeping consumption synchronized between two measuring components that meter the same quantity but at different frequencies is a complicated task. As such the configuration for this process is diffuse and requires settings across several key areas of the system to be aligned. This section will help guide you through the process of configuring consumption synchronization.

The consumption synchronization process is really a collection of processes within Oracle Utilities Meter Data Management that all work together to ensure that quantities between related channels remain consistent:

- Estimation VEE Rules: several rules align consumption between related channels. These rules allow estimations to be refined based on higher quality measurements from a related channel. These rules are core to the consumption synchronization process. In simple scenarios where a few intervals are missing for an interval measuring component these rules are all that are necessary for synchronization with the related channel (note: it requires that the related channels data for that same time period has already been processed through to final measurements). For more information on these rules please visit the following sections:
  - Interval Adjustment from Scalar

Copyright © 2026, Oracle and/or its affiliates.

- Scalar Calculation from Interval
- Scalar Proration
- Sum Check
- Final Measurement Validation

> **Note**
>
> Refer to About IMD Estimations for more information about these rules

- Periodic Estimation: both the interval and scalar variations of periodic estimation play an important role in consumption synchronization by ensuring that both channels of data are without gaps. In simple scenarios where one channel is missing data and the other is not periodic estimation is all that is require to produce synchronized consumption. It is important to note that this process itself does not perform estimations but rather it is responsible for creating estimation initial measurements to trigger the estimation VEE rules.
- Consumption Synchronization Activities: these activities work to fix alignment issues that occur when data for the related channels are processed out of order or when there are complex outage scenarios. These activities are created when higher quality data is received for one channel and the related channel has measurements that are eligible to be recalculated and adjusted to align the total consumption for the period across the two channels. It is important to note that this process itself does not perform estimations but rather it is responsible for creating estimation initial measurements to trigger the estimation VEE rules.

Device Configuration Type

As a default the system will not generate consumption synch activities for related measuring components. Turning on consumption synchronization activities is done through a few key fields on the device configuration type:

- **Keep Consumption Reference MC in Sync**: defines if the related measuring components should be kept in sync. It can be configured to provide a one way synchronization from primary to secondary or a two way synchronization between primary and secondary.
- **Minimum Condition to Sync Primary MC**: when the secondary measuring component can initiate synchronization of the primary this provides an ability to limit those situations to when the incoming data is of a minimum quality.
- **Sum Check VEE Exception Type**: provides further ability to limit initiation of synchronization. When configured the synchronization activities will only be created when the initial measurement being processed encounters a VEE exception of the type configured. More specifically, a sum check VEE exception which will indicate that the two channels are out of synch by a minimum tolerance amount. This can be used to avoid synchronization either when there is no difference between the channels or when there is only a small difference between the channels.
- **Enable Extended Reconciliation**: defines if synchronization calculations should extend to cover the total contiguous period of consumption synchronization eligible final measurements across both channels in the relationship. When set to “Yes” consumption synchronization algorithms on initial measurements will execute a measuring component relationship level gap analysis and create a consumption synchronization for an entire gap that will impact both measuring components in the relationship.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail about these fields can be found in the embedded help for the device configuration type.

Register Auto-Read Measuring Component Type

The configuration available for the register auto-read measuring component type has an indirect impact on the consumption synchronization process. This configuration is primarily intended to allow for a register to re-evaluate previously created estimations when new more accurate readings are received even when no related measuring components exist. Where this has impact to the consumption synchronization process is that when a new scalar reading is received after a period of estimation it will result in the time period for that new initial measurement being expanded into the past. This is because any estimates prior to this new higher quality initial measurement will be logically removed and the start reading for the initial measurement will in turn be the last non-estimated measurement prior to the initial measurement. This creates an initial measurement that spans the entire period for which estimations exist and as a result when a consumption synchronization activity is generated it will result in that same period of time being re-evaluated on the related measuring component.

This functionality is controlled by a few key fields on the register auto-read measuring component type:

- **Ignore Estimates as IMD Start Reading**: controls whether estimates directly previous to newly received incoming initial measurement should be logically removed when that newly received initial measurement data is non-estimated.
- **Flag Future Estimates as Do Not Use**: controls whether estimates that come directly after newly received incoming initial measurements should be logically removed when that newly received initial measurement data is non-estimated
- **Actuals or Corrections Initial Re-Estimation:** works in tandem with the above two fields. When either scenario results in measurements being logically removed this field, when turned on, will reset the last contiguous measurement date/time for the measuring component making it eligible once more for periodic estimation. With the end result being that those estimated measurements that were removed would recreated as new estimates.

Measuring Components

In order for consumption synchronization to work it must know that two measuring components are related in a way that indicates they are measuring the same consumption. This is achieved by configuring the "Consumption Reference Measuring Component" field on the measuring component.

Not only does this establish the relationship but it also establishes which measuring component is considered to be primary and which is considered to be secondary.

A measuring component is considered to be secondary when it holds the relationship to the Consumption Check Measuring Component (the primary measuring component).

For example, consider the following related measuring components.

- Scalar Measuring Component: ER-SM-007 / 1 / Electric kWh Daily
- Interval Measuring Component: ER-SM-007 / 2 / Electric kWh 60min

If the scalar measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the interval measuring component

Copyright © 2026, Oracle and/or its affiliates.

specifies the scalar measuring component as the "Consumption Reference Measuring Component".

If the interval measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the scalar measuring component specifies the interval measuring component as the "Consumption Reference Measuring Component".

It is important to note that pending initial measurements for the secondary measuring component are processed by the D1-IMD batch process after the initial measurements for the primary measuring component. This ensures that the primary measuring component's final measurements will be available to the secondary measuring component's initial measurements VEE process to provide better quality estimations and validations.

Initial Load and Manual Initial Measurement Algorithms

Key to the consumption synchronization process are the algorithms that reside on initial load and manual initial measurements for both scalar and interval measuring components. These algorithms contain logic to identify final measurements on a related channel that are consumption synchronization eligible (typically estimated).

These algorithms allow definition of:

- The consumption synchronization activity created
- The condition range that defines consumption synchronization eligible final measurements

Please refer to the algorithm type descriptions for more information:

Algorithm Type	Description	Where used
D1-UPD-DTMC	Update Latest Measurement Date/Time on MC with Consumption Sync	D1-InitialLoadIMDInterval D1-ManualIMDInterval
D1-UDTSCMCRE	Update Latest Measurement Date/Time on Scalar MC with Consumption Sync	D1-InitialLoadIMDScalar D1-ManualIMDScalar

Consumption Synchronization Activities

The consumption synchronization activities are initiated by initial measurements and are responsible for generating the appropriate estimation initial measurements to re-evaluate any consumption synchronization eligible final measurements for the related measuring component being processed.

These activities can be associated to one or more initiating initial measurements and are able to handle a broad combination of time periods which are not required to be contiguous.

If any generated estimation initial measurement does not finalize processing will be halted and details about the failed estimation initial measurement will be provided.

The following table identifies the catalogue of consumption synchronization activities:

Activity Type	Description	Activity Business Object

Copyright © 2026, Oracle and/or its affiliates.

Related MC Consumption Sync - Scalar	This activity is generated to re-evaluate consumption sync eligible final measurements on scalar measuring components. One estimation initial measurement will be created per measurement needing to be re-evaluated.	D1-RelMCREScalar
Related MC Consumption Sync - Interval	This activity is generated to re-evaluate consumption sync eligible final measurements on interval measuring components. One estimation initial measurement will be created per contiguous set of interval measurements needing to be re-evaluated. Note the generated initial measurements will include non-consumption sync eligible final measurements which will help to feed into the estimation process to provide more accurate results.	D1-RelMCREInterval
Gap Period Consumption Sync	This activity will re-evaluate both measuring components in the relationship for a wider time period than the initiating initial measurement. Specifically the time period will be the total contiguous period of consumption sync eligible final measurements across both channels in the relationship.	D1-GapPeriodConsumpnSync

Each consumption synchronization activity has an algorithm that performs the core logic of the consumption synchronization process.

These algorithms allow definition of:

- The condition range that defines consumption synchronization eligible final measurements

Please refer to the algorithm type descriptions for more information:

Algorithm Type	Description	Where used
D1-RE-SCMC	Scalar MC Consumption Sync	D1-RelMCREScalar
D1-RE-INTVMC	Interval MC Consumption Sync	D1-RelMCREInterval
D1-GAPCSYNC	Gap Period Consumption Sync	D1-GapPeriodConsumpnSync

Periodic Estimation

The algorithm that initiates periodic estimation which is plugged in to the Smart Meter device business object has a key configuration setting related to consumption synchronization:

Copyright © 2026, Oracle and/or its affiliates.

- **MC Type to Process First**: determines whether interval or scalar measuring components should be estimated first. This should be set to process whichever measuring component type is considered to be primary first.

Please refer to the algorithm type description for more information:

Algorithm Type	Description	Where used
D1-PERESTM	Periodic Estimation	D1-SmartMeter

Related Batch Controls

There are a few batches involved with Consumption Synchronization:

- **Related MC Consumption Sync** (D1-RMCRE): this processes any pending consumption synchronization activities.
- **Related MC Consumption Sync - Retry Act** (D1-RMCRR): this retries any consumption synchronization activities that fail to the Issue Detected state.

Defining Dashboard Options

- Configuring the Customer Operational Dashboard
- Configuring the Meter Operational Dashboard
- Configuring the System Health Check Dashboard
- Configuring the Service Order Management Dashboards
- Configuring the Product Use Metrics Dashboard

Configuring the Customer Operational Dashboard

This section describes the process for configuring the Customer Operational Dashboard.

Refer to the Using the Customer Operational Dashboard section of the *Oracle Utilities Customer To Meter Business User Guide* for more information on this functionality.

Configuration for the Customer Operational Dashboard is performed by adding or changing a Master Configuration. You can access the portal from the **Admin > General > Master Configuration**.

Once the Master Configuration search screen returns, configure the **Customer Operational Dashboard Configuration** record. Use the Add button beside the record to configure for the first time. If a record has already been added, then click the Edit button instead. Use the embedded help to guide you through the meaning of each configuration field.

This dashboard also leverages a method of pre-staging data known as Statistics Snapshots. Refer to About Statistics in the *Oracle Utilities Application Framework Administrative User Guide* for more information.

A method for tracking Performance Targets (also known as Service Level Agreements) is also available in the Customer Operational Dashboard. To understand how to set up Performance Targets, refer to About Performance Targets in the *Oracle Utilities Application Framework Administrative User Guide*. The Performance Targets you configure will then be leveraged on the **Batch Performance** page of the Customer Operational Dashboard.

Copyright © 2026, Oracle and/or its affiliates.

Configuring the Meter Operational Dashboard

This section describes the process for configuring the Meter Operational Dashboard.

Refer to the Using the Meter Operational Dashboard section of the *Oracle Utilities Meter Solution Business User Guide* for more information on this functionality.

This dashboard leverages a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on Statistics Snapshots.

A method for tracking Performance Targets (also known as Service Level Agreements) is available within the Meter Operational Dashboard as well. To understand how to set up Performance Targets, refer to Understanding Performance Targets. The Performance Targets you configure will then be leveraged on the **Batch Performance** and/or **Database** tabs of the Meter Operational Dashboard.

Configuring the **Meter Operational Dashboard** includes the following:

Master Configuration

You can access the Master Configuration portal by selecting **Admin**, then **General**, then **Master Configuration**.

Select the MDM Operational Dashboard Configuration.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

General Admin Data

Create the following administrative data.

Service Type:

- **Service Type**: MISC
- **Description**: Miscellaneous

Unit of Measure:

- **Unit of Measure**: CT
- **Description**: Count
- **Service Type**: Miscellaneous
- **Decimal Positions**: 0
- **Allowed on Measuring Component**: Yes
- **Measures Peak Quantity**: No
- **Magnitude**: 1
- **Base Unit of Measure**: BLANK

Activity Types for Payload Statistics

Configure the following payload statistics activity types:

- Payload Statistics

Copyright © 2026, Oracle and/or its affiliates.

- Payload Error Notification
- Payload Summary

See Configuring Activity Types for more information.

See Processing Statistics in the *Business User Guide* for more information about payload statistics processing.

Measuring Component Types for Payload Statistics

Configure the following measuring component types:

Initial Measurements

Field / Section	Value
Measuring Component Type	STATS-IMD
Description	IMD Upload Statistics Aggregator
Measuring Component Type Business Object	Activity Upload Statistic Aggregator Type
Measuring Component Business Object	Payload Statistics Aggregator - IMD
Measurement Business Object	Measurement
Service Type	Miscellaneous
Statistic Type	Initial Measurements
Display Parcel Size	24:00:00
Aggregation	Aggregation Horizon: 5 Aggregation Lag: 2
Valid Activity Types to Aggregate	Activity Type: Payload Statistics
Value Identifiers	

Value Identifier Type	Short Hand Description	Unit of Measure
Measurement	Total Number of IMDs	Count
Value 01	Count of Outstanding IMDs	Count
Value 02	Count of IMDs in Error	Count
Value 03	Count of Pending IMDs	Count
Value 04	Count of Completed IMDs	Count

Display Configuration	Hours of Chart Data to Display: 24

Device Events

Field / Section	Value
Measuring Component Type	STATS-EVENT
Description	Event Upload Statistics Aggregator
Measuring Component Type Business Object	Activity Upload Statistic Aggregator Type

Copyright © 2026, Oracle and/or its affiliates.

Field / Section	Value
Measuring Component Business Object	Payload Statistics Aggregator - Event
Measurement Business Object	Measurement
Service Type	Miscellaneous
Statistic Type	Events
Display Parcel Size	24:00:00
Aggregation	Aggregation Horizon: 5 Aggregation Lag: 2
Valid Activity Types to Aggregate	Activity Type: Payload Statistics
Value Identifiers	

Value Identifier Type	Short Hand Description	Unit of Measure
Measurement	Total Number of Events	Count
Value 01	Count of Outstanding Events	Count
Value 02	Count of Events in Error	Count
Value 03	Count of Pending Events	Count
Value 04	Count of Completed Events	Count

Display Configuration	Hours of Chart Data to Display: 24

Statistic Controls and Statistics Snapshots

Statistic Controls are the source for recording the transactions which have gone into exception/ discarded state or have a pending to-do associated with them. These records are then fetched by MDM Operational Dashboard's respective zones for Statistic and Trend display.

Statistic Snapshots capture the transactions which have gone into exception/discarded state or have a pending to-do associated with them at a specific time.

Statistic Controls are configuration records that record the daily transactions of Incomplete IMD by Status, VEE Exceptions by Type, Usage Transaction by Status, and Usage Transaction by To-Do.

Statistic Snapshots are captured daily via the Statistics Control Monitor (F1-STATS) batch process.

Configure the following Statistics Control records:

Statistics Control	Description	Category	Status	Snapshot Frequency Hours	Retention Mode	Snapshot Business Object
INCOMPLETE-IMD-STATUS	Incomplete IMDs by Status	Operational	Active	1.00	Retain All Snapshots	Same as Description

Copyright © 2026, Oracle and/or its affiliates.

Statistics Control	Description	Category	Status	Snapshot Frequency Hours	Retention Mode	Snapshot Business Object
INCOMPLETE-UT-STATUS	Incomplete Usage Transaction by Status	Operational	Active	1.00	Retain All Snapshots	Same as Description
INCOMPLETE-UT-TD-EXCP	Incomplete Usage To Dos by Exception	Operational	Active	1.00	Retain All Snapshots	Same as Description
INCOMPLETE-VEE-TD-EXCP	Incomplete VEE To Dos by Exception	Operational	Active	1.00	Retain All Snapshots	Same as Description

Payload Statistics Aggregator Measuring Components

Create a measuring component for each head end system for each measuring component type. For example, assume the following head end systems are being used:

- MV90: Usage
- Landis+Gyr: Usage and Device Events

In this example, you would create the following measuring components:

- MV90 (STATS-IMD Measuring Component Type)
- L+G (STATS-IMD Measuring Component Type)
- L+G (STATS-EVENT Measuring Component Type)

Field	MV90 Usage	L+G Usage	L+G Device Events
Measuring Component Type	IMD Upload Statistics Aggregator	IMD Upload Statistics Aggregator	Event Upload Statistics Aggregator
Status	Active	Active	Active
How To Use	Additive	Additive	Additive
Head End System	MV90	L+G	L+G

Batch Processing

The Meter Operational Dashboard is configured to be refreshed daily. To refresh the data used in the dashboard, the following batch processes should be processed in sequence after daily payload processing from the head end systems is complete.

- **D1-PSACC** (Payload Statistics Accumulation - Monitor): Used to initiate aggregation and accumulation of payload statistics in payload statistics activities. This executes the Perform Activity Aggregation (D1-PRFACTAGG) algorithm in the activity's "Active" state and transitions the activity to the "Accumulate Statistics" state which executes the Accumulate Statistics Algorithm (D1-ACCSTATS) algorithm.

  Use the following parameters when running this batch process:

  – Measuring Component Maintenance Object (maintenanceObject): D1-ACTIVITY

  – Restrict by Batch Code (isRestrictedByBatchCode): true

Copyright © 2026, Oracle and/or its affiliates.

- **D1-MC** (MC MO Periodic Monitor Process): Used to initiate aggregation of statistics for statistics aggregator measuring components. This transitions measuring components from the "Active" state to the "Aggregate" state, which executes the Aggregate IMD Statistics (D1-AGIMDSTAT) or Aggregation Event Statistics (D1-AGEVTSTAT) algorithms that aggregate measurement and/or device event statistics.

  Use the following parameters when running this batch process:

  – Maintenance Object (maintenanceObject): D1-MEASRCOMP

- **F1-STATS** (Statistics Control Monitor): Used to generate Statistic Snapshots based on Statistic Controls.

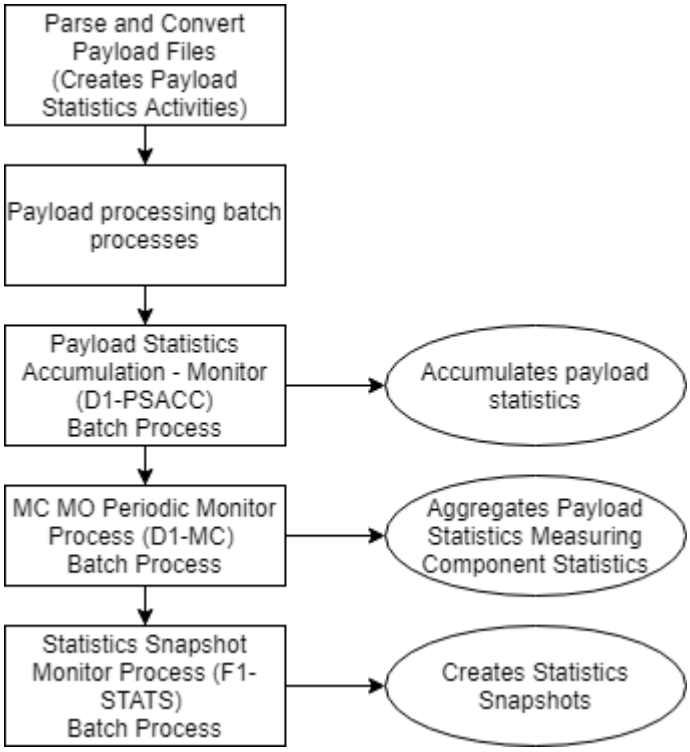
  Use the following parameters when running this batch process:

  – Maintenance Object (maintenanceObject): F1-STATS

  – Restrict by Business Object (restrictToBusinessObject): F1-Statistics

  – Restrict by Status Code (restrictToBOStatus): ACTIVE

The flowchart below illustrates the order which payload processing and these batch processes should be executed.



Populating the Dashboard Zones

The table below outlines the source of data for the various zones in the Meter Operational Dashboard.

Dashboard Zone	Source of Data
IMD Loading Trend	Usage Payload Statistics Aggregator Measuring Components (which aggregate data from payload statistics activities) (MV90 Usage and L+G Usage measuring components from the example above)

Copyright © 2026, Oracle and/or its affiliates.

Dashboard Zone	Source of Data
Device Event Loading Trend	Device Event Payload Statistics Aggregator Measuring Components (which aggregate data from payload statistics activities) (L+D Device Events measuring component from the example above)
Incomplete IMDs by Status Incomplete IMDs Trend	Incomplete IMDs by Status Statistics Control and Statistics Snapshots
VEE To Dos by Exception Type VEE To Dos Trend	Incomplete VEE To Dos by Exception Statistics Control and Statistics Snapshots
Usage Transaction by Status Usage Transactions Trend	Incomplete Usage Transactions by Status Statistics Control and Statistics Snapshots
Incomplete Usage To Dos by Exception Usage Transactions To Dos Trend	Incomplete Usage To Dos by Exception Statistics Control and Statistics Snapshots

Configuring the System Health Check Dashboard

This section describes the process for configuring the **System Health Check** tab of the **Meter Operational Dashboard**.

Refer to Using the System Health Check Tab in the *Business User Guide* for more information on this functionality.

This dashboard leverages a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on statistics snapshots.

Configuring the**System Health Check** tab includes the following:

Master Configuration

To access the **Master Configuration** portal select **Admin**, select **General**, and select **Master Configuration**.

Select the **System Health Check Configuration** master configuration.

You can also access this master configuration by clicking **Configure** on the **System Health Check** tab on the **Meter Operational Dashboard**.

Click the **Add** icon beside the record to configure for the first time. If a record has already been added, then click the **Edit** icon instead. Use the embedded help to guide you through the meaning of each configuration field.

This master configuration includes the following options:

- **IMD Trend Days**: specifies the horizon of collected data for IMD Processing and IMD Quality statistics snapshots. The number of days captured in each snapshot. Default 30 days.
- Chart Options:
  - **Processing**: Allows configuration of the colors used for IMD Records, Average Measurements, and Maximum Measurements data on the **IMD Processing Trend** graph.

Copyright © 2026, Oracle and/or its affiliates.

- **Quality**: Allows configuration of the colors used for VEE Exceptions, Average Measurements, and Maximum Measurements data on the **IMD Quality Trend** graph.

Configure Statistic Control and Statistic Snapshots

Statistic controls are the source for capturing the numbers of initial measurement records, VEE exceptions, and average and maximum measurements displayed on the **Meter Health Check** tab.

Statistic snapshots capture the numbers for each metric. Statistic controls are configuration records that define the specific types of metrics to capture.

Statistic snapshots are captured periodically via the F1-STATS (Statistics Control Monitor) batch program.

Configure the following Statistics Control records:

Statistics Control	Description	Category	Status	Snapshot Frequency Hours	Retention Mode	Snapshot Business Object
Processing	IMD Processing Statistics - Hourly	Operational	Active	24:00	Retain All Snapshots	IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly)
Quality	IMD Quality Statistics - Hourly	Operational	Active	24:00	Retain All Snapshots	IMD Quality Snapshot - Hourly (D1-IMDQuality Hourly)
IMD Control Seeder	IMD Control Seeder - Hourly	Operational	Active	24:00	Retain All Snapshots	IMD Control Seeder – Hourly (D1-IMDCtrlSeeder)
IMD Control	IMD Control - Hourly	Operational	Active	24:00	Retain All Snapshots	IMD Control – Hourly (D1-IMDCtrl)

See Capturing Statistics for more information about statistics.

Configure Statistics Control Business Object

The Statistics Control (F1-Statistics) business object should be configured to specify the "Statistics Control Monitor" batch control in the **Monitor Process** drop-down list on the "Active" state.

To configure this business object, do the following:

1. Open the Statistics Control (F1-Statistics) business object in the **Business Object** portal.
2. Navigate to the **Lifecycle** tab, and expand the "Active" state.
3. Select "Statistics Control Monitor" from the **Monitor Process** drop-down list.
4. Click **Save**.

Copyright © 2026, Oracle and/or its affiliates.

Batch Processing

The **System Health Check** tab is configured to be refreshed regularly (as frequently as daily). To refresh the data used in the dashboard, the F1-STATS batch process should be processed in sequence after other regularly schedule batch processing is complete to generate statistic snapshots based on statistic controls.

Use the following parameters when running this batch process:

- Maintenance Object (maintenanceObject): F1-STATS
- Restrict by Business Object (restrictToBusinessObject): F1-Statistics
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Configuring the Service Order Management Dashboards

This section describes the process for configuring the **Service Order Operational Dashboard** and the **Service Order Trends Dashboard**.

Refer to the Using the Service Order Operational Dashboard and Using the Service Order Trends Dashboard sections of the *Business User Guide* for more information on this functionality.

Configuring the **Service Order Operational Dashboard** and **Service Order Trends Dashboard** includes the following:

Configure Activity Type

Configure the following payload statistics activity type:

- Activity Statistics Aggregator Creator

Use the following procedure to create this activity type:

1. Select **Admin**, then **Communication**, then **Activity Type**. The **Activity Type** portal and **Activity Type List** zone open.
2. Click the **Add/Edit** icon for the "Activity Statistics Aggregator Creator" activity type in the **Activity Type List** zone.
3. Specify the following:
   - **Create Aggregators for Field Activity Statistics**: Yes
   - **Create Aggregators for Command Activity Statistics**: Yes
   - **Activity-Requesting Service Providers**: The external system(s) that sends service requests
   - **Exception Handling**: **To Do Type**, **To Do Role**, and other values as appropriate for your implementation.
4. Click **Save**.

See **Configuring Activity Types** for more information.

Configure Measuring Component Types

Configure the following measuring component types for each service type in your implementation:

- Activity Processing Accumulation Aggregator

Copyright © 2026, Oracle and/or its affiliates.

- Activity Aging Snapshot Aggregator
- Activity Processing Snapshot Aggregator
- Activity Processing Derived Snapshot Aggregator

The tables below provide details for these measuring component types based on the "Electric" service type from the demonstration database.

Sub-Aggregator Measuring Component Types:

Field	Activity Processing Snapshot Aggregator	Activity Processing Derived Snapshot Aggregator
Measuring Component Type	ACT_PROC_SNAP_AGG	ACT_PROC_DER_SNAP_AGG
Description	Activity Processing Snapshot Aggregator	Activity Processing Derived Snapshot Aggregator
Measuring Component Type Business Object	Activity Statistics Sub-Aggregator Type	Activity Statistics Sub-Aggregator Type
Measuring Component Business Object	Activity Processing Snapshot Aggregator	Activity Derived Snapshot Aggregator
Measurement Business Object	Measurement	Measurement
Service Type	Electric	Electric
Interval Size	24:00:00	24:00:00
Fallback Time Zone	US Eastern Time	US Eastern Time
Consumptive/Subtractive	Consumptive	Consumptive
Activity Aggregator Type	Activity Processing Snapshot	Activity Processing Derived Snapshot
Aggregation	Aggregation Horizon: 7 Aggregation Lag: 2 Aggregation Cut-Off Time: 12:00:00	Aggregation Horizon: 7 Aggregation Lag: 1 Aggregation Cut-Off Time: 12:00:00

Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field</th><th>Activity Processing Snapshot Aggregator</th><th>Activity Processing Derived Snapshot Aggregator</th></tr>
<tr><td>Value Identifiers</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Created</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Discarded</td><td>Count</td></tr>
</table>
</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Total</td><td>Count</td></tr>
<tr><td>Value 10</td><td>Discarded</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error Total</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Error in Field</td><td>Count</td></tr>
<tr><td>Value 04</td><td>Error in Command</td><td>Count</td></tr>
<tr><td>Value 05</td><td>Error in Orchestrator</td><td>Count</td></tr>
<tr><td>Value 06</td><td>In Progress Total</td><td>Count</td></tr>
<tr><td>Value 07</td><td>In Progress in Field</td><td>Count</td></tr>
<tr><td>Value 08</td><td>In Progress in Command</td><td>Count</td></tr>
<tr><td>Value 09</td><td>In Progress in Orchestrator</td><td>Count</td></tr>
</table>
</td></tr>
<tr><td>Sub-Aggregator Types</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Display Configuration</td><td>Hours of Chart Data to Display: 480
Default Measurement Condition: Regular</td><td>Hours of Chart Data to Display: 480
Default Measurement Condition: Regular</td></tr>
</table>

Master Aggregator Measuring Component Types:

Field	Activity Processing Accumulation Aggregator	Activity Aging Snapshot Aggregator
Measuring Component Type	ACT_PROC_ACC_AGG	ACT_AGN_SNAP_AGG
Description	Activity Processing Accumulation Aggregator	Activity Aging Snapshot Aggregator

Copyright © 2026, Oracle and/or its affiliates.

Field	Activity Processing Accumulation Aggregator	Activity Aging Snapshot Aggregator
Measuring Component Type Business Object	Activity Statistics Master Aggregator Type	Activity Statistics Master Aggregator Type
Measuring Component Business Object	Activity Processing Accumulation Aggregator	Activity Aging Snapshot Aggregator
Measurement Business Object	Measurement	Measurement
Service Type	Electric	Electric
Interval Size	24:00:00	24:00:00
Fallback Time Zone	US Eastern Time	US Eastern Time
Consumptive/Subtractive	Consumptive	Consumptive
Activity Aggregator Type	Activity Processing	Activity Aging Snapshot
Aggregation	Aggregation Horizon: 5 Aggregation Lag: 2 Aggregation Cut-Off Time: 12:00:00	Aggregation Horizon: 5 Aggregation Lag: 2 Aggregation Cut-Off Time: 12:00:00

Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field</th><th>Activity Processing Accumulation Aggregator</th><th>Activity Aging Snapshot Aggregator</th></tr>
<tr><td>Value Identifiers</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Activity Count</td><td>Count</td></tr>
<tr><td>Value 10</td><td>Discarded Count</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed Count</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error Count</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Error in Field</td><td>Count</td></tr>
<tr><td>Value 04</td><td>Error in Command</td><td>Count</td></tr>
<tr><td>Value 05</td><td>Error in Orchestrator</td><td>Count</td></tr>
<tr><td>Value 06</td><td>In Progress Count</td><td>Count</td></tr>
<tr><td>Value 07</td><td>In Progress in Field</td><td>Count</td></tr>
<tr><td>Value 08</td><td>In Progress in Command</td><td>Count</td></tr>
<tr><td>Value 09</td><td>In Progress in Orchestrator</td><td>Count</td></tr>
</table>
</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Average Completion Time</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed Within Tolerance</td><td>Count</td></tr>
<tr><td>Value 02</td><td>12-19 Hours Late</td><td>Count</td></tr>
<tr><td>Value 03</td><td>20-27 Hours Late</td><td>Count</td></tr>
<tr><td>Value 04</td><td>28+ Hours Late</td><td>Count</td></tr>
</table>
</td></tr>
<tr><td>Sub-Aggregator Types</td><td>
<table>
<tr><th>Activity Aggregator Type</th><th>Measuring Component Type</th></tr>
<tr><td>Activity Processing Derived Snapshot</td><td>Activity Processing Derived Snapshot Aggregator</td></tr>
<tr><td>Activity Processing Snapshot</td><td>Activity Processing Snapshot Aggregator</td></tr>
</table>
</td><td>N/A</td></tr>
</table>

Copyright © 2026, Oracle and/or its affiliates.

Field	Activity Processing Accumulation Aggregator	Activity Aging Snapshot Aggregator
Display Configuration	Hours of Chart Data to Display: 720	Hours of Chart Data to Display: 480
	Default Measurement Condition: Regular	Default Measurement Condition: Regular

Create Activity Aggregator Measuring Components

Create an activity aggregator measuring component for each activity type you wish to capture for each measuring component type you created above. For example, if you wanted to capture statistics for request orchestration activity types (such as Enable Service, Disable Service, and so on) for the above measuring component types, you would create a total of 24 measuring components, as follows:

Measuring Component Type	Aggregation Dimensions — Activity Type	Aggregation Dimensions — Service Type
Activity Processing Accumulation Aggregator	Back to Back Orchestrator	Electric
Activity Aging Snapshot Aggregator	Back to Back Orchestrator	Electric
Activity Processing Snapshot Aggregator	Back to Back Orchestrator	Electric
Activity Processing Derived Snapshot Aggregator	Back to Back Orchestrator	Electric
Activity Processing Accumulation Aggregator	Cut for Non Payment Orchestrator	Electric
Activity Aging Snapshot Aggregator	Cut for Non Payment Orchestrator	Electric
Activity Processing Snapshot Aggregator	Cut for Non Payment Orchestrator	Electric
Activity Processing Derived Snapshot Aggregator	Cut for Non Payment Orchestrator	Electric
Activity Processing Accumulation Aggregator	Disable Service	Electric
Activity Aging Snapshot Aggregator	Disable Service	Electric
Activity Processing Snapshot Aggregator	Disable Service	Electric
Activity Processing Derived Snapshot Aggregator	Disable Service	Electric
Activity Processing Accumulation Aggregator	Enable Service	Electric
Activity Aging Snapshot Aggregator	Enable Service	Electric
Activity Processing Snapshot Aggregator	Enable Service	Electric
Activity Processing Derived Snapshot Aggregator	Enable Service	Electric

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Type	Aggregation Dimensions — Activity Type	Aggregation Dimensions — Service Type
Activity Processing Accumulation Aggregator	Exchange Meter	Electric
Activity Aging Snapshot Aggregator	Exchange Meter	Electric
Activity Processing Snapshot Aggregator	Exchange Meter	Electric
Activity Processing Derived Snapshot Aggregator	Exchange Meter	Electric
Activity Processing Accumulation Aggregator	Reconnect Service for Payment Orchestrator	Electric
Activity Aging Snapshot Aggregator	Reconnect Service for Payment Orchestrator	Electric
Activity Processing Snapshot Aggregator	Reconnect Service for Payment Orchestrator	Electric
Activity Processing Derived Snapshot Aggregator	Reconnect Service for Payment Orchestrator	Electric

If you wanted to also capture statistics for smart meter commands or field task types, you would need to create activity aggregator measuring components for each individual command or field task type for each measuring component type.

If you are creating service order activities from multiple initiating systems (such as a customer information system and a network management system), you would need to create separate sets of activity aggregator measuring components for each initiating system.

Configure the Service Order Management Master Configuration

You can access the Master Configuration portal by selecting **Admin**, then **General**, then **Master Configuration**.

Select the Service Order Management Master Configuration.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

Create an Activity for each Measuring Component Type

Create an activity for each measuring component type created above. Based on the example above (from the demonstration database) you would create an activity for each of the following measuring component types for each service type:

- Activity Processing Accumulation Aggregator
- Activity Aging Snapshot Aggregator
- Activity Processing Snapshot Aggregator
- Activity Processing Derived Snapshot Aggregator

For example, if you configured measuring component types for both "Electric" and "Water" service types, you would create eight activities (two of each of the types listed above).

Use the following procedure to create these activities:

Copyright © 2026, Oracle and/or its affiliates.

1. Select **Menu**, then **Communication**, then **Activity**, then **Add**. The **Select Type** page opens.
2. Select "Activity Statistics Aggregator Creator" from the **Activity Type** drop-down list and click **OK**. The **Activity Statistics Aggregator Creator** page opens.
3. Select the appropriate measuring component type from the **Aggregator Measuring Component Type** drop-down list.
4. Select the appropriate service type ("E, "Water", and so on) from the **Service Type** drop-down list.
5. Click **Save**.

See Maintaining Activities in the *Business User Guide* for more information about creating and maintaining activities.

Aggregation and Batch Processing

The Service Order Management dashboards are based on aggregated data, and are configured to be refreshed daily. To refresh the data used in the dashboard, the following aggregation and batch processes should be processed in sequence each day.

Dimension Scanning

- **D1-ADS** (Aggregation Dimension Scanner Monitor): Used to transition "Activity Statistics Aggregator Creator" activities (created above) to the "Scan" state which checks for aggregator measuring components based on each measuring component type for each combination of dimensions. If needed, this process creates additional aggregator measuring components.

Use the following parameters when running this batch process:

- Measuring Component Maintenance Object (maintenanceObject): D1-ACTIVITY
- Restrict by Batch Code (isRestrictedByBatchCode): true
- Restrict to Business Object (restrictToBusinessObject): D1-ActivityStatsAggtorCreator
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Note: Dimension scanning can also be performed manually by clicking **Scan** on the individual activities in the **Activity** portal.

Aggregation

- **D1-ACTAG** (Activity Statistics Aggregation Monitor): Used to aggregate activity data (via aggregator measuring components) to be displayed in the Service Order Management dashboards.

Use the following parameters when running this batch process:

- Restrict by Batch Code (isRestrictedByBatchCode): true
- Restrict to Business Object (restrictToBusinessObject): Aggregator Measuring Component business object, based on the measuring component types configured earlier:
  - D1-ActivityAgingSnapAggtor (Activity Aging Snapshot Aggregator)
  - D1-ActivityProcsAggregator (Activity Processing Accumulation Aggregator)
  - D1-ActivityProcDerivedSnapAgg (Activity Derived Snapshot Aggregator)
  - D1-ActivityProcSnapAggtor (Activity Processing Snapshot Aggregator)
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Copyright © 2026, Oracle and/or its affiliates.

- Override Maximum Errors (maxErrors): 1000

Note: Aggregation can also be performed manually by clicking **Update Statistics** on the **Activity Statistics** dashboard zone.

Configuring the Product Use Metrics Dashboard

This section describes the process for configuring the **Product Use Metrics** dashboard used with Oracle Utilities applications, including Oracle Utilities Billing Cloud Service (BCS), Oracle Utilities Customer to Meter (C2M), Oracle Utilities Customer Care and Billing (CCB), Oracle Utilities Customer Care and Billing Cloud Service (CCSBCS), Oracle Utilities Customer Cloud Service (CCS), Oracle Utilities Meter Data Management (MDM), Oracle Utilities Meter Solution Cloud Service (MSCS), Oracle Utilities Work and Asset Management (WAM), and Oracle Utilities Work and Asset Cloud Service (WACS).

The Product Use Metrics dashboard displays customer counts such as device counts, channels of an active meters, number of billable services, head end device counts, and other similar data. The data presented on each dashboard is tailored to provide summary information for management. Refer to Product / Metrics for more information about this dashboard.

The dashboard contains a number of zones based on the product you're using. The table below lists the zones and the products for which they are available.

Zone	Products	Uses Statistics Snapshots
Asset Count	BCS, C2M, CCS, WAM, WACS	No
Billable Services by CIS Division	BCS, C2M, CCB, CCBCS, CCS	Yes
Billable Services by SA Type	BCS, C2M, CCB, CCBCS, CCS	Yes
Billable Services by Service Type	BCS, C2M, CCB, CCBCS, CCS	Yes
Channel Counts	BCS, C2M, CCS, MDM, MSCS	Yes
Device Counts	BCS, C2M, CCS, MDM, MSCS	Yes
Head End Device Count	BCS, C2M, CCS, MDM, MSCS	Yes
Usage Subscription Customer Count	BCS, C2M, CCS, MDM, MSCS	Yes
User Count	BCS, C2M, CCS, WAM, WACS	No

Several of the zones on this dashboard leverage a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on Statistics Snapshots.

Configuring the **Product Use Metrics** dashboard includes the following:

Configure Statistics Category Lookup Value

Statistic Controls (see below) reference a "Statistics Category" defined as a value for the Statistics Category lookup (STAT_CATEGORY_FLG). Create a value for this lookup as follows:

- **Field**: PUMT
- **Description**: Product Use Metrics
- **Java Field Name**: productUseMetrics
- **Status**: Active

Copyright © 2026, Oracle and/or its affiliates.

Configure Statistic Control and Statistic Snapshots

Statistic Controls are the source for capturing the numbers of active usage subscriptions and, device counts per head end system.

Statistic Snapshots capture the numbers for each metric. Statistic Controls are configuration records that define the specific types of metrics to capture.

Statistic Snapshots are captured periodically via the F1-STATS (Statistics Control Monitor) batch program.

Configure the following Statistics Control records:

Statistics Control	Description	Category	Status	Snapshot Frequency Hours	Retention Mode	Snapshot Business Object
HEAD-END-DEVICE-COUNT	Head End Device Count	Product Use Metrics	Active	1.00	Only Retain Most Recent Snapshot	Head End Device Count Snapshot
US-CUSTOMER-COUNT	Usage Subscription Customer Count	Product Use Metrics	Active	1.00	Only Retain Most Recent Snapshot	Usage Subscription Customer Count Snapshot

See Capturing Statistics for more information about statistics.

Configure Device Count and Channel Count Product Metric Types

Product Metric Types define configuration related to a given metric, including which cloud offerings it applies to and the product metric snapshot business object to use when capturing the metric. Corfirm the following Product Metric Types are defined.

Product Metric Type	Description	Snapshot Business Object	Extrenal Name(s)
D1-CHANNELCOUNTMETRICTYPE	Channel Count Metric Type	Channel Count Metric (D1-ChannelCountMetric)	CHANNEL COUNT TOTAL DAILY READ COUNT TOTAL
D1-DEVICECOUNTMETRICTYPE	Device Count Metric Type	Device Count Metric (D1-DeviceCountMetric)	DEVICE COUNT

See Capturing Statistics for more information.

Configure Device Count and Channel Count Snapshot Business Objects and Algorithms (Optional)

By default, the Device Count Metric and Channel Count Metric business objects check for active usage subscriptions when deriving the total count of active devices and measuring components. However, in some implementations this is not required or appropriate.

The usage subscription check can be disabled if needed by setting the "Exclude US Check (Y or N)" parameter on the "Create XXXXX Count Metric Shapshot" Enter algorithms on the

Copyright © 2026, Oracle and/or its affiliates.

Complete state of the Snapshot business objects to "Y" (this parameter is set to "N" by default).

The table below lists the snapshot business objects and their respective algorithms.

Snapshot Business Object	Count Snapshot Algorithm
Channel Count Metric (D1-ChannelCountMetric)	Create Channel Count Metric Snapshot (D1CHLCNTSNP)
Device Count Metric (D1-DeviceCountMetric)	Create Device Count Metric Snapshot (D1DEVCNTSNP)

Configure Customer Metrics Dashboard Configuration Master Configuration (Optional)

The Customer Metrics Dashboard Configuration master configuration defines properties used with the Billable Services zones in the Products Use Metrics dashboard, including:

- Water Service Types and Waste Water Service Type
- Excluded Billable SA Types
- Excluded Debt Classes
- Service Type Colors
- CIS Division Colors

Use the **Master Configuration** portal to configure this master configuration as appropriate for your implementation.

Configure Statistics Control Business Object

The Statistics Control (F1-Statistics) business object should be configured to specify the "Statistics Control Monitor" batch control in the **Monitor Process** drop-down list on the "Active" state.

To configure this business object, do the following:

1. Open the Statistics Control (F1-Statistics) business object in the **Business Object** portal.
2. Navigate to the **Lifecycle** tab, and expand the "Active" state.
3. Select "Statistics Control Monitor" from the **Monitor Process** drop-down list.
4. Click **Save**.

Batch Processing

The **Product Use Metrics** dashboard is configured to be refreshed periodically. To refresh the data used in the dashboard, the following batch processes should be processed in sequence after other regularly schedule batch processing is complete.

- **F1-STATS** (Statistics Control Monitor): Used to generate Statistic Snapshots based on Statistic Controls.

Use the following parameters when running this batch process:

- Maintenance Object (maintenanceObject): F1-STATS
- Restrict by Business Object (restrictToBusinessObject): F1-Statistics
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Copyright © 2026, Oracle and/or its affiliates.

Measurement Reprocessing

- Configuring Measurement Reprocessing

Configuring Measurement Reprocessing

This section describes the process for configuring Measurement Reprocessing.

Refer to the About Measurement Reprocessing section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

Activity Type Configuration

Configuration for Measurement Reprocessing is performed by adding or changing an Activity Type. You can access the portal by selecting **Admin**, **Communication**, **Activity Type**.

Once the Activity Type search screen returns, locate the following records:

- **Measurement Reprocess Activity - Interval:** this Activity Type handles measurement reprocessing for interval Measuring Components.
- **Measurement Reprocess Activity - Scalar:** this Activity Type handles measurement reprocessing for scalar Measuring Components.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field. By adding these Activity Types you are activating the process within Oracle Utilities Meter Data Management that will monitor changes to either the Measuring Component multiplier (the **Register Multiplier** for register measuring components or **Channel Multiplier** for interval and auto-read register measuring components) or the Install Event's **Installation Constant** or **Installation Date/Time**. If any of these attributes change for a device, then a new Activity will be created that attempts to reprocess the measurements for the affected period.

Algorithm Configuration

The Detect Installation Constant Change for Install Event (D1-DETIECHG) algorithm detects changes to install events, including changes to the **Installation Constant**, changes to the **Installation Date/Time**, and creation of new install events whose date range includes preexisting measurements and which include a change to the **Installation Constant**. When the algorithm detects a change to any of these, it initiates measurement reprocessing.

The "Ignore Install Event Date Changes" parameter can disable automatic initiation of measurement reprocessing in the event of changes to the **Installation Date/Time**.

The "Ignore New Install Events" parameter can disable automatic initiation of measurement reprocessing in the event of a new install event being created with a date range that includes preexisting measurements with a different **Installation Constant**.

Related Batch Controls

There are a few batches involved with Measurement Reprocessing:

- **Measurement Reprocessing Monitor** (D1-MRAC): this processes any new Measurement Reprocessing activities that are created.

Copyright © 2026, Oracle and/or its affiliates.

- **Measurement Reprocessing - Retry Monitor** (D1-MRER): this process retries any Measurement Reprocessing activities that fail to the Issue Detected state.

Information Lifecycle Management (ILM)

- Understanding Information Lifecycle Management (ILM)
- Configuring Information Lifecycle Management (ILM)
- Archiving Financial Objects
- Preparing For Financial Transaction (FT) ILM

Understanding Information Lifecycle Management (ILM)

Information Lifecycle Management (ILM) is a separately licensable component of Oracle Utilities Customer to Meter that works in tandem with Oracle database feature of the same name.

For further background on the overall ILM process see the Information Lifecycle Management chapter of the Oracle Utilities Application Framework *Administrative User Guide*.

This section provides an overview of the components used by the Information Lifecycle Management functionality, including:

- ILM-Enabled Maintenance Objects: Single Retention Period
- ILM-Enabled Maintenance Objects: Multiple Retention Periods
- Adjusting Eligibility of Non-Final Transactions
- Archive Eligibility Hierarchy
- Multiple Retention Period Eligibility Crawling Strategies

Refer to the Information Lifecycle Management chapter of the Oracle Utilities Application Framework *Administrative User Guide* for general information about how ILM works with OUAF applications.

ILM-Enabled Maintenance Objects: Single Retention Periods

The majority of Maintenance Objects support a single retention period. They can either inherit the system wide retention days that are configured on the ILM Configuration master configuration or they can have a retention period specified via the Maintenance Object option ILM Retention Period in Days.

For these objects either the Maintenance Object specific retention period or the system wide retention period will apply to all transactions.

ILM-Enabled Maintenance Objects: Multiple Retention Periods

For several Maintenance Objects the retention period may vary based on the type of transaction. For example, initial measurement data for non-billed units of measure such as voltage might be archived much more rapidly than measurements that feed into the production of billing determinants.

These maintenance objects are differentiated from the single retention period maintenance objects in a few ways:

Copyright © 2026, Oracle and/or its affiliates.

- The primary table for the entity has an additional column Retention Period (RETENTION_PERIOD) that captures the retention period in days for each transaction. This value is populated based on the ILM Configuration - MDM master configuration.
- There is additional configuration available on the ILM Configuration -MDM master configuration to define multiple retention periods by maintenance object specific criteria
- Initial Measurement Data: retention periods can be specified by type of IMD (Interval vs Scalar), and primary UOM of the measuring component
- Device Events: retention periods can be specified by device event type category
- Activities: retention periods can be specified by activity type category
- There is a special ILM Crawler batch that supports crawling transactions by their specific retention periods
- A combination of each transactions ILM Date and Retention Period are considered when identifying records to evaluate for eligibility. Refer to batch controls ILM Crawler - Device Event (D1-DECRL), ILM Crawler - IMD (D1-IMDCL), or ILM Crawler - Activity (D1-ACTCR) for more details.
- There is an additional maintenance object option ILM Date Partition Months which feeds into the ILM Crawler batch to delay processing until all records on a partition would be eligible. This is explained later in this section under the header Multiple Retention Period Eligibility Crawling Strategies.

Adjusting Eligibility of Non-Final Transactions

As delivered Oracle Utilities Meter Data Management and Customer to Meter maintenance objects allow non-final transactions to be eligible for archive. This is a logical approach when transactional data has a very long retention period (e.g. 3 year old non-final Initial Measurement Data is unlikely to have relevance and in many circumstances shouldn't be finalized).

If non-final transactions should not be eligible for archive this can be controlled at either the Maintenance Object (MO) or Business Object (BO) level:

- **MO - ILM Restrict By Status (Y/N)**: Setting this to "Y" opts into the ability to restrict archiving by status, this is required to be set for either of the MO or BO level restrictions to be considered. To enforce the status it requires that either the **ILM Restrict By BO Final Status (Y/N)** or **ILM Final Status Field Value** options be configured as well.
- **MO - ILM Restrict By BO Final Status (Y/N)**: Setting this to "Y" indicates the transactions for the MO must be in a final status to be eligible for ILM. A status is considered to be final based on the BO lifecycle configuration. This can be overridden by the BO option **Final Status Required for Archive (Y/N)**. This is only applicable for an MO that is maintained by business objects.
- **BO - Final Status Required for Archive (Y/N):** Setting this to "Y" indicates the transactions for this particular BO must be in a final status. This setting overrides the MO level **ILM Restrict By BO Final Status (Y/N).**

The following table helps to illustrate how these three settings impact whether a given transaction must be in a final status:

• MO: ILM Restrict By Status	• MO: ILM Restrict By BO Final Status	• BO: Final Status Required for Archive	• Final Status Required for Eligibility?
• N	• Y	• Y	• N
• N	• Y	• N	• N

Copyright © 2026, Oracle and/or its affiliates.

MO: ILM Restrict By Status	MO: ILM Restrict By BO Final Status	BO: Final Status Required for Archive	Final Status Required for Eligibility?
N	N	Y	N
Y	N	N	N
Y	Y	N	N
Y	Y	Y	Y
Y	N	Y	Y
Y	Y	Not Provided	Y
Y	N	Not Provided	N

> **Note**
>
> There are a few MOs that do not support the BO level setting. To best understand how these fields are interpreted for an MO, refer to the detailed description of the algorithm plugged into the ILM Eligibility system event for the MO. For example, the generic ILM Eligibility Based on Status (F1-ILMELIG) algorithm does not support the BO level override option.

CCB ILM-Managed MOs

The CCB MOs support only a single retention period and some MOs do not support the BO level setting.

Archive Eligibility Hierarchy

The table below summarizes details of the eligibility algorithms for each Oracle Utilities Meter Data Management maintenance object supported by ILM. Information on this table includes:

- Cascaded By: Indicates how archiving for transactions for the maintenance object is initiated. For example, device events can be archived by themselves or as part of a related activity.
- ILM Date Computation: Indicates the type(s) of transactions that impact the calculation of the ILM Date for transactions for the maintenance object. "Itself" indicates that the ILM date is not impacted by other transactions.
- Specific Eligibility Considerations: Indicates specific considerations used by the eligibility algorithm.
- Cascaded Transactions: Indicates other transactions that are archived when transactions for the maintenance object are archived. For example, VEE exceptions are archived with initial measurements.

Copyright © 2026, Oracle and/or its affiliates.

Maintenance Object	Cascaded By	ILM Date Computation	Specific Eligibility Considerations	Cascaded Transactions
Initial Measurement		Create date	VEE Exceptions must be eligible Associated Related MC Synchronization activity must be complete	VEE Exceptions
VEE Exception	Initial Measurement	Earlier of related Initial Measurement or VEE Exception Create Date	Related Service Issue Monitor must be complete	
Usage Transaction		The latest create date for all related usage transactions	All related usage transactions must be eligible for archiving	Related Usage Transactions
Device Event	Activity	Create date Superseded by paired event create date if earlier (where applicable)	Related paired events must be eligible Related service issue monitor must be complete Related activity must be complete	
Activity	Parent activity	Create date Superseded Device event, child activity, or completion event create date supersede (where applicable)	Cascaded transactions must be eligible Command requests must not be associated to a non-final service issue monitor Command requests must not be associated to an install event on off history entry	Child activities Device events Outbound communications Inbound communications Completion events
Outbound Communication	Activity	Create Date Superseded by initiating activity's create date (where applicable)		Inbound communication
Inbound Communication	Outbound Communication	Create date Superseded by initiating outbound communication create date (where applicable)		Device events

Copyright © 2026, Oracle and/or its affiliates.

Proactive Archive Eligibility for Transactional Data

Some types of transaction data, including Device Events, Initial Measurement Data, and Usage Transactions can be analyzed for archive eligibility as they are received rather than waiting until their retention period has expired. This means that these records are judged ready to be archived as they are created and do not need to be read, analyzed, and updated during the ILM crawling process. This greatly reduces the number of records to be crawled at the end of the retention period and shortens the window of time required for ILM crawling. This section outlines how the ILM Archive switch (ILM_ARCH_SW) is set to “Y” (Yes) for records of these types as part of their initial processing.

- **Device Events**:
  - The ILM Archive switch is set to “Y” for device events that meet the following criteria:
    - No service issue monitors were created for the device event
    - The device event is not a paired device event
  - The ILM Archive switch is set to “Y” for device events via the following algorithms:
    - Create Service Issue Monitor from Device Event (D1-DVCEVTSIM)
    - Create Service Issue Monitor from Device Event for Paired Event (D1-PRDVEVSIM)
- **Initial Measurement Data**:
  - The ILM Archive switch is set to “Y” by default for initial measurements (this is a different approach than the other record types).
  - The ILM Archive switch is set to “N” for initial measurements that meet the following criteria:
    - A consumption synchronization activity has been created for the initial measurement
    - A service issue monitor initiated from a VEE exception related to the initial measurement
  - The ILM Archive switch is set to “N” for initial measurements that meet the above criteria via the following algorithms:
    - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
    - Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
    - Create Service Issue Monitor from VEE Exception (D2-VEEEXCSIM)
- **Usage Transactions**:
  - The ILM Archive switch is set to “Y” when a usage transaction enters the “Sent” or “Discarded” state (or the “Calculated” or “Discarded” states for a sub-usage transaction) via the following algorithms:
    - Send Usage (D2–SEND-USG)
    - Set ILM Switch to ‘Y’ (D1-SETILMSWY)

Multiple Retention Period Eligibility Crawling Strategies

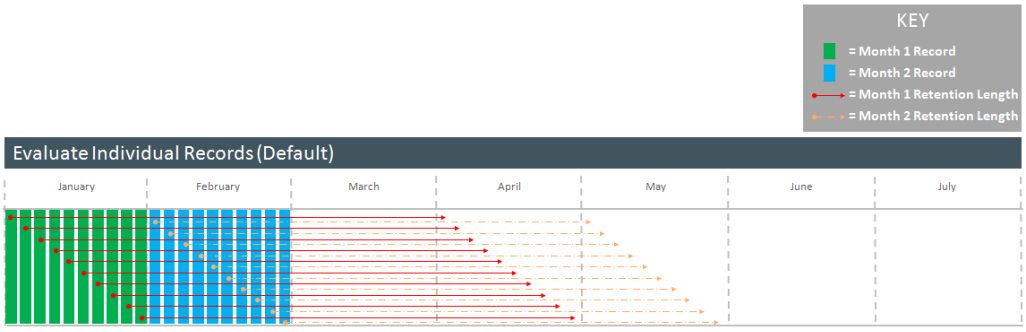
When multiple retention periods are defined for a maintenance object each record will have its retention period written to the database at the time of record creation. This retention period is then used to identify when the record will be ready for eligibility evaluation (aka crawling).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> As specified in the Oracle Utilities Meter Data Management Database Administrator's Guide each unique retention period for the maintenance object will be a sub-partition within a given ILM date partition.

The system supports two methodologies for doing so:

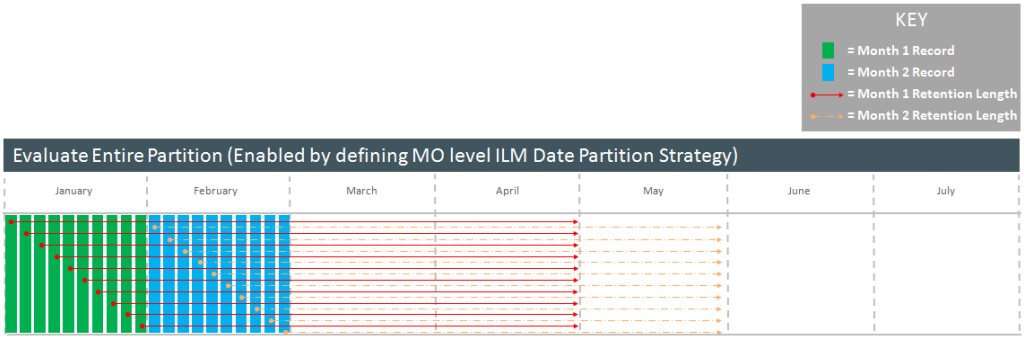
- Evaluate Individual Records:
  - This is the default approach.
  - This is the standard approach to evaluating ILM retention for all maintenance objects with a single retention period.
  - The ILM date for each record will be compared against the current date less the retention period for that type of record. Simply stated, any records older than the retention period will be evaluated for eligibility.
  - In this option, a portion of a partition is processed each day. It is only after the entire partition is processed that the DBA can archive the partition
  - This results in many visits to the partition, to determine eligibility.
  - The image below illustrates how the 'evaluate individual records' method will crawl one day of records on a partition and sub-partition each day until the retention period for the last day of the partition has expired. All records would then be archived together when the last day of the partition is marked as eligible. This results in daily processing of small amounts of records but frequent visits to the partition.



- Evaluate the Entire Partition:
  - This option is enabled by configuring the ILM Date Partition Months maintenance object option.
  - In this option, all records on an ILM date partition retention period sub-partition will be evaluated at the same time.
  - For example, for a monthly partitioning strategy in the month of January, records in the January partition will not be evaluated until the January 31 records have reached the end of their retention period.
  - This means that all records for the 1st through the 31st are evaluated at one time.
  - This option typically results in a single visit to the partition, to determine eligibility
  - The image below illustrates how the 'evaluate entire partition' method will only crawl records from a given partition and sub-partition when all records on that partition are eligible to be evaluated. Therefore, when the last day of the partition is ready to be

Copyright © 2026, Oracle and/or its affiliates.

evaluated, the entire partition will be evaluated. This results in periodic processing with, and in many cases, a single visit to the partition for ILM purposes.



Configuring Information Lifecycle Management (ILM)

For additional information on configuring ILM, see Enabling ILM for Supported Maintenance Objects in the Oracle Utilities Application Framework *Administrative User Guide*.

The tasks described in the OUAF guide must be applied in conjunction with the task of configuring the Oracle Utilities Meter Data Management specific master configuration.

You can access the portal from by selecting **Admin**, then **General**, then **Master Configuration**.

Once the Master Configuration screen returns, locate the following record:

- **ILM Configuration - MDM:** this enables multiple retention periods to be configured for Initial Measurements, Device Events, and Activities.

Related Eligibility Algorithms

There are a several Oracle Utilities Meter Data Management specific algorithms involved with ILM refer to the detailed description of the algorithm type of each algorithm for more information about the functionality each provides:

1. **ILM Eligibility - Activity** (D1-ILMELGACT): determines eligibility of Activities and cascades child Activities, Communication Out, Communication In, Device Events, and Completion Events.
2. **ILM Eligibility - Communication In** (D1-ILMELGCI): determines eligibility of Communication In.
3. **ILM Eligibility - Communication Out** (D1-ILMELGCO): determines eligibility of Communication Out and cascades Communication In.
4. **ILM Eligibility - IMD** (D1-ILMELGIMD): determines eligibility of Initial Measurement data and cascades VEE Exceptions.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If applicable, update the **Populate measurement reading with IMD read when archiving (true/false)** parameter on this algorithm to `true` before running the ILM Crawler - IMD (D1-IMDCL) batch control.
> Setting this parameter to "true" instructs the algorithm to populate the measurement reading value from the Initial Measurement Data (IMD) while D1-IMDCL processes the records. This configuration is required only when the measurement reading value exists in the Initial Measurement Data (IMD) but is not populated in the corresponding measurement record. Customers should confirm this condition before changing the D1-ILMELGIMD parameter to `true`. This setting is not required when the measurement records already contain the reading value.

> **Note**
>
> If an Initial Measurement Data record is marked as ready for archive and a corresponding control table record still exists this algorithm will delete the control table record.

5. **ILM Eligibility - Device Event** (D1-ILMELGDE): determines eligibility of Device Event data and cascades VEE Exceptions.
6. **ILM Eligibility - Usage Transaction** (D1-ILMELIGUT): determines eligibility of Usage Transactions.

> **Note**
>
> If a Usage Transaction record is marked as ready for archive and a corresponding control table record still exists this algorithm will delete the control table record.

7. **ILM Eligibility - Usage Transaction Exception** (D1-ILMELGUEX): determines eligibility of Usage Transaction Exceptions.
8. **ILM Eligibility - VEE Exception** (D1-ILMELGVEX): determines eligibility of VEE Exceptions.

The following provides additional information beyond that provided in the ILM Eligibility - Activity (D1-ILMELGACT) algorithms' detailed descriptions on how specific activity types are handled:

- Request Orchestration Activities: Request orchestrations also archive any child activities that were created, and those child activities archive any child activities or data (completion events, device events, etc.).
- Field Activities / Command Activities: Field activities and command request activities are archived either as part of a request orchestration or by themselves. When a field activity/ command request is archived it also archives any of the following child transactions:
  - Update/Cancel Orchestrators
  - Communication Out
  - Communication In
  - Completion Event

Copyright © 2026, Oracle and/or its affiliates.

- Non-Dispatchable Activities: Non-Dispatchable activities archived either as part of a request orchestration or by themselves. When a non-dispatchable activity is archived it also archives child completion events.
- Orchestration Maintenance Activities: Orchestration Maintenance activities are only archived by themselves if they do not have a related activity. When an orchestration maintenance activity is archived it also archives any of the following child transactions:
  - Communication Out
  - Communication In
- Device Event Activities: Presently device event activities are limited to outages which have an initiating and an ending device event. These types of activities cannot be archived until their related device events are either archived or ready to archive. However, the device events do not have to wait for the activity to archive so the device events are not updated with the activity's ILM date.
- Bulk Activities: Bulk Activities are comprised of the Bulk Header and the Bulk Request/Response, and depending on how the header was created there will be one header to many bulk requests/responses. The Bulk Header will only be eligible for archive if all related Bulk Request activities are also eligible for archiving. All related bulk activities are archived together. Bulk activities also generate one-to-many command request activities. Those individual command activities are archived separately.
- Extract Request Activities: These activities are used to request data from the head-end system on a periodic basis. Extract request activities should be in a final state prior to being archived.
- Other Activities: Other types of activities can be archived provided they do not require any special logic for handling for archiving purposes, such as checking for related data. These activity types can archive without checking related transactions:
  - Consumption Sync
  - Dimension Scanner
  - Error Activity
  - Measurement Quantity (deprecated)
  - Meter Read Download Activity
  - Suppression
  - Usage Transaction Correction Processor

In the event that your implementation uses custom activity types that require special handling, a custom algorithm should be created and added prior to the base package algorithm to preemptively handle the activity type category.

Related Batch Controls

There are a several Oracle Utilities Meter Data Management specific batch controls involved with ILM:

1. **ILM Crawler - Activity** (D1-ACTCR): identifies and executes eligibility evaluation for Activities. This batch control supports multiple retention periods.
2. **ILM Crawler - Communication In** (D1-CICRL): identifies and executes eligibility evaluation for Communication In.
3. **ILM Crawler - Communication Out** (D1-COCRL): identifies and executes eligibility evaluation for Communication Out.

Copyright © 2026, Oracle and/or its affiliates.

4. **ILM Crawler - Device Event** (D1-DECRL): identifies and executes eligibility evaluation for Device Events.
5. **ILM Crawler - IMD** (D1-IMDCL): identifies and executes eligibility evaluation for Initial Measurement Data. This batch control supports multiple retention periods.
6. **ILM Crawler - Usage Transaction Exception** (D1-UEXCL): identifies and executes eligibility evaluation for Usage Transaction Exceptions.
7. **ILM Crawler - Usage Transaction** (D1-UTCRL): identifies and executes eligibility evaluation for Usage Transactions.
8. **ILM Crawler - VEE Exception** (D1-VEXCL): identifies and executes eligibility evaluation for VEE Exceptions.

For additional details see the Batch Processes section the Oracle Utilities Application Framework *Administrative User Guide*.

Archiving Financial Objects

This section describes considerations related to the ILM-enabled financial objects, such as bills, payments, adjustments, match events and financial transactions.

Retention Periods of Financial Objects

Each financial MO's eligibility criteria include checks on the status of related objects. For instance, financial transactions (FTs) are eligible for archive only if the related bill segment, payment segment, adjustment or match event is either archived (dropped from partition) or eligible for archive (record still exists with the ILM Archive Switch set to 'Y'). So it's important that the retention periods of the related financial objects align with each other.

Archiving Historical Bill Segments

For historical bill segments being archived, your implementation should not archive bills and bill segments that are eligible to be rebilled or those that are required for re-billing subsequent bills or bill segments - e.g., for determining bill segment start dates, if historical service quantities are required for calculation purposes, etc.. From a billing perspective, these may still be considered active.

Preparing For Financial Transaction (FT) ILM

This section describes the preliminary tasks needed for enabling Financial Transaction (FT) Information Lifecycle Management. Note that while the FT table (CI_FT) has the ILM Date and ILM Archive switch, the actual partitioning of the FT table is not yet enabled. However, the batch processes described in this section are provided, to allow your implementation to prepare your financial transaction data in the meantime.

For general information on ILM enabling, see Enabling ILM for Supported Maintenance Objects in the Oracle Utilities Application Framework *Administrative User Guide*.

The tasks described in the OUAF guide must be applied in conjunction with the batch processes that are described here.

FT ILM Objects

Since FT data is typically high-volume, the FT ILM crawler is a specific plugin-driven batch process that retrieves and updates the FTs in bulk (i.e. set-based update). This is for

Copyright © 2026, Oracle and/or its affiliates.

performance considerations. The template ILM crawler batch process in OUAF evaluates one record at a timeline and is unlikely to perform well on high-volume FT records.

The other aspect that sets the FT MO apart from other ILM-enabled MOs is that it does not use the **ILM Eligibility** plugin spot. The eligibility criteria checking is built into the FT ILM crawler's processing, for efficiency. It saves from having to call another plugin spot, which can make a difference in high-volume batch processing.

An FT will be marked as eligible for ILM if the FT is already marked as Redundant (switch is set to **Y**) and the FT is linked to a Bill Segment, Pay Segment, Adjustment or Match Event that either has the ILM Archive Switch set to **Y** or no longer exists (i.e., dropped from the partition).

Refer to the FT ILM Crawler (C1-FTCRL) batch process, for more information.

FT ILM Batch Processes

There are several batch controls involved with FT ILM. The batch processes are listed in the order in which they should run:

1. **Update FT ILM Fields** (C1-FTUPD) is a plugin-driven batch process that is meant to be run after the upgrade, at which point, the ILM Dates and ILM Archive switches in the FT table are null. This upgrade batch process sets the ILM Date to the FT's creation date and the ILM Archive Switch to **N**. The FT records are updated in bulk, for performance considerations.
    - Due to typical high volumes of FT data, the batch runs should be staggered by Cutoff Date, to manage the load of each run.
    - To facilitate the set-based processing of FTs, specify the appropriate Thread Option, which is either by Service Agreement ID or by Accounting Date. The default option is Accounting Date.
    - This batch process is required to be run if the FT table has records with null values in the ILM Date and/or ILM Archive Switch. This will be true for FT records created before the upgrade to 25.10. All FT records must have the ILM Date and ILM Archive Switch populated, to enable the FT MO for ILM.
2. An FT is eligible for ILM only if it is marked as redundant. FTs are marked as redundant when, together with other old FTs on the same SA, their balance sums to zero.

    The base product provides two batch processes for marking old FTs as redundant - one each for balance forward accounting and open item accounting. Run the applicable batch process before running the FT ILM crawler. Both processes consider an FT as old if the FT's arrears date is less than or equal to the input Cutoff Date, which is either supplied in the batch parameter, or the default (process date minus the Installation record's Oldest Bucket Age (in days)).
    - **Mark FTs Redundant for Balance Forward Account** (C1FTRDBF) is a plugin-driven batch process that marks old FTs of balance-forward account SAs.
    - **Mark FTs Redundant for Open Item Accounts** (C1FTRDOI) is a plugin-driven batch process that marks FTs as redundant for balanced match events where all associated FTs are considered old.
3. **ILM Crawler - Financial Transaction** (C1-FTCRL) is a plugin-driven batch process that reviews FTs and marks the ILM Archive Switches to **Y**, if the FT meets the eligibility criteria. An FT is considered eligible for archiving if:
    - The FT is marked as Redundant (switch is **Y**).
    - The FT's ILM Date is before the retention period.

Copyright © 2026, Oracle and/or its affiliates.

- The associated bill / payment / adjustment / match event is either marked for archiving or already archived (removed from partition).
- Due to typical high volumes of FTs, the batch runs should be staggered by Cutoff Date, to manage the load of each run.
- The Process Algorithm associated with this batch process performs the eligibility criteria checks prior to the FT updates
- To facilitate the set-based processing of FTs, specify the appropriate Thread Option, which is either by Service Agreement ID or by Accounting Date. The default option is Accounting Date.

Refer to these batch controls in the application, for more information.

For general information on batch processes, see the Batch Processes section the Oracle Utilities Application Framework *Administrative User Guide*.

Outage Storm Mode

- Configuring Outage Storm Mode
- Detailed Examples of Outage Storm Mode

Configuring Outage Storm Mode

This section describes the process for configuring Outage Storm Mode.

> **Note**
>
> Refer to the About Outage Storm Mode section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

Master Configuration settings

Configuration for Outage Storm Mode is performed by adding or changing a Master Configuration. You can access the portal from the **Admin**, and then **General**, and then **Master Configuratio**

Once the Master Configuration search screen returns, locate the **MDM Master Configuration**. Use the Add button beside the record to configure for the first time. If a record has already been added, then click the Edit button instead. The **Estimation Eligibility For Widespread Outages** section is the primary section that needs to be configured for Outage Storm Mode to be enabled. Use the embedded help to guide you through the meaning of each configuration field.

Setup Meter Communication Tracking Aggregation

As part of Outage Storm Mode, aggregations are performed to determine the percentage of meter communication that has occurred for an area. The following configuration supports this:

1. Access the portal for Measuring Component Type through **Admin > Device > Measuring Component Type**. Add a new Measuring Component Type using the **Meter Communication Tracking Aggregator Type** (D2-MtrCommTrckngAggregatorType)

Copyright © 2026, Oracle and/or its affiliates.

business object. Follow the embedded help provided for each section. Two areas listed below are crucial to preparing for this specific aggregation:

a. When selecting a Measuring Component Business Object for this record, choose the **Meter Communication Tracking Aggregator** Business Object.

b. For the Value Identifiers, there are a few metrics that are can be defined as listed on the D2-AGG-MCTM Algorithm Type:

   i. Read Percentage

   ii. Count of Received Measuring Components

   iii. Count of Total Measuring Components

2. Access the portal for Activity Type through Admin > Communication > Activity Type. Add a new record for the Meter Communication Tracking Dimension Scanner Activity Type.

3. Access the portal for Activity through **Main > Communication > Activity**. Add a new record for the Activity Type you defined in the prior step.

Standard Event Name Configuration

If there is a desire to indicate that a widespread outage has ended for a device when it receives certain device events, the **Standard Event Name** Extendable Lookup can be used for this. You can access the portal from the **Admin**, and then **General**, and then **Extendable Lookup**

Any Device Event that should close out Estimation Suppression Activities must be configured with the "End Estimation Suppression" algorithm (D2-EN-ESTSUP) in the Additional Processing Algorithms section. This is often configured for "Power Up" events to indicate that normal power flow has resumed for the meter.

Related Batch Controls

There are a few batches directly involved with Outage Storm Mode:

- **Dimension Scanner** (D1-ADS): monitors the "Meter Communication Tracking Dimension Scanner" business object. New Aggregator Measuring Components will be created for every applicable Service Type, Postal Code, and Head End in the system.
- **Aggregation Monitor** (D2-AGG): runs to execute the logic for the "Meter Communication Tracking Aggregator" Measuring Component. This logic will aggregate all meters for the defined Service Type, Postal Code, and Head End. Once the read percentage is found for that day then a measurement will be created for the Aggregator Measuring Component.
- **Estimation Suppression Monitor** (D1-ESTSU): This batch monitors the Estimation Suppression activities for processing. This batch should be scheduled to run regularly so it can clean up Suppression Activities if they have ended.
- **Widespread Outage** (D2-WSO): This batch queries the Meter Communication Tracking Aggregators to determine if any have experienced a widespread outage. For any actively in outage mode, then the batch will find all related Service Points and execute the logic on the algorithm plugged into the BO system event of "Estimation Suppression". It's recommended that D2-WSO be scheduled shortly after aggregation D2-AGG.

Copyright © 2026, Oracle and/or its affiliates.

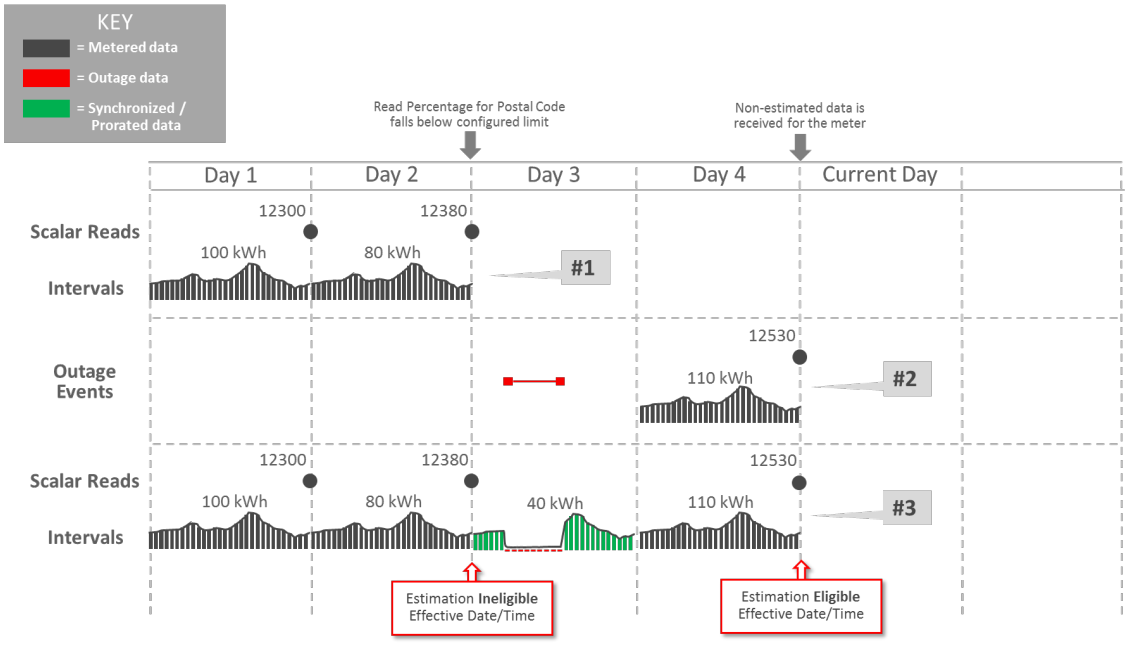
Detailed Examples of Outage Storm Mode

Scenario 1: Meter is marked estimation ineligible, then metered data is received with outage events

The following configuration settings should be assumed for this example:

Master Config option	Value
Check Widespread Outages	Yes
Read Percentage Days to Examine	7
Percent Drop From Norm - Estimation Ineligible	30.0%
Minimum Device Count For Ineligibility	100
Days Before Becoming Ineligible	0
Fill Missing Data With Zero	No
Days Before Filling Zero	
Maximum Estimation Ineligibility Days	3
Percent Within Norm - Resume Estimation	5.0%

Diagram:



Explanation for diagram:

1. The meter was communicating consistently but then a widespread outage occurs in the same postal code so the meter has an estimation suppression created.
2. Two events are received...a Power Down event and a Power Up event. Also, Regular usage data is received from the meter. The reception of non-estimated data marks the meter as eligible for estimation again.

Copyright © 2026, Oracle and/or its affiliates.

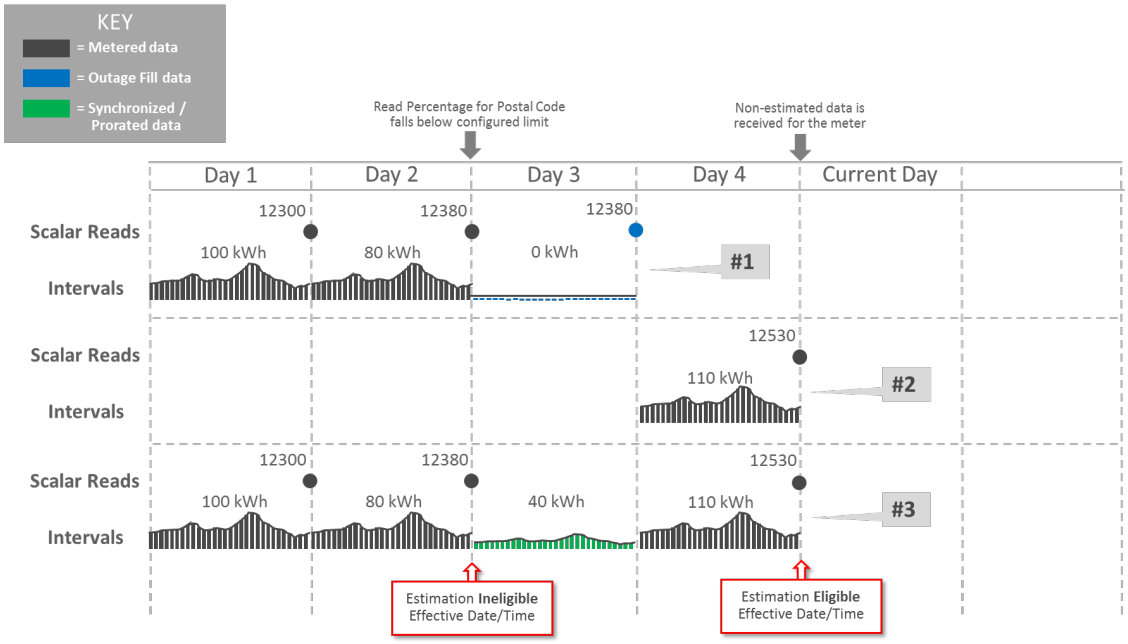
3. Oracle Utilities Meter Data Management estimates data and takes the two events into account. Intervals marked as outage and filled with zeros are created during the outage period by the Device Periodic Estimation (D1-DVCPE) batch.

Scenario 2: Meter is marked estimation ineligible, is filled with zeros, then receives actual data

The following configuration settings should be assumed for this example:

Master Config option	Value
Check Widespread Outages	Yes
Read Percentage Days to Examine	7
Percent Drop From Norm - Estimation Ineligible	30.0%
Minimum Device Count For Ineligibility	100
Days Before Becoming Ineligible	0
Fill Missing Data With Zero	**Yes**
Days Before Filling Zero	**0**
Maximum Estimation Ineligibility Days	3
Percent Within Norm - Resume Estimation	5.0%

Diagram:



Explanation for diagram:

1. The meter loses communication and periodic estimation fills in outage intervals.
2. The meter reestablishes communication and sends in the latest readings and interval data.
3. Oracle Utilities Meter Data Management creates new data that's synchronized to fall in line with the new data from the meter by the Consumption Sync batch. This data replaces the outage measurements as it balances out to the scalar and interval data.

Copyright © 2026, Oracle and/or its affiliates.

Running Initial Periodic Estimation

This section provides guidelines for running an initial periodic estimation process.

- Overview
- Avoiding Long Execution Time
- Overview of Options and Start of Estimation Horizon
- Testing Periodic Estimation on Small Data Sets
- Testing Periodic Estimation on Large Data Sets

Overview

Prior to running the first execution of periodic estimation an assessment should be made regarding the current state of measurement data on your measuring components. Without proper preparation the first execution can take an extremely long time due to large gaps on measuring components. Since the Device Periodic Estimation (D1-DVCPE) batch process is threaded by device, all of the initial measurements for a device's interval and scalar channels will be created within a single work unit. This will lead to an abnormally large number of transactions being created within a single commit which can degrade performance.

There are several options available for establishing a consistent starting point for estimation across all measuring component types:

1. Configure the **Maximum Days to Estimate** on each periodic estimation eligible measuring component type. This limits the number of days periodic estimation will look backwards to create estimations. For example, if set to 60 the no estimations over 60 days old would be created and all measuring components would have their **Last Contiguous Measurement Date Time** set to 60 days ago or more recent.
2. Configure the **No of Hours in the Past to Retrieve Last Usable Measurement** option on the Measurement Data Options feature configuration. This configuration works in two specific ways:
   - When the measuring component's **Last Contiguous Measurement Date/Time** has not been initialized then this setting is used to search for usable measurements starting from the **Process Date/Time** less the number of hours configured.
     - If a measurement is found, the most recent measurement within that window will be used.
     - If no measurements are found, estimates will be generated starting from the calculated start date/time.

     > **Note**
     >
     > Note: if the calculated start date/time is very recent and not yet eligible for estimation then the measuring component's **Latest Contiguous Measurement Date/Time** will be updated with that value.

   - When the **Last Contiguous Measurement Date/Time** has already been set for a measuring component this setting will be used similar to the **Maximum Days To Estimate** and will limit how far back estimation will occur.

Copyright © 2026, Oracle and/or its affiliates.

3. Use the **Initialize Measuring Component Dates** (D1-IMCDT) batch control just prior to the first run of periodic estimation to explicitly set the **Latest Contiguous Measurement Date/Time** across all measuring components to the date you would like to estimate from.
4. The **Perform Changes in New Session** parameter on periodic estimation algorithms (including D1-CIITBIH and D1-ARSPE) can improve processing when many initial measurements are being created in one run of these algorithms. When this parameter is set to "Y" each initial measurement will be created in a separate session and will be committed as they are completed. This improves performance since there are fewer changes being cached and ensures progress is saved in the event of errors during processing.

It is important to take into consideration that the periodic estimation process is tuned with the expectation that it will create estimations for a small percentage of the meter population (generally around 2%) for one day. If on the initial run all devices have measuring components that require estimations to be created, it is increasing the typical work load exponentially: 100% of the meter populate for many days.

Lastly, it is very important, regardless of the method used, to test a small population of meters before running the Device Periodic Estimation (D1-DVCPE) batch process on your entire meter population. Isolated testing will alert you to potential performance issues without having to cancel a multi-day D1-DVCPE execution.

Avoiding Long Execution Time

In general the single most important thing to avoid long execution times on your first run of the Device Periodic Estimation batch process (D1-DVCPE) is to ensure that your measurement data is as current as it possibly can be. You should not run the Device Periodic Estimation batch process (D1-DVCPE) until you have completed conversion and you have loaded your initial set of initial measurements. If you run the Device Periodic Estimation batch process (D1-DVCPE) any time prior to that you risk generating estimates for data that you have but just haven't loaded yet.

If you must run the Device Periodic Estimation batch process (D1-DVCPE) prior to having all of your measurement data current, you should use the **Batch Business Date** on the batch control to limit the horizon of time you will estimating. The **Batch Business Date** won't impact the start of your horizon but it will limit the end.

The following sections provide input on several ways you can control the start of the horizon.

Overview of Options and Start of Estimation Horizon

The table below summaries various options and their impact on the Start of the Estimation horizon.

> **Note**
>
> Always ensure the **Most Recent Measurement Date/Time** is set to the measurement date/time furthest into the future. The **Initialize Measuring Component Dates** (D1-IMCDT) batch control has functionality for initializing the Most Recent Measurement Date/Time as well (if it was not set during conversion).

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Type Maximum Days to Estimate	Feature Configuration	Measuring Component Dates	Measurement Data	Start of Estimation Horizon
Not Configured	Not Configured	Not Set	None	Horizon will begin based on the installation date/time: • Interval uses the installation date/time • Scalar uses the On event
Not Configured	Not Configured	Not Set	Yes, with some gaps	Horizon will begin based on the installation date/time: • Interval uses the installation date/time • Scalar uses the On event
Configured	Any	Not Set	Any	Horizon will begin based on the process date/time less the **Maximum Days to Estimate**
Configured	Any	Set	Yes - prior to Maximum Days to Estimate	Horizon will begin based on the process date/time less the **Maximum Days to Estimate**
Configured	Any	Set	Yes - after Maximum Days to Estimate	Horizon will begin based on the **Latest Contiguous Measurement Date/Time**

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Type Maximum Days to Estimate	Feature Configuration	Measuring Component Dates	Measurement Data	Start of Estimation Horizon
Not Configured	Configured	Not Set	None	Horizon will begin based on the process date/time less the hours configured on the feature configuration. If it is not yet time to estimate based on the calculated start of the estimation horizon, then: • Interval: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start • Scalar: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start and **Adjusted Last Contiguous Measurement Date/Time** is set to the appropriate value based on the periodic estimation configuration. Note: this would happen when the **Hours Before Estimation** (for example 72) is a larger number of hours than the **No of Hours in the Past to Retrieve Last Usable Measurement** (for example 24)

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Type Maximum Days to Estimate	Feature Configuration	Measuring Component Dates	Measurement Data	Start of Estimation Horizon
Not Configured	Configured	Not Set	Yes - outside feature configuration window	Horizon will begin based on the process date/time less the hours configured on the feature configuration. If it is not yet time to estimate based on the calculated start of the estimation horizon, then: • Interval: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start • Scalar: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start and **Adjusted Last Contiguous Measurement Date/Time** is set to the appropriate value based on the periodic estimation configuration. Note: this would happen when the **Hours Before Estimation** (for example 72) is a larger number of hours than the **No of Hours in the Past to Retrieve Last Usable Measurement** (for example 24)
Not Configured	Configured	Not Set	Yes - inside feature configuration window	Horizon will begin as of the most recent measurement date/time that was found for the measuring component.

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Type Maximum Days to Estimate	Feature Configuration	Measuring Component Dates	Measurement Data	Start of Estimation Horizon
Not Configured	Configured	Set	Any	Horizon will begin based on the most recent of the following: • Process date/time less the hours configured on the feature configuration • **Last Contiguous Measurement Date/Time**

Testing Periodic Estimation on Small Data Sets

This topic provides guidelines for testing periodic estimation on small data sets.

How to Override the System Date

The system or process date is a key driver in the periodic estimation process. It will determine the end date/time of the estimation horizon (along with the periodic estimation configuration). When testing a single device this override can ensure that estimations never go further in time then you would like if you are working with an older set of test data.

There are two methods to override the system date (both use the format YYYY-MM-DD) that will work for the methods being described in this section.

- The **System Override Date** option on the **General System Configuration** feature configuration: this impacts all users and batches
- The **System Override Date** characteristic for a specific user: this impacts only processes executed by a particular user

Testing Periodic Estimation on Large Data Sets

This topic provides guidelines for testing periodic estimation on large data sets.

Before running periodic estimation on a large data set it is important to have an idea of roughly how many devices and for how long you expect to estimate. This can be gauged fairly quickly by looking at the **Latest Measurement Date/Time** on the measuring component. Here is a sample query that will provide a count of measuring components by the number of months from a supplied process date the last contiguous measurement date/time is:

```
select count(*)
, round((to_date(:processDateTime,'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm)/31,0) MONTHS
from d1_measr_comp mc
group by round((to_date(:processDateTime,'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm)/31,0)
```


Copyright © 2026, Oracle and/or its affiliates.

```
order by 2
;
```

Below is a more complex query that will show you by thread how many measuring components for a given **Latest Measurement Date/Time** you will process, it does a bit more work to look at only installed devices and measuring components that are configured for periodic estimation and the measuring component is actually ready to estimate based on that configuration.

The parameters for this are:

- threads: enter 1 to avoid splitting the results into threads
- processDateTime: leave blank to use the system date, otherwise provide a date/time in the format: YYYY-MM-DD-HH24.MI.SS

```
with threads as (
 select rownum thread_nbr
 , lpad(round((rownum - 1) * (rpad('9',12,'9')+1)/:threads,0),12,'0') low_id
 , lpad(round(rownum * ((rpad('9',12,'9')+1)/:threads),0)-1,12,'0') high_id
 from dual
 connect by rownum <= :threads
)
select /*+ parallel */ threads.thread_nbr
, threads.low_Id
, threads.high_id
, nvl(mc.adj_latest_msrmt_dttm,mc.latest_msrmt_dttm) LATEST_MSRMT_DTTM
, count(*) MC_COUNT
from threads
inner join d1_dvc dvc
 on dvc.d1_device_id between threads.low_id and threads.high_id
inner join d1_dvc_cfg dc
 on dc.d1_device_id = dvc.d1_device_id
inner join d1_install_evt ie
 on ie.device_config_id = dc.device_config_id
 and ie.d1_install_dttm < sysdate
 and (ie.d1_removal_dttm is null or ie.d1_removal_dttm >= sysdate)
inner join d1_measr_comp mc
 on mc.device_config_id = dc.device_config_id
inner join d1_measr_comp_type mct
 on mct.measr_comp_type_cd = mc.measr_comp_type_cd
 and hrs_wait_before_est > 0
where (to_date(nvl(:processDateTime,sysdate),'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm) * 24 > mct.hrs_wait_before_est
group by threads.thread_nbr
, threads.low_Id
, threads.high_id
, nvl(mc.adj_latest_msrmt_dttm,mc.latest_msrmt_dttm)
order by 1, 4
;
```

Defining Customer Program Management Options

Copyright © 2026, Oracle and/or its affiliates.

- Customer Program Management Configuration
- Customer Program Management Options
- Customer Program Management Integrations
- Defining Other Options

Customer Program Management Configuration

This section describes general configuration involved in implementing Customer Program Management.

General Options

Refer to the General Options section for information about configuration of general options including Installation Options, color options, and access control.

Cross Product Configuration

The Cross Product Configuration master configuration provides a single point of access for configuration information related to consolidated data maintenance, data mapping, and other master configurations used with Customer Program Management.

To open the master configuration, select **Admin**, select **General**, and select **Cross Product Configuration**. The master configuration opens in the **Cross Product Configuration** portal.

The **External System** field references the external system used for communication between functional components within the application such as data synchronization requests (for example, when a controllable asset is created the system also creates a controllable device record).

Customer and Device Value Mapping

The **Customer and Device Mapping** section lists objects used to map data between customer-related objects and meter-related objects when synchronizing data between the applications and external applications including Oracle Utilities Customer to Meter.

SA Type Mapping references the extendable lookup used to map service agreement (SA) types to metered service types. Click **C2M SA Type Mapping Values** to view values for this extendable lookup.

The table lists fields that use Lookup Value Mapping to map values between functional components within the application. Click the Description link to view and/or edit mapping values for a specific field. A plus sign indicates that mapping values have not been defined for that field.

Click the icon to configure values for that field.

Device and Asset Value Mapping

The Device and Asset Value Mapping section lists objects used to map data between asset-related objects and meter-related objects (used when creating new controllable assets and devices).

The table lists fields that use Lookup Value Mapping to map values between functional components within the application. Click the Description link to view and/or edit mapping values for a specific field. A plus sign indicates that mapping values have not been defined for that field. Click the icon to configure values for that field.

Copyright © 2026, Oracle and/or its affiliates.

Customer Program Management Integration Master Configuration

The Customer Program Management Integration master configuration is used to define configuration options used when integrating Customer Program Management with Oracle Utilities Network Management System and/or an external analytics application.

To open the master configuration, select **Admin**, select **General**, select **Master Configuration**, then click either the **Broadcast** or **Add** icon for the Customer Program Integration Master Configuration.

Refer to the **Oracle Utilities Customer Program Management Cloud Service Integration to Oracle Utilities Distributed Energy Resources Management System** documentation for more information about how this master configuration is used with these integrations.

Enabling Customer Program Management

Customer Program Management functionality requires a separate license and is not accessible by default. To configure the system such that menu items and functionality used with Customer Program Management are available, application users should be assigned to the following set of user groups:

- Digital Asset Services (DACS_SERVICES)

Refer to Defining Users in the *Oracle Utilities Application Framework Administrative User's Guide* for more information about setting up users.

- Customer Program Management Table Setup Sequence

Customer Program Management Table Setup Sequence

The table below provides the suggested order for the setup of administrative data used with Customer Program Management.

Order	Entity	Description
1	Feature Configuration	These are options that can be set to impact system processing. Typically these have wide ranging impact across several modules (e.g., multi-time zone support)
1	Country	Your organization's country.
1	Currency	Your organization's native currency.
1	Display Profile	Controls how dates, times, and numbers displayed.
1	Language	The language to use for this implementation.
1	Time Zone	Your organization's base time zone.
1	Work Calendar	The work calendar for your organization, which identifies your public holidays.
2	Installation Options	Control various global aspects of the system.
2	To Do Role	Used to associate users with To Do entries.
2	To Do Type	Used to define types of To Do Entries.
3	User	Defines a user's user groups, data access roles, portal preferences, default values, and To Do roles.

Copyright © 2026, Oracle and/or its affiliates.

Order	Entity	Description
3	User Group	A group of users who have the same degree of security access.
3	Service Area	Defined a logical boundary of your organization's service territory.
3	Service Class	Defines different categories and types of work.
3	Service Type	Defines specific types of service for which usage can be recorded and captured (electric, gas, steam, etc.).
4	Division	Delineates between different operating companies within a large conglomerate of utilities.
4	Unit of Measure	Quantities measured and recorded by the system (CCF, KWH, KW, etc.).
4	Service Quantity Identifier	Used to further distinguish between measured quantities that have identical UOM/TOU combinations.
4	Time of Use	Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used (On- Peak, Off-Peak, etc.).
4	Outbound Message	Defines messages sent to external systems.
4	IMD Type	Define how initial measurements are processed using Direct Measurement Processing. **Note** The base package includes several pre-defined IMD types that can be used when processing data from different head end systems supported through Smart Grid Gateway Adapters.
5	External System	Defines External Systems with which Oracle Utilities Meter Data Management should be able to communicate.
5	Interval Price Set Types	Defines properties belonging to a class of interval price sets
6	Head End Systems	External entities that serve various roles relative to the application (head-end systems, outage management systems, etc.).
6	DER Aggregators	External systems that facilitate communication between a utility and Distributed Energy Resource (DER) devices.

Copyright © 2026, Oracle and/or its affiliates.

Order	Entity	Description
6	External Applications	External entities that serve various roles relative to the application (head-end systems, outage management systems, etc.).
6	Program Contact Type	Defines properties of a class of entities (businesses, persons).
7	Activity Type	Defines properties common to a specific type of activity.
7	Communication Type	Define properties common to a specific type of communication.
7	Service Task Type	Defines specific types of enrollment and unenrollment processes
7	Program Type	Defines information common to dynamic options of a specific type.
7	Manufacturer	Individual companies that makes devices. Manufacturers also reference models.
7	Business Flag Type	Defines the business flag Standard Name and business flag business object to be used when creating business flags
7	VEE Group	Collections of VEE Rules that are applied to initial measurement data.
7	Device Event Type	Defines information about a class of device events. Device events are used to capture periods of time during which customers make themselves unavailable to participate in program events.
8	Controllable Asset Type	Defines information about a class of controllable assets.
9	Controllable Device Type	Defines information about a class of controllable devices.
9	Measuring Component Type	Defines the most important properties of a measuring component.
9	Device Configuration Type	Defines properties of Device Configurations of a given type.
9	Device Type	Defines information about a class of devices.
10	Device Location Type	Defines specific types of locations at which devices are installed.
10	Metered Service Point Type	Defines specific types of points where services are provided.
10	Entity Group	Defines groups of entities, such as devices (including meters or controllable devices) or service points.
10	Weather Station	Defines specific weather stations
10	Postal Code Default	Defines postal code defaults
10	Usage Transaction Exception Types	Defines types of exceptions that may arise as part of program event settlement calculations (such as insufficient or invalid information, or an exception during calculation processsing or when retrieving data).

Copyright © 2026, Oracle and/or its affiliates.

Order	Entity	Description
11	Program Calculation Group	Collections of program calculation rules used to calculate device control scores.
11	Program Settlement Calculation Group	Collections of program settlement calculation rules used to perform program settlement transaction calculations.
12	Program Calculation Rule	Defines rules that calculate device control scores
12	Program Settlement Calculation Rule	Defines rules that perform program settlement transaction calculations
13	Metered Service Type	Defines collections of properties common to a set of metered service records
13	Program Subscription Type	Defines collections of properties common to a set of Program Subscriptions.
14	Program Rules	Defines specific rules and parameters used by programs and program calculation rules.
14	Program Initiatives	Defines criteria used to determine if customers are eligible to participate in programs.
14	Customer Questions	Defines specific questions regarding customers and their service.
15	M&V Calculation Group	Collections of M&V calculation rules used to perform M&V transaction calculations.
15	M&V Calculation Rule	Defines rules that perform M&V transaction calculations
15	M&V Subscription Type	Defines collections of properties common to a set of M&V Subscriptions.
16	Data Source	Defines the source of data for dynamic aggregation, such as Entity Groups for Controllable Devices or Service Points
16	Dynamic Aggregation Measuring Component Type	Define the most important properties of a measuring component used with dynamic aggregation.
16	Measuring Compoment Set	Define the dimensions and criteria by which dynamic aggregation will be performed.
16	Aggregation Group	Define the ordering of a series of related aggregations based on a set of configured measuring component Sets.
17	Service Codes	Defines sets of customer questions used with service calls.
18	Processing Methods	Control various behaviors for external applications, head end systems, and market participants within the system such as which message is sent, how an external value is translated, among others.
19	Master Configuration	Configuration that applies to series of modules that acts as a central point of configuration rather than embedding repetitive configuration throughout a set of algorithms.

Copyright © 2026, Oracle and/or its affiliates.

Order	Entity	Description
20	Information Lifecycle Management	Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle.

Customer Program Management Options

- Understanding Customer Program Management
- Configuring Customer Program Management Options

Understanding Customer Program Management

This section describes customer program management administrative options used with Oracle Utilities Customer Program Management Cloud Service and Oracle Utilities Customer Program Management.

- Understanding Controllable Asset Types
- Understanding Device Location Types
- Understanding Head End Systems
- Understanding DER Aggregators
- Understanding Metered Service Types
- Understanding Metered Service Point Types
- Understanding Program Control Score Calculation Groups
- Understanding Program Control Score Calculation Rules
- Understanding Program Initiatives
- Understanding Program Contact Types
- Understanding Program Subscription Types
- Understanding Program Rules
- Understanding Program Types
- Understanding Program Settlement Calculation Groups
- Understanding Program Settlement Calculation Rules
- Understanding Usage Transaction Exception Types
- Understanding Interval Price Set Types
- Understanding Customer Questions
- Understanding Service Areas
- Understanding Service Classes
- Understanding Service Codes
- Understanding Service Types

Copyright © 2026, Oracle and/or its affiliates.

- Understanding Units of Measure
- Understanding Time of Use Periods
- Understanding Service Quantity Identifiers
- Understanding Divisions
- Understanding Entity Groups
- Understanding Weather Stations
- Understanding Postal Code Defaults
- Understanding M&V Options
- Understanding Customer Program Service Task Types

Understanding Controllable Asset Types

Controllable asset types define the attributes for controllable assets of a certain type, including information such as the valid location types and service history types for the assets and whether or not such assets can have attached components.

Controllable asset types have a related device type record. When a controllable asset type is created (or updated), a corresponding device type is created (or updated). See Understanding Device Types, below for more information about device types.

Controllable asset types are defined by the following:

- **Asset Business Object**: The asset business object used when creating controllable assets of this type.
- **Service Type**: The type of service (such as Electric) associated with controllable assets of this type.
- **Asset Category**: The category of assets to which controllable assets of this type belong. Controllable asset types generally belong to the “Controllable Device” category.

> **Note**
>
> The "Controllable Asset" category is used for assets owned by the customer, such as an HVAC unit, captured for informational purposes.

- **Component Relationship**: A flag that indicates if components can be installed on controllable assets of this type
- **Asset Management**:
  - **DER Processing Flag**: A flag that indicates how registration is handled for Distributed Energy Resource (DER) controllable assets of this type. This flag is required for DER assets. Options include:
    - **Interconnection and Registration**: Registrations of controllable assets of this type requires two registration requests, an interconnection request and a DER Registration request. The Interconnection request should be sent first and must be completed before the DER Registration request can be completed.
    - **Registration Only**: Registrations of controllable assets of this type requires only a DER Registration request
  - **Failure Profile**: Valid failure information (including failure types, failure modes, failure repairs, and failure components) associated with controllable assets of this type.

Copyright © 2026, Oracle and/or its affiliates.

- **Valid Location Types**: A list of valid location types for each asset disposition. Note that the device location types for device locations where controllable asset of this type are to be installed must be specified for the "Installed" asset disposition.
- **Valid Service History Types**: A list of valid service history types for controllable assets of this type.
- **Asset Attribute Types**: A list of valid attributes that can be defined for controllable assets of this type. Ad-hoc or predefined characteristic types linked to the "Asset Attributes" characteristic entity can be defined as asset attribute types. Values for these attributes should be defined in the **Asset Attributes** zone of the **Asset** portal.
- **Engineering Specification Attribute Types**: A list of valid engineering specification attributes that can be defined for controllable assets of this type. Ad-hoc or predefined characteristic types linked to the "Specification Attributes" characteristic entity can be defined as engineering specification attribute types. Values for these attributes should be defined in the **Specification Attributes** zone on the **Specification** portal.
- **Access Control**:
  - **Owning Organization**: The access group to which controllable assets of this type belong.
  - **Instance Organization**: Optional additional access group to which controllable assets of this type belong.

System Data

Controllable asset types are based on the system data:

Data Type	Description
Maintenance Object	Asset Type (W1-ASSETTYPE)
Business Object(s)	Controllable Asset Type (X1-ControllableAssetType)

Understanding Device Types

Device types define information about a class of controllable devices, including properties that apply to all devices of a given type.

Device types are created from corresponding controllable asset types. When a controllable asset type is created (or updated), a corresponding device type is created (or updated).

Controllable device types are defined by the following:

- **Controllable Device Business Object**: The business object used to create controllable devices of this type.
- **Device Classification**: The general class to which controllable devices of this type belong.
- **Category**: The category to which controllable devices of this type belong.
- **Service Type**: The type of service (such as Electric) associated with controllable devices of this type.
- **Track Location**: Flag that indicates if the location of controllable devices of this type is tracked.
- **Attached to**: Flag that indicates to which type of object (service point or meter) controllable devices of this type are attached.

System Data

Device types associated with controllable asset types are based on the system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Device Type (D1-DEVICETYP)
Business Object(s)	Item Type(D1-ItemType)

Distributed Energy Resource (DER) Device Attributes

Registration of DER devices includes populating various attributes of the device and related specifications. These DER attributes, including interconnection attributes, are captured as characteristics whose Characteristic Types are defined for the "Asset Attribute" and "Specification Attribute" characteristics entities. Characteristic types for DER attributes are defined in the DER Attribute Mapping (X1-DERAttributeMapping) extendable lookup. Values in this extendable lookup falls into one of three categories, based on the **Attribute Category**:

- **DER Capability** attributes are read-only values established by the DER manufacturer by design or manufactured configuration. Each DER Device with the same manufacturer and model has the exact same set of DER Capability (nameplate) attributes. An example DER Capability attribute is "modesSupported" (Indicates support for each control mode function). Values for DER Capability characteristics are captured as Specification Attributes (see **Engineering Specification Attribute Types** above).
- **Device information** attributes contain identification and other information about the device that changes very infrequently typically only when updates are applied, if ever. An example Device Information attribute is "mfSerNum" (Serial Number). Values for Device Information attributes are captured as Asset Attributes (see **Asset Attribute Types** above).
- **Interconnection** attributes details define the specific DER settings for a particular customer based on the contractual agreement with the utility. An example Interconnection attribute is "interconnectionType" (Interconnection Type). Values for Interconnection attributes are captured as Asset Attributes (see **Asset Attribute Types** above).
- **Additional Information** attributes are used to capture additional information. Values for Additional Information attributes are captured as either Asset Attributes or Specification Attribute depending on whether they are defined as Asset Attribute Types or Engineering Specification Attribute Types on the asset type.

Implementations can add additional attributes to the DER Attribute Mapping (X1-DERAttributeMapping) extendable lookup if needed. Adding DER Capability and Device Information attributes also requires extending corresponding data areas as outlined in the following table:

Attribute Category	Date Area (Used to define attributes for inbound and outbound messages)
DER Capability	DER Capability attributes (X1-DERCapabilityAttributes)
Device information	DER device information attributes (X1-DERDeviceInfoAttributes)
Interconnection	N/A (Adding additional Interconnection attributes does not require extending any data areas.)
Additional Information	N/A (Adding additional Additional Information attributes does not require extending any data areas.)

Understanding Device Location Types

Copyright © 2026, Oracle and/or its affiliates.

Device location types define the details of locations of a certain type where controllable assets can be installed.

Device location types are defined by the following:

- **Service Type**: The type of service (such as Electric) available at device locations of this type.
- **Device Location Business Object**: The business object used to create device locations of this type.
- **Divisions**: One or more divisions associated with device locations of this type.
- **Valid Controllable Device Types**: A list of one or more controllable device types considered valid to be installed at device locations of this type.

System Data

Device location types are based on the system data:

Data Type	Description
Maintenance Object	Service Point Type (D1-SPTYPE)
Business Object(s)	Device Location Type (X1-DeviceLocationType)

Understanding Head End Systems

Head end systems represent external systems that handle the communication with controllable devices. With Customer Program Management, Oracle Utilities Network Management System can be used as a head end system to facilitate communication with and control of controllable devices.

Head end systems utilize processing methods that specify the types of messages that are processed and sent to the external system.

Head end systems are defined by the following:

- **External Reference ID**: An ID used to reference the head end system in external systems and applications.
- **External System**: The external system associated with the head end system.
- **Our Name in Their System**: The name for Oracle Utilities Customer Cloud Service in the external system.
- **Service Provider Type**: The general type of service provider to which the head end system belongs (default is "Head End System").
- **Processing Methods**: One or more processing methods used to define the format or means by which the head end system receives or sends data from or to the application. Processing methods can also be used to define the information an external system wishes to subscribe to receive from the application.
- **Translation Methods**: One or more methods used for translating data sent between the head end system and Oracle Utilities Customer Cloud Service.

System Data

Head End Systems are based on the system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Service Provider - MDM (D1-SVCPROVDR)
Business Object(s)	Controllable Device Head End System (X1-CtrlDvcHeadEndSystem)

Understanding DER Aggregators

DER (Distributed Energy Resource) aggregators are external systems that facilitate communication between a utility and Distributed Energy Resource (DER) devices and, in some cases control and manage DER devices to provide energy and other services to utilities. Based on the IEEE 2030.5 protocol, utilities typically enter into a contractual agreement with one or more aggregators such that authorized aggregators can offer contracted programs to the end customers on behalf of the utility.

As DER Aggregators are created and updated, aggregator data is synchronized with Oracle Utilities Network Management System. DER Aggregator data synchronization includes contact and program offering information. Note that deletion of DER does **not** trigger data synchronization.

DER aggregators utilize processing methods that specify the types of messages that are processed and sent to the external system.

DER aggregators are defined by the following:

- **Service Provider Type**: The general type of service provider to which the DER aggregator belongs (default is "DER Aggregator").
- **Status**: The current status of the DER aggregator in the system.
- **Start Date**: The date on which the DER aggregator is eligible to provide service.
- **End Date**: The date after which the DER aggregator is no longer eligible to provide service.
- **Contact ID**: The program contact for the DER aggregator.
- **External Reference ID**: An ID used to reference the DER aggregator in external systems and applications.
- **External System**: The external system associated with the DER aggregator.
- **Our Name in Their System**: The name for Oracle Utilities Customer Cloud Service in the external system.
- **Certification Status**: The IEEE 2030.5 certification status of the DER aggregator. Valid values include "Certified", "In Progress", "Not Applied", and "Rejected".
- **Energy Source**: The source of energy provided by the aggregator.
- **DER Aggregator Class**: The general class to which the DER aggregator belongs, such as "Independent System Operator", "Distribution System Operator", or "Both".
- **Program Offering**: One or more programs that the aggregator can offer to end customers on behalf of the utility
- **Processing Methods**: One or more processing methods used to define the format or means by which the DER aggregator receives or sends data from or to the application. Processing methods can also be used to define the information an external system wishes to subscribe to receive from the application.

System Data

Copyright © 2026, Oracle and/or its affiliates.

DER Aggregators are based on the system data:

Data Type	Description
Maintenance Object	Service Provider - MDM (D1-SVCPROVDR)
Business Object(s)	DER Aggregator (X1-DERAggregator)

Understanding Metered Service Types

Metered service types a collection of properties defining a class of metered services. Metered service type records are most often created via data synchronization with Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter) and corresponding to usage subscription types in those systems.

Metered service types are defined by the following:

- **Metered Service Business Object**: The business object used when creating metered service records of this type.
- **Customer Settlement BO**: The settlement transaction business object used for customer settlements for metered service records of this type.
- **Valid Metered Service Point Types**: One or more metered service point types considered valid for metered service records of this type.
- **Attributes**: Characteristic types that may be defined for metered service records of this type.

System Data

Metered service types are based on the following system data:

Data Type	Description
Maintenance Object	Usage Subscription Type (D1-USTYPE)
Business Object(s)	Metered Service Type (X1-MeteredServiceType)

Understanding Metered Service Point Types

Metered service point types define a specific type of point at which service is delivered. Metered service point type are most often created via data synchronization with Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter) and corresponding to service point types in those systems.

Metered service point types are defined by the following:

- **Service Type**: The type of service (such as Electric) available at metered service points of this type.
- **Metered Service Point Business Object**: The business object used when creating metered service points of this type.
- **SP Class**: The class of service points to which metered service points of this type belong (inherited from the business object).
- **Divisions**: One or more divisions associated with metered service points of this type.

System Data

Metered service point types are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Service Point Type (D1-SPTYPE)
Business Object(s)	Metered Service Point Type (X1-MeteredServicePointType)

Understanding Program Control Score Calculation Groups

Program control score calculation groups are collections of program control score calculation rules that are used to calculate device control scores. During the program control score process, the system executes the program control score calculation rules defined in the program control score calculation group referenced on the program subscription. The rules within a program control score calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Program control score calculation groups are associated with specific program subscriptions and program subscriptions types (or both). When assigned to program subscriptions, program control score calculation groups contain the program control score calculation rules to be used to calculate device scores associated with the device locations defined for the program subscription. Program control score calculation groups associated with program subscription types are those groups considered valid for program subscriptions of that type.

Program control score calculation groups are defined by the following:

- **Division**: The division to which the calculation group belongs.
- **Calculation Group Class**: The class to which the program calculation group belongs (default is Program Calculation Group).
- **Program Control Score Calculation Rules List**: One or more program control score calculation rules belonging to the program calculation group

System Data

Program control score calculation groups are based on the following system data:

Data Type	Description
Maintenance Object	Usage Group (D1-USGGRP)
Business Object(s)	Control Score Calculation Group (X1-ProgramCalculationGroup)

Understanding Program Control Score Calculation Rules

Program control score calculation rules are rules that perform device score calculations. Program control score calculation rules are created for a specific program control score calculation group. For example, if you were configuring two program control score calculation groups and both included a specific program control scorecalculation rule, you would need to create two instances of the program control score rule, one for each group.

The base package includes the following program control score calculation rule business objects:

Copyright © 2026, Oracle and/or its affiliates.

Program Calculation Rule	Description
Generic Control Score Calculation Rule (X1-GenericProgramCalcRule)	This is a parent business object for all Program Control Score Calculation Rule business objects, and is used to define the lifecycle and schema of child rule business objects.
Score Calculation Rule (X1-ScoreCalculationRule)	This rule is used to calculate control scores for controllable devices.

System Data

Program control scorecalculation rules are based on the following system data:

Data Type	Description
Maintenance Object	Usage Rule (D1-USGRULE)
Business Object(s)	Generic Control Score Calculation Rule (X1-GenericProgramCalcRule) Score Calculation Rule (X1-ScoreCalculationRule)

Understanding Program Initiatives

Program initiatives define criteria used to determine if customers are eligible to participate in programs.

Program initiatives are defined by the following:

- **Status**: The current status of the initiative.
- **Initiative Eligibility Criteria**: Criteria used to determine if a customer is eligible for a program.

System Data

Program initiatives are based on the system data:

Data Type	Description
Maintenance Object	Initiative (C1-INITIATIV)
Business Object(s)	Digital Asset Initiative (X1-InitiativeDigitalAsset)

Understanding Initiative Eligibility Criteria

A program initiative's eligibility criteria are used to confirm that a customer is eligible to enroll and participate in a program. An initiative may have zero, one, or many eligibility criteria. Each customer is subjected to these criteria to determine whether they are eligible to enroll in the program. Eligibility criteria can perform comparisons with scalar values (such as, a number of days, number of kwh) or with derived values (such as, attributes of a customer).

The Metered Service Criteria (X1-MeteredServiceCrit) business object can be used to define program initiative eligibility criteria. The Derive Value Algorithm Type options on this business object define a number of algorithm types that can be used in comparison operations as part of determining eligibility. Algorithms based on these algorithm types can retrieve data from a customer's metered service or metered service points.

Copyright © 2026, Oracle and/or its affiliates.

The base package includes the following algorithm types and algorithms for eligibility criteria comparisons:

Algorithm Type	Algorithm
Retrieve Metered Service Characteristic (X1-ICGETSCEL)	Retrieve Metered Service Characteristic - Active Service (X1-ICGETASVC)
Retrieve Metered Service Point Attribute (X1-ICGETSPEL)	Retrieve Locations Life Support / Sensitive Load (X1-ICGETLSSL)
	Retrieve Locations Division (X1-ICGETDIV)
Evaluate Rate Eligibility (X1-ICGETRATE)	Return true if rate is eligible for program (X1-ICGETRATE)
Retrieve Metered Service Attribute (X1-ICGETMSEL)	No base package algorithms

To set up an eligibility criterion, you must define the following:

- When the eligibility criterion should be executed, known as the **Criteria Execution Point**
- The field to be compared
- The comparison method

The **Criteria Execution Point**, for program initiative eligibility is "Enrollment", meaning this criteria is used when evaluating if a customer is eligible for enrollment in a program.

For the field to be compared, a **Criteria Field Algorithm** is used to:

- Retrieve a field value or values from the customer's metered service or metered service points, such as
- Derive a value based on some data from someplace else in the system (such as whether or not the customer's rates is eligible for the program)

For the comparison method, you select the **Operator** used to compare the retrieved value to a **Comparison Value**. The result from the comparison will be TRUE, FALSE, or INSUFFICIENT DATA. Each of these results should specify the next action. The valid actions include:

- Indicate that the customer is eligible
- Indicate that the customer is not eligible
- Check the next condition
- Issue an error (this option is only available for Insufficient Data)

For example, a program's enrollment eligibility criteria might include only allowing customers who are not on life support or have what is considered to be a "sensitive load" (designated by a value of "Yes" in the **Life Support / Sensitive Load** field on the device location), and whose rate is eligible for program. In other words, if the **Life Support / Sensitive Load** field on the device location is set to "Yes" or the customer's rate is not defined as eligible for the program then the customer is not eligible, Eligibility criteria for this might look like the following:

Copyright © 2026, Oracle and/or its affiliates.

Seq.	Description	Criteria Field Algorithm	Comparison Value	If True	If False	If Insufficient Data	Criteria Execution Point
10	Life Support	Retrieve Locations Life Support / Sensitive Load	Y	Not Eligible	Check Next Condition	Error	Enrollment
20	Eligible Rate	Return true if rate is eligible for program	true	Eligible	Not Eligible	Error	Enrollment

System Data

Initiative Eligibility Criteria are based on the system data:

Data Type	Description
Maintenance Object	Initiative Criteria (C1-INTV-CR)
Business Object(s)	Metered Service Criteria (X1-MeteredServiceCrit)

Understanding Program Contact Types

Program contact types define the properties of a class of program contacts (such as businesses or persons).

Program contact types are defined by the following:

- **Contact Use**: Flag that indicate whether contacts of this type are the primary customer contact and point of contact ("Customer and Point of Contact") or just the point of contact ("Point of Contact Only").
- **Contact Type Business Object**: The business object used when creating program contacts of this type.

System Data

Program contact types are based on the system data:

Data Type	Description
Maintenance Object	Contact Type (D1-CONTTYPE)
Business Object(s)	Contact Type (D1-ContactType)

Understanding Program Subscription Types

Program subscription types define properties defining a class of program subscriptions. Program subscription types also control valid values for various attributes of program subscriptions.

Program subscription types are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Program Subscription Business Object**: The business object used when creating program subscriptions of this type.
- **Program Control Score Business Object**: The business object used when creating program control scores for program subscriptions of this type.
- **Program Settlement BO**: The business object used when creating program settlements for program subscriptions of this type.
- **Valid Device Location Types**: A list of one or more device location types considered valid for program subscriptions of this type.
- **Valid Transaction Calculation Groups**: A list of one or more program calculation groups considered valid for program subscriptions of this type.
- **Fallback Transaction Calculation Groups**: A list of one or more effective dated program calculation groups used if a program calculation group is not defined on the program subscription..
- **Valid Program Types**: A list of one or more program types considered valid for program subscriptions of this type.

System Data

Program subscription types are based on the system data:

Data Type	Description
Maintenance Object	Usage Subscription Type (D1-USTYPE)
Business Object(s)	Program Subscription Type (X1-ProgramSubscriptionType)

Understanding Program Rules

Program rules are used to define specific rules and parameters used by programs and program calculation rules. A program can have one or more associated program rules. If a program has more than one rule, the system uses the rule with the latest effective date/time.

Program rules are defined by the following:

- **Rule Type**: The type of rule.
- **Effective Date**: The date on which the rule is considered to be in effect. Only rules that are currently in effect can be used for score calculations.
- **Expiration Date**: The date on which the rule expires. Expired rules cannot be used for score calculations.
- **Rule-Specific Parameters**: Parameters specific to individual program rules.

System Data

Program rules are based on the following system data:

Data Type	Description
Maintenance Object	Rule (D1-RULE)
Business Object(s)	Device Score Rule (X1-ProgramRules) Incentives and Fees Rule (X1-IncentivesFeesProgramRule)

- Device Score Rules

Copyright © 2026, Oracle and/or its affiliates.

- Incentives and Fees Rules

Device Score Rules

Device score rules are used to calculate device control scores, and use the following parameters:

- **Common Program Rule Parameters**: See Understanding Program Rules.
- **Standard Score Parameters**: Details about device score calculation cycles, including:
  - **Cycle Frequency**: The frequency of score calculation cycle periods. Can be either "Weekly" or "Monthly".
  - **Week Start**: The day of the week on which each week is considered to start, applicable to "Weekly" cycle frequency.
  - **Participation Limit**: The maximum number of times a device can successfully participate in an event in a cycle. The **Score Calculation Factors** section should have values defined for a number of cycles equal to this.
  - **Participation Duration**: The maximum cumulative duration of a device's successfully event participation in a cycle.
  - **Duration Between Participation**: The minimum duration between successful event participation for a device in a cycle.
  - **Max Overrides**: The maximum number of manual overrides allowed per enrolled device
  - **Max Consecutive Days**: The maximum number of days the enrolled device can be consecutively called upon
  - **Max Cumulative Event Duration**: The maximum duration of participation across all events for the enrolled device in HH:MM:SS format.
  - **Allow Event Days**: The allowed day types ("All Days", "Weekday", or "Weekend") on which the enrolled device can be called upon.
  - **Allow On Holidays**: A flag to indicates ("Yes" or "no") whether enrolled device can be called on a Holiday
  - **Holiday Calendar**: The Work Calendar used to define holidays. See **Defining Work Calendar** for more information about setting up work calendars.
  - **Start Hour From**: The earliest start hour of the day at which the event can be initiated. Applies to Programs that can only be run during certain hours of the day.
  - **Start Hour To**: The latest start hour of the day at which the event can be initiated. Applies to Programs that can only be run during certain hours of the day.
  - **End Hour From**: The earliest end hour of the day that can be assigned to an event. Applies to Programs that can only be run during certain hours of the day.
  - **End Hour To**: The latest end hour of the day that can be assigned to an event. Applies to Programs that can only be run during certain hours of the day.
  - **Start Time Offset**: Allowable delay between a scheduled event start time and the actual time at which enrolled devices initiate a response.
- **Score Calculation Factors**: Details used in score calculations, including:
  - **Cycle**: The cycle in in which a device is currently participating. Calculated as the number of the device's successful event participations plus 1. There should be a

Copyright © 2026, Oracle and/or its affiliates.

number of cycles defined equal to the value of the **Participation Limit** in the **Standard Score Parameters** section.

- **Activation Cost**: The activation cost associated with the cycle.
- **Burn Rate Factor**: The burn rate factor associated with the cycle.
- **Score Factor**: The score factor associated with the cycle.

When device sore rules derive values from the **Score Calculation Factors** section, values are defined per **Cycle**, which is the number of times the device has successfully participated in an event during the current cycle (week or month), plus 1. A device's event participation is considered successful in the following circumstances:

- The Event Participation has an **Outcome** of "Successful Participation".
- The Event Participation has an **Outcome** of "Canceled". This happens when a device has been reserved but the event stopped halfway through due to reasons such as, target load control reached. It is considered as a success in customer perspective, since it is the utility that stopped the event which resulted to a cancelled outcome.
- The Event Participation has a blank **Outcome** and the device was not preemptively opted out of the event (**Preemptive Opt-Out** is "No" or blank). Posting of an event's outcome may be delayed, and until then the event participation of a reserved device is considered as a success as long as the customer did not opt out before the event transpires.

For instance, if a device has never participated in an event, the algorithms would use the values in the Cycle 1 row. If a device has successfully participated in 3 previous events, the algorithms would use the values in the Cycle 4 row.

If a Device Score rule is added to program, or an existing Device Score Program Rule associated with a program is edited, a synchronization request and outbound message are created to update external DERMS systems such an Oracle Utilities Grid Edge DERMS.

Incentives and Fees Rules

Incentives and fees rules are used to calculate incentives and fees related to a program. Incentives include sign-up incentives for enrollments, referral rebates, and incentives for remaining enrolled during the peak season. Penalties or fees are applied in certain circumstance, for instance if they terminate their enrollments before the contracted period, or when the utility incurs charges related to device repairs, or device upgrades. Incentives and Fees rules should define parameters for all of the incentives and/or fees applicable for the program.

These use the following parameters:

- **Common Program Rule Parameters**: See Understanding Program Rules.
- **Rule Parameters**, including:
  - **Sequence**: The sequence of the rule.
  - **Category**: The category for the incentive and fee such as **Fixed Rebate, Enrollment Signup, Early Termination Fee** and others. Category values are defined using the Incentive and Fee Type Category (X1-IncentiveFeeTypeCatLookup) extendable lookup.
  - **Calculation Group**: The program settlement calculation groups that contains the program settlement calculation rules for the incentive and fee.
  - **Calculation Frequency**: The frequency of an incentive and fee category. Cyclical fixed rebates may be set as **Monthly, Quarterly or Annual**. Enrollment Signup and Installation Rebate may be set as **One Time** . Service Charge and Early Termination

Copyright © 2026, Oracle and/or its affiliates.

Fee may be set as **Ad hoc**. Frequency values are defined using the Incentive and Fee Calculation Frequency (X1-IncentiveFeeCalcFreqLookup) extendable lookup.

- **Settlement Frequency**: This parameter is currently not used.
- **Line External Id**: An identifier or code that can be used by the external system that receives the calculated incentive and fee line amount. These values can be mapped to the distribution code of general ledger entry for the line amount. Values are defined using the Line Identifiers (X1-LineIdentifiersLookup) extendable lookup.
- **Seasonal**: Flag that indicates if the incentive or fee will be calculated only for a particular season.
- **Season Start/Season End**: The start and end date of the season. This applies only if the Seasonal parameter is set to "Yes". Entered as **MM-DD** (for month and day), displayed as **MMDD** .
- **Eligibility Algorithm**: An algorithm used to determine if the program subscription is eligible for the incentive and fee. See **Incentive and Fee Eligibility** below for more information.
- **Minimum Days**: A number of days added to the transaction end date of the latest calculated incentive and fee type to determine if next transaction for the same category may be calculated. applies to cyclical calculation.

Incentive and Fee Eligibility

The rules and conditions under which a program subscription is considered eligible for a given incentive or fee are implementation-specific, and can be based on information on a wide variety of records, such as the program subscription, a related device location or controllable asset or device, or the existence of work order activities for customer. The base package does not include algorithms for determining eligibility of incentives and fees. Implementations must configure their own algorithms based on their business requirements and specify those algorithms on the appropriate Rule Parameter.

Understanding Program Types

Program types define properties belonging to a class of program.

Program types are defined by the following:

- **Dynamic Option Class**: The class of program to which programs of this type belong.
- **Program Category**: The category to which programs of this type belong. Options include:
  - Control Group
  - DER Registration
  - Device Control
  - In Home Energy Assessment
  - Rate-Based
- **Device Required**: Flag that indicates (Yes or No) if programs of this type require controllable devices.
- **Location Required Source**: Flag that indicates (Yes or No) if programs of this type require a location. Most programs for which a service of some type is performed (such as installation of a device or a home energy assessment) require a location. An example of a program type that does not require a location is a Critical Peak Pricing (CPP) program.

Copyright © 2026, Oracle and/or its affiliates.

Note that some steps such as creation of a device location and device are skipped in enrollment processing for programs based on a type where this is set to "No".

- **Program Business Object**: The business object used when creating programs of this type.
- **Program Event Business Object**: The business object used when creating program events for programs of this type.
- **Event Participation**: The business object used when creating event participation records for programs of this type.
- **Time Zone**: The time zone associated with programs of this type.
- **Settlement Group Determination**: An optional algorithm used to determine the appropriate Program Settlement Calculation Group to use when calculating program settlement transactions for programs of this type. For example, the base package include the Determine Program Usage Group by Date (X1-DETPTUGRP) algorithm, which can specify whether to use an program settlement transaction's Start Date or End Date when determining which calculation group to use.
- **Event Opt Out Parameters**: Parameters that define if customers can "opt out" of events for programs of this type, including:
  - **Allow To Opt Out**: A flag that indicates (Yes or No) if if customers can "opt out" of events for programs of this type
  - **Opt Out Threshold Minutes**: The minimum number of minutes before an event takes place by which a customer may opt out of events for programs of this type
  - **Eligible Categories**: A list of Event Categories for the types of events from which customers can opt out for programs of this type
- **Related Program Type**: A list of one or more related program types. Not applicable if "DER Registration" is selected as the **Program Category**.
- **Program Quantity Types**: A list of one or more valid program quantity types for programs of this type. Available program quantity types are based on values defined in the Program Quantity Type (X1-ProgramQuantityType) extendable lookup.

System Data

Program types are based on the system data:

Data Type	Description
Maintenance Object	Dynamic Option Type (D1-DOPTYPE)
Business Object(s)	Program Type (X1-ProgramType)

Setting up Program Quantity Types

Program quantity types available for program types and programs are defined in the Program Quantity Type (X1-ProgramQuantityType) extendable lookup.

Program quantity types are defined by the following:

- **Program Quantity Type**: A code for the program quantity type
- **Description**: A description of the program quantity type
- **Status**: The current status of the program quantity type
- **Service Type**: The type of service (such as "Electric" or 'Water") associated with the program quantity type

Copyright © 2026, Oracle and/or its affiliates.

- **UOM**: The unit of measure for quantities of the program quantity type
- **TOU**: The time of use period for quantities of the program quantity type
- **SQI**: The service quantity identifier for quantities of the program quantity type
- **Category**: The category to which the program quantity type belongs. To add other Program Quantity Type Category values, add new Characteristic Values to the Program Quantity Type Category (X1-RCCAT) characteristic type.

Understanding Program Settlement Calculation Groups

Program settlement calculation groups are collections of program settlement calculation rules that are used to calculate program settlement transactions, which represent credits that will be applied to a customer's bill based on participation in an individual event (see About Program Settlement Transactions). The rules within an program settlement calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Program settlement calculation groups are associated with specific programs. When assigned to programs, program settlement calculation groups contain the program settlement calculation rules to be used to calculate settlements associated with events for the program.

Program settlement calculation groups are defined by the following:

- **Division**: The division to which the calculation group belongs.
- **Calculation Group Class**: The class to which the program settlement calculation group belongs (default is Program Settlement Calculation Group).
- **Transaction Type**: A flag that indicates the type of transaction calculated by the program settlement calculation group. This allows configuration of multiple groups to support different types of calculations required for the same Event Settlement Method (After Event Participation or End of Season)., Values for this flag are defined as values for the Settlement Transaction Type (X1-STTRT) Characteristic Type.
- **Program Settlement Calculation Rules List**: One or more program settlement calculation rules belonging to the program calculation group

System Data

Program settlement calculation groups are based on the system data:

Data Type	Description
Maintenance Object	Usage Group (D1-USGGRP)
Business Object(s)	Calculation Group Root (D1-UsageGroupRoot) - Parent Business Object Program Settlement Calculation Group (X1-EventSettlementCalcGroup)

Understanding Program Settlement Calculation Rules

Program settlement calculation rules are rules that perform program event settlement calculations and create program event settlement transactions, which represent credits that will be applied to a customer's bill based on participation in an individual event (see About Program Event Settlement Transactions). Program settlement calculation rules are created for a specific program settlement calculation group. For example, if you were configuring two program settlement calculation groups and both included a specific program settlement calculation rule,

Copyright © 2026, Oracle and/or its affiliates.

you would need to create two instances of the program settlement calculation rule, one for each group.

The base package includes the following program calculation rule business objects:

Event Settlement Calculation Rule	Description
End of Event Program Settlement Rule (X1-EventSettlementCalcRule)	This rule is used to calculate settlement results for consumption-based ("kWh Avoided") events.
End of Season Program Settlement Rule (X1-EventSetCalcRuleDemandBased)	This rule is used to calculate settlement results for demand-based ("kW Drop") events.
Incentive and Fee Calculation Rule (X1-IncentiveCalculationRule)	This rules is used to calculate program incentives and fees.
Baseline Calculation (D2-BaselineCalc)	This rule is used to calculate estimated baseline consumption for a metered service point that represents an estimate of the customer's consumption or demand for an event period had they not participated in the event. The Apply Eligibility Criteria - Check Optional UOM/TOU/SQI (D2-UTS-AEC) can be used to deetermine if a particular UOM/TOU/SQI in the rule configuration before executing the rule.
Vector and Service Quantity Math (D2-Math)	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations. This can be used to calculate reductions in consumption or demand by subtracting actual consumption from baseline consumption calculated using the Baseline Calculation rule.
Execute Usage Calculation Group (D1-UsageRuleReferredUsageGroup)	This rule is used to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. For example, this can be used to group a Baseline Calculation rule and Vector and Service Quantity Math rule to calculate reductions in consumption or demand, which can then be passed to an End of Event Program Settlement Rule or End of Season Program Settlement rule.

System Data

Program settlement calculation rules are based on the following system data:

Data Type	Description
Maintenance Object	Usage Rule (D1-USGRULE)

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Business Object(s)	End of Event Program Settlement Rule (X1-EventSettlementCalcRule) End of Season Program Settlement Rule (X1-EventSetCalcRuleDemandBased) Incentive and Fee Calculation Rule (X1-IncentiveCalculationRule) Baseline Calculation (D2-BaselineCalc) Vector and Service Quantity Math (D2-Math) Execute Usage Calculation Group (D1-UsageRuleReferredUsageGroup) Generic Usage Calculation Rule (D1-GenericUsageRule)

Eligibility Criteria

Eligibility criteria are user-definable conditions that could cause a given rule rule to be applied or skipped. This can involve the evaluation of some attribute of the program subscription or metered service point, or something else entirely. Prgram settlement calculation rules can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

System Data

Eligibility criteria are based on the following system data:

Data Type	Description
Maintenance Object	Usage Calculation Rule Eligibility Criteria (D1-USGRLELIG)
Business Object(s)	Usage Eligibility Criteria (D1-UsgRuleEligibilityCriteria) Apply Eligibility Criteria - Check Optional UOM/TOU/SQI (D2-UTS-AEC)

Understanding Usage Transaction Exception Types

Usage Transaction Exception Types define the groupings of exceptions for a Usage Transaction based on their functional similarity. This provides a way to define Usage Transaction Exceptions in a distinct enough way to understand the root issue that was generated from the usage calculation rule.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section.

Understanding Interval Price Set Types

Interval price set types define properties belonging to a class of interval price sets.

Interval price set types are defined by the following:

- **Service Type**: The service type to which interval price sets and interval prices based on this type will be applied
- **Interval Size**: The size of the intervals for interval prices based on this type, represented as hours:minutes:seconds (HH:MI:SS).

Copyright © 2026, Oracle and/or its affiliates.

- **UOM**: The unit of measure used to define interval price sets and interval prices based on this type
- **TOU**: The time of use period used to define interval price sets and interval prices based on this type
- **SQI**: The service quantity identifier used to define interval price sets and interval prices based on this type
- **Short-Hand Description**: A short-hand description used for interval price sets of this type

> **Note**
>
> Either a UOM or SQI **must** be defined when creating an interval price set type.

System Data

Interval Price Set Types are based on the following system data:

Data Type	Description
Maintenance Object	Measuring Component Type (D1-MCTYPE)
Business Object(s)	Interval Price Set Type (D1-IntervalPriceType)

Understanding Customer Questions

Customer questions are used to capture specific information related to a customer or their service. Questions are associated with programs to help determine a customer's eligibility for that program.

Customer questions are defined by the following:

- **Question**: A name or code for the question.
- **Question Text**: The specific text of the question (such as "Do you have Internet service?").
- **Answer Style:** The data type for responses to the question (such as checkbox, radio button, date or date/time, text input, or pre-defined characteristic type). This also controls how valid answers are rendered in the user interface. Refer to the embedded help for additional details.
- **Status**: The current status of the question.
- **Valid Answers**: One or more answers to the question, including the sequence (if applicable), a code for the answer (such as "Y" or "N"), answer text (such as "Yes" or "No"), if different from the code, and status.

System Data

Customer questions are based on the system data:

Data Type	Description
Maintenance Object	Question - CCB (C1-QSTN)
Business Object(s)	Customer Question (C1-Question)

Understanding Service Areas

Copyright © 2026, Oracle and/or its affiliates.

Service areas define a logical boundary of your organization's service territory. A service area can be specified on a metered service point to indicate the service area in which the metered service point is located.

Crew shifts are populated with a list of service areas that the crew shift is allowed to work. This is matched up with the service area indicated on the activity (from the metered service point) to help determine which crew shift can work the activity.

System Data

Service areas are based on the system data:

Data Type	Description
Maintenance Object	Service Area (W1-SVCAREA)
Business Object(s)	Service Area (W1-ServiceArea)

Understanding Service Classes

Service classes are used to categorize different types of work.

Service classes are defined by the following:

- **Service Class Type** : A broader category of the type of work, such as Maintenance, Renewal, Construction, and and so on.
- **CWIP Cost Center**: Not applicable for Customer Program Management.

System Data

Service classes are based on the system data:

Data Type	Description
Maintenance Object	Service Class (W1-SVCCLASS)
Business Object(s)	Service Class (W1-ServiceClass)

Understanding Service Codes

Service codes are used to define sets of questions used with service calls. Service codes are referenced on service task types, and used to display customer questions on service call process flows.

Refer to Understanding Customer Questions for more information about customer questions.

Service codes are defined by the following:

- **Questions**: The sequence of questions that comprise the service code.

System Data

Service classes are based on the system data:

Data Type	Description
Maintenance Object	Service Code - CCB (C1-SVCCODE)
Business Object(s)	Program Service Call Service Code (X1-ProgramSvcCallServiceCode)

Copyright © 2026, Oracle and/or its affiliates.

Understanding Service Types

Service types define specific types of service used by and with controllable assets, such as electric, gas, water, and so on.

System Data

Service types are based on the system data:

Data Type	Description
Maintenance Object	Service Type - MDM (D1-SVCTYPE)
Business Object(s)	Service Type (D1-ServiceType)

Understanding Units of Measure

Units of Measure (UOM) identify quantities measured and recorded, such as KWH, KW, cubic feet, degrees Celsius, etc. UOMs are based on a specific service type. When used to define interval prices, units of measure might include currency denominations such as dollars or cents.

System Data

Units of measure are based on the following system data:

Data Type	Description
Maintenance Object	Unit Of Measure - MDM (D1-UOM)
Business Object(s)	Unit Of Measure (D1-UnitOfMeasure)

Important Unit of Measure System Events

Unit of Measure supports the following business object algorithm system events:

- Axis Conversion: this event receives a list of measurements along with a source UOM and interval size and a target UOM and interval size. It should then perform the necessary actions to convert the source UOM and interval size into the target UOM and interval size. Refer to the algorithm Axis Conversion algorithm (D1-AXIS-CONV) as an example.

  When used with Oracle Utilities Customer Program Management Cloud Service and Oracle Utilities Customer Program Management, axis conversion is most commonly used to convert “kWh avoided” interval values to match the interval size of the interval prices used in program settlement calculations.

Understanding Time of Use Periods

Time of Use (TOU) periods are modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

System Data

Time of use periods are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Time Of Use - MDM (D1-TOU)
Business Object(s)	Time Of Use (D1-TimeOfUse)

Understanding Service Quantity Identifiers

Service Quantity Identifiers (SQI) are used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Some examples include: Generated, Consumed, etc. When used to define interval prices, service quantity identifiers might represent things like Market Prices or Real-Time Prices.

SQIs can also be used as a stand-alone representation of a service quantity that is not measured (one that is not properly described as a UOM) within a usage service quantity collection (such as a billing determinant).

System Data

Service quantity identifiers are based on the following system data:

Data Type	Description
Maintenance Object	Service Quantity Identifier - MDM (D1-SQI)
Business Object(s)	Service Quantity Identifier (D1-ServiceQuantityIdentifier)

Understanding Divisions

Divisions are used to delineate between different operating companies within a large conglomerate of utilities. Divisions can have one or more characteristics defined.

Divisions can be defined on several objects and entities in the system, including:

- Device Location Types
- Device Locations
- Metered Service Point Types
- Metered Service Points
- Programs
- Program Calculation Groups

System Data

Divisions are based on the system data:

Data Type	Description
Maintenance Object	Division (D1-DIV)
Business Object(s)	Division (D1-Division)

Understanding Entity Groups

Copyright © 2026, Oracle and/or its affiliates.

Entity Groups are used to define groups of entities, such as devices (including meters or items, as well as controllable devices used with Customer Program Management Cloud Service and Customer Program Management) or service points.

An entity can belong to more than one group if they have different Group Categories. For example, a controllable device associated with a specific program and rate could belong to a group category based on the specific program as well as a group category based on the specific rate.

Entity group processing uses dynamic SQL queries to identify entities that belong to a entity group. These dynamic queries are based on configuration of the Group Category (which defines Template SQL that contains tokens) and Criteria Values (which define specific values for the tokens in the Template SQL). These queries are used by "Eligible <Entity>s for this Group" zones that identify entities eligible for but not currently assigned to a group, as well as <Entity> Group Evaluation batch processes (see **Batch Processing** in **Configuring Entity Groups** for more information).

Entity Groups are defined by the following:

- **Status**: The current status of the entity group
- **Group Category**: The category to which the entity group belongs. Group categories are defined using the Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup. See below for more information.
- **Group Deactivation Option**: A flag that defines what happens to the relationship (captured in the Device Entity Group or Service Point Entity Group table, as appropriate) between the group and entities belonging to the group if or when the group is deactivated. Options include "Expire" and "Do not Expire".
- **Group Assignment Option**: A flag indicating the manner in which groups are assigned to entities. Options include System (via batch process), User (manually), System and User (either).
- **Criteria Value**: One or more criteria values used for selecting entities for the group, based on a Criteria Source and Criteria Value.
  - **Criteria Source**: Defines the source for criteria used for selecting entities for the group (such as a Specification, Rate, or Program). Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See below for more information.
  - **Criteria Value**: Defines specific values for the criteria that are used when selecting entities for the group. For instance, values for a **Criteria Source** of "Rate" could be rates defined in an extendable lookup.

System Data

Entity Groups are based on the following system data:

Data Type	Description
Maintenance Object	Entity Group (D1-ENTGRP)
Business Object(s)	Entity Group (D1-EntityGroup)

Entity Group Category Extendable Lookup

The Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup defines categories of entity groups. These are used to define groups of entities (either devices or service points) that share the criteria and template SQL for identifying and retrieving entities but that may have different criteria values. For example, two device entity groups, "Device

Copyright © 2026, Oracle and/or its affiliates.

Group A" and "Device Group B" can reference the same "Specification and Rate" category, which uses Specification and Rate Schedule criteria to identify and retrieve devices (devices in group A are those with a unique combination of Specification and Rate Schedule, while those in group B have a different unique Specification and Rate combination). The category also defines template SQL that uses the "Specification" and "Rate" criteria when identifying and retrieving devices. The Entity Group Category Template SQL contains tokens, and an Entity Groups Criteria Values define specific values for the tokens in the Template SQL.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group category.
- **Target Maintenance Object**: The maintenance object of the entities that belong to entity groups of this category. Valid options include Device (D1-DEVICE) or Service Point - MDM (D1-SP).
- **Criteria Option**: Flag that indicates whether or not the criteria source along with the SQL template and SQL builder algorithm will be recorded for the entity group in this category. Valid values are "Applicable" and "Not Applicable".
- **Template SQL**: The template SQL used to identify and retrieve entities for entity groups of this category. Applicable only when Criteria Option is set to "Applicable".
- **Entity Group Algorithm**: An algorithm that executes the template SQL, inserting the criteria values from the entity group, to identify and retrieve entities for entity groups of this category.
- **Criteria Sources**: The criteria sources used by the template SQL when identifying and retrieving entities for entity groups of this category. Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See below for more information. Applicable only when Criteria Option is set to "Applicable". Each criteria source is defined by the following:
  - **Criteria Source**: The name of the criteria source (from the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup)
  - **Is Required**: A flag that indicates if a value is required for the criteria source (Required or Optional)
  - **Marker**: The token used in the template SQL for the criteria source and value, in the format of "#xxx". For example, the marker for a Specification criteria source might be "#SPEC".
  - **Condition Operator**: The operator used in `WHERE` clauses in the template SQL to filter data based on the criteria source. Values are defined in the Condition Operator (CONDITION_OPR_FLG) lookup.

The base package includes the "SP Measurement Extract" value for this extendable lookup.

Criteria Source Extendable Lookup

The Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup defines the criteria that can be configured for an entity group. Each value defines a single criteria source, such as Specification, Rate, or Program, as well as how the user will enter criteria values for the criteria source, and the source of criteria values for the criteria source. This configuration is used in the **Criteria Value** section of an Entity Group.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group criteria source
- **Presentation**: The presentation used for the criteria source. Options include:
  - **Drop down**: The user selects a the criteria value from a drop-down list.

Copyright © 2026, Oracle and/or its affiliates.

  - **Input Text**: The user enters text.
  - **Input Text with Date Picker**: The user selects a date.
  - **Input Text with Date Time Picker**: The user selects a date and time.
- **Value Source**: The source of values for the criteria source. Options include:
  - **Characteristic Type (Pre-Defined)**: Values are from a list of pre-defined values for a characteristic type
  - **Extendable Lookup**: Values are from an extendable lookup
  - **Service Script**: Values are retrieved by a service script
  - **Standard Lookup**: Values are from a standard lookup
- **Value Retriever**: The specific value source used for the criteria source, based on the Value Source. For example, if the Value Source is "Standard Lookup", this would be the lookup code.

Example: A "Rate" criteria source where values are derived from the Customer Rate Schedule (D2-CCBRateScheduleLookup) extendable lookup could be defined as follows:

- **Criteria Source**: Rate
- **Description**: Rate
- **Status**: Active
- **Presentation**: Drop down
- **Value Source**: Extendable Lookup
- **Value Retriever**: D2-CCBRateScheduleLookup

Understanding Weather Stations

Weather stations are facilities that measure atmospheric conditions and collect weather data such temperature, humidity, wind speed and direction, and barometric pressure.

Weather stations are defined by the following:

- **Weather Station**; An ID for the weather station.
- **Weather Meter**: The weather meter (see below) that captures weather data for the weather station.
- **Status** A flag that indicates if the weather status is Active or Inactive.
- **Geographic Longitude**: The geographic longitude of the location of the weather station.
- **Geographic Latitude**: The geographic latitude of the location of the weather station.
- **External ID**: An ID used for the weather stations in external systems, such as a distributed energy resource management (DERM) system.

Weather Meters

Weather meters are virtual devices used to store weather data for a weather station within the application. Weather meters can have one more interval measuring components, each configured to store a specific type of weather data (temperature, wind speed, humidity, and so on).

Weather meters are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Device Type**: The device type for the weather meter. This should be based on the Weather Meter Type (D1-WeatherMeterType) business object.
- **Status**: The current status of the weather meter (Active or Inactive)
- **Serial Number**: An identifier for the weather meter. This is based on the ID of the Weather Station.

Like other devices, weather meters require device types. Device types for weather meters are created based on the Weather Meter Type (D1-WeatherMeterType) business object. In addition, weather meters have related device configurations based on the Device Configuration (D1-DeviceConfiguration) business object, and measuring components based on the Interval Channel (D1-IntervalChannel) business object.

See Creating Weather Metersfor more information about how weather meters are created.

Understanding Postal Code Defaults

Postal Code Defaults are used to define ranges of postal codes associated with a weather station. These are used when mapping service points (including metered service points) to weather stations when the longitude and latitude of the service point is not available.

Postal code defaults are defined by the following:

- **Postal Code From**: The starting point of the range of postal codes for the postal code default.
- **Postal Code To**: The ending point of the range of postal codes for the postal code default.
- **Country**: The country in which the postal code default resides.
- **City**: The city in which the postal code default resides.
- **County**: The county in which the postal code default resides.
- **State**: The state in which the postal code default resides.
- **Version**: The version of the postal code default.
- **Time Zone**: The time zone in which the postal code default resides.
- **Weather Station**: The weather station associated with the postal code default.

Understanding M&V Options

This section describes administrative options related to M&V processing used with Oracle Utilities Customer Program Management Cloud Service and Oracle Utilities Customer Program Management.

Configuration Overview

Configuring the system for M&V processing involves creating the following types of administrative data.

Configure M&V Calculation Groups & Rules

Create M&V calculation groups and rules to perform any needed calculations on regression output quantities. If different types of customers will uses different calculations, you will to create separate M&V calculation groups and rules for each. Refer to Understanding M&V Calculation Groups and Understanding M&V Calculation Rules for more information about these.

Copyright © 2026, Oracle and/or its affiliates.

Configure M&V Subscription Types

Create M&V subscription types for each type of customer for which you'll perform regression analysis and related calculations. Reference the appropriate M&V Calculation Groups on each subscription type. Refer to Understanding M&V Subscription Types for more information about these.

Create Device Population and Control Groups

Create Device Population and Control Groups for the different types of groups of customers for which you plan to run M&V processing. Refer to Understanding Device Population and Control Groups for more information about these.

Configure Dynamic Aggregation

Configure the following objects used with Dimension Scanning and Find Constituents dynamic aggregation processing. Refer to the corresponding sections in Understanding Dynamic Aggregation Options and Configuring Dynamic Aggregation Options for more information about these objects.

- Data Sources
- Measuring Component Types
- Aggregation Groups
- Measuring Component Sets
- Understanding Device Population and Control Groups
- Understanding M&V Subscription Types
- Understanding M&V Calculation Groups
- Understanding M&V Calculation Rules

Understanding Device Population and Control Groups

Device Population and Control groups are entity groups that represent groupings of customers used in M&V processing. Device Population groups are "device" entity groups and are groupings of customers who participated in program events, while Control groups are "service point" entity groups that are groupings of customers that do not participate. These groups are used when aggregating data for each type of group.

In addition to the standard attributes for entity groups, Device Population groups have a corresponding Control group.

M&V Subscriptions are created whenever a new Device Population group is created or changed. A M&V Subscription's Device Population group defines the group of customers whose data is aggregated and processed when Measurement and Verification Transactions are created.

In addition, the Dimension Scanning process creates new aggregation measuring components when new Device Population groups are created.

See Understanding Entity Groups and Entity Groups for more information about the data used to define entity groups.

System Data

Device Population and Control Groups are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Entity Group (D1-ENTGRP)
Business Object(s)	Entity Group (D1-EntityGroup) - Parent Control Group (X1-ControlGroup) Device Population (X1-DevicePopulation)

Device Population Synchronization

As Device Population groups are created and updated, the data is synchronized with Oracle Utilities Network Management System via an audit algorithm and outbound message. When the data is synchronized, Oracle Utilities Network Management System sends an acknowledgement and the Device Population group is updated to reflect its synchronization status. The **Sync Status** field on Device Population groups indicates if the group was not successfully synchronized. Click **Send Details** to retry the synchronization.

Enabling Device Population Synchronization

Enabling Device Population data synchronization involves the following steps:

1. Select "Device Population" in the **Entity Group Category** in the Customer Program Management Integration master configuration.
2. Configure the "Device Population" outbound message type in the **Device Population** field in the **Outbound Message Types** section of the Customer Program Management Integration master configuration.
3. Create a Message Sender for the outbound Device Population synchronization.
4. Associate the Device Population Outbound Message Type and Message Sender with the target external system, such as Edge DERMS/DRMS (See Outgoing Messages, Message Sender, and External Systems in the *Framework Administrative User Guide* for more information.

Understanding M&V Subscription Types

M&V subscription types define properties defining a class of M&V subscriptions. M&V subscription types also control valid values for various attributes of M&V subscriptions.

M&V subscription types are defined by the following:

- **Division**: The division in which the customers in Device Population groups for M&V subscriptions of this type reside.
- **Subscription Business Object**: The business object used when creating M&V subscriptions of this type.
- **Transaction Business Object**: The business object used when creating M&V transactions for subscriptions of this type.
- **Subscription Class**: The class to which the M&V subscription belongs (default is Measurement and Verification).
- **Valid M&V Calculation Groups**: A list of one or more M&V calculation groups used when performing calculations for M&V subscriptions of this type

System Data

M&V subscription types are based on the following system data:

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Maintenance Object	Usage Subscription Type (D1-USTYPE)
Business Object(s)	M&V Subscription Type (X1-MsrmtVerifSubscriptionType)

Understanding M&V Calculation Groups

M&V calculation groups are collections of M&V calculation rules that are used to calculate M&V transactions which capture the results of the regression analysis and calculations preformed on those results (see About M&V Transactions). The rules within an M&V calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

M&V calculation groups are associated with specific M&V subscriptions. When assigned to M&V subscriptions, M&V calculation groups contain the M&V calculation rules to be used to calculate M&V transactions associated with events for the program associated with the M&V subscription.

M&V calculation groups are defined by the following:

- **Division**: The division to which the calculation group belongs.
- **Calculation Group Class**: The class to which the M&V calculation group belongs (default is Measurement and Verification Calculation Group).
- **Measurement and Verification Calculation Rules List**: One or more M&V calculation rules belonging to the M&V group

System Data

M&V calculation groups are based on the following system data:

Data Type	Description
Maintenance Object	Usage Group (D1-USGGRP)
Business Object(s)	Calculation Group Root (D1-UsageGroupRoot) - Parent Business Object
	M&V Calculation Group (X1-MsrmtVerificationCalcGroup)

Understanding M&V Calculation Rules

M&V calculation rules are rules that perform M&V calculations and create M&V transactions, which which capture the results of the regression analysis and calculations preformed on those results (see About M&V Transactions). M&V calculation rules are created for a specific M&V calculation group. For example, if you were configuring two M&V calculation groups and both included a specific M&V calculation rule, you would need to create two instances of the M&V calculation rule, one for each group.

The base package includes the following calculation rule business objects used with M&V calculations:

Copyright © 2026, Oracle and/or its affiliates.

M&V Calculation Rule	Description
Vector and Service Quantity Math (D2-Math)	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations.

System Data

M&V calculation rules are based on the following system data:

Data Type	Description
Maintenance Object	Usage Rule (D1-USGRULE)
Business Object(s)	Vector and Service Quantity Math (D2-Math)
	Generic Usage Calculation Rule (D1-GenericUsageRule)

Eligibility Criteria

Eligibility criteria are user-definable conditions that could cause a rule to be applied or skipped. This can involve the evaluation of some attribute of the M&V subscription or metered service point, or something else entirely. A rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

System Data

Eligibility criteria are based on the following system data:

Data Type	Description
Maintenance Object	Usage Calculation Rule Eligibility Criteria (D1-USGRLELIG)
Business Object(s)	Usage Eligibility Criteria (D1-UsgRuleEligibilityCriteria)

Understanding Customer Program Service Task Types

Customer program service task types define properties common to specific types of customer program service tasks used to manage enrollment, unenrollment, and other processes. Each process has an associated service task, which is based on a specific service task type.

Customer program service task types are defined by the following:

- **Service Task Business Object**: The business object used to create service tasks of this type.
- Other details specific to the service task type. See the sections below for more details.

System Data

Customer program service task types are based on the system data:

Data Type	Description
Maintenance Object	Service Task Type (F1-STASKTYPE)

Copyright © 2026, Oracle and/or its affiliates.

Data Type	Description
Business Object(s)	Customer Program Enrollment Task Type (X1-DigitalAssetEnrollTaskType) Customer Program Unenrollment Task Type (X1-DigitalAssetUnenrollTaskTyp) Program Service Call Task Type (X1-ProgramServiceCallTaskType) DER Interconnection Request Task Type (X1-DERInterconnectionTaskType) Customer Program Enroll Additional Device Task Type (X1-DAAddDeviceToEnrollTaskType)

Enrollment Service Task Type

Digital Asset Enrollment service task types include the following

- **Programs**: The programs that will use this service task type.
- **Exception Handling**: Exception handling details, including To Do Type, To Do Role, and Enrollment Expiration Days.
- **Enrollment Setup Defaults**: The Program Subscription Type and Device Location Type for the Program Subscriptions and Device Locations to be created by the enrollment process.
- **Device Setup**: The actions to perform during the enrollment device setup stage, including the following for each:
  - **Sequence**: The sequence for the device setup actions collection. Actions are evaluated in sequence order based on each unique combination of **Installation Method** and **Asset Ownership**. The setup actions whose combination of **Installation Method** and **Asset Ownership** matches that specified in the service task are used in the enrollment process.
  - **Installation Method**: The installation method for the controllable devices associated with the enrollment process. Options include:
    - **Bring Your Own**: The customer supplies their own device.
    - **Re-enroll**: The customer is re-enrolling in a program for a utility-owned device at a different location.
    - **Ship Device**: The utility will ship the device to the customer.
    - **Utility Installation**: The utility will install the device at the customer's device location.
  - **Asset Ownership**: Who owns the assets associated with the enrollment process, based on the Installation Method. Options include:
    - **Customer Owned**: The customer owns the asset/device.
    - **Utility Owned**: The utility owns the asset/device.
  - **Device Setup Actions**: The actions to be performed for installing or confirming the controllable device status at the device location, including the following for each:
    - **Sequence**: The sequence for the actions collection. Actions are processed in order based on sequence.
    - **Setup Action**: The specific action to be performed for devices with this combination of Installation Method and Asset Ownership. Options include:

Copyright © 2026, Oracle and/or its affiliates.

* **Create Activity**: Used to create Service Order Management field activities to perform device installation or inspection. See **Defining Customer Program Service Order Management Options** for more information.
* **Create To Do**: Used to create a To Do Entry.
* **Create Work Order**: Used to create Work and Asset Management work orders to perform device installation or inspection.

> **Note**
>
> This option is not available when using Customer Program Managemeng with Customer Cloud Service.

* **Initiate Controllable Device Command**: Used to create and send commands to the controllable device to test if it's capable of communicating with the control system.
* **No Action Needed**: Indicates that no device setup action is needed for devices with this combination of Installation Method and Asset Ownership.

* **Setup Action Details**: Details for the action, based on the selected **Setup Action**.
  * **Create Activity**: The **Activity Type** and **Field Task Type** for enrollment field activities. See **Defining Customer Program Service Order Management Options** for more information.
  * **Create To Do**: The **To Do Type** and **To Do Role** for the To Do Entry to be created.
  * **Create Work Order**: The **Template Work Order Identifier** for the template work order. Template work orders can create up to two activities. This is used when a customer is enrolling in a related program, in which case an activity template can be configred for each program. Note that template work orders can have a maximum of two activity templates.

> **Note**
>
> This option is not available when using Customer Program Managemeng with Customer Cloud Service.

  * **Initiate Controllable Device Command**: The **Command Activity Type** for commands. Supported commands include "Controllable Device Status Check"

Unenrollment Service Task Type

Digital Asset Unenrollment service task types include the following

- **Programs**: The programs that will use this service task type.
- **Exception Handling**: Exception handling details, including To Do Type, To Do Role, and Unenrollment Expiration Days.
- **Device Setup**: The actions to perform during the unenrollment device setup stage, including the following for each:
  - **Sequence**: The sequence for the device setup actions collection. Actions are evaluated in sequence order based on each unique combination of **Installation Method** and **Asset Ownership**. The setup actions whose combination of **Removal**

Copyright © 2026, Oracle and/or its affiliates.

Method and **Asset Ownership** matches that specified in the service task are used in the unenrollment process.

- **Removal Method**: . The removal method for the controllable devices associated with the unenrollment process. Options include:
  - **Keep**: The customer or utility keeps the device.
  - **Ship Device**: The customer will ship the device to the utility.
  - **Utility Removal**: The utility will remove the device from the customer's device location.
- **Asset Ownership**: Who owns the assets associated with the unenrollment process, based on the Removal Method. Options include:
  - **Customer Owned**: The customer owns the asset/device.
  - **Utility Owned**: The utility owns the asset/device.
- **Device Setup Actions**: The actions to be performed for removing the controllable device status at the device location, including the following for each:
  - **Sequence**: The sequence for the actions collection. Actions are processed in order based on sequence.
  - **Setup Action**: The specific action to be performed for devices with this combination of Renoval Method and Asset Ownership. Options include:
    - **Create Activity**: Used to create Service Order Management field activities to perform device removal. See **Defining Customer Program Service Order Management Options** for more information.
    - **Create To Do**: Used to create a To Do Entry.
    - **Create Work Order**: Used to create Work and Asset Management work orders to perform device removal.

      > **Note**
      >
      > This option is not available when using Customer Program Managemeng with Customer Cloud Service.

    - **Initiate Controllable Device Command**: Used to create and send commands to the controllable device to test if it's capable of communicating with the control system.
    - **No Action Needed**: Indicates that no device setup action is needed for devices with this combination of Removal Method and Asset Ownership.
  - **Setup Action Details**: Details for the action, based on the selected **Setup Action**.
    - **Create Activity**: The **Activity Type** and **Field Task Type** for unenrollment field activities. See **Defining Customer Program Service Order Management Options** for more information.
    - **Create To Do**: The **To Do Type** and **To Do Role** for the To Do Entry to be created.
    - **Create Work Order**: The **Template Work Order Identifier** for the template work order.
    - **Initiate Controllable Device Command**: The **Command Activity Type** for commands. Supported commands include "Controllable Device Status Check"

Copyright © 2026, Oracle and/or its affiliates.

Service Call Service Task Type

Digital Asset Service Call service task types include the following:

- **Escalation To Do Type**: The To Do Type used for To Do Entries created for "Escalation" service calls. The default value for this is X1-TRESC
- **Template Work Order**: A list of available template work orders used with "Create a Work Order" service calls.
- **Questionnaires**: A list of questionnaires that apply to service calls of this type. Questionnaires are maintained as service codes which contain one or more customer questions. See Understanding Service Codes for more information.
- **Exception Handling**: Exception handling details, including To Do Type, To Do Role, and Enrollment Expiration Days.

DER Interconnection Service Task Type

DER Interconnection service task types include the following

- **Device Location Type**: The Device Location Type for the Device Locations to be created by the interconnection process.
- **Specifications**: One or more specifications for assets created by the interconnection process.
- **Exception Handling**: Exception handling details, including To Do Type, and To Do Role

Enroll Additional Device Service Task Type

Enroll Additional Device service task types include the following

- **Exception Handling**: Exception handling details, including To Do Type, and To Do Role.

Configuring Customer Program Management Options

Use the information in this section when defining customer program management options.

- Defining Controllable Asset Types
- Defining Device Location Types
- Defining Head End Systems
- Defining DER Aggregators
- Defining Metered Service Types
- Defining Metered Service Point Types
- Defining Program Control Score Calculation Groups
- Defining Program Control Score Calculation Rules
- Defining Program Initiatives
- Defining Program Rules
- Defining Program Contact Types
- Defining Program Subscription Types
- Defining Program Types

Copyright © 2026, Oracle and/or its affiliates.

- Defining Program Settlement Calculation Groups
- Defining Program Settlement Calculation Rules
- Configuring Usage Transaction Exception Types
- Defining Interval Price Set Types
- Defining Customer Questions
- Defining Service Areas
- Defining Service Classes
- Defining Service Codes
- Defining Service Types
- Defining Units of Measure
- Defining Time of Use Periods
- Defining Service Quantity Identifiers
- Defining Divisions
- Entity Groups
- Weather Stations
- Postal Code Defaults
- Configuring M&V Options
- Defining Customer Program Service Task Types
- Defining Customer Program Service Order Management Options
- Defining Enrollment Options
- Defining Program Service Call Options
- Defining Manual Work Activity Completion Options
- Configuring the Customer Program Summary Portal
- Configuring the Program Dashboard
- Adding Custom Fields to Customer Program Process Flows

Defining Controllable Asset Types

You use the **Asset Type** and **Controllable Asset Type** portals to search for, view, and maintain controllable asset types.

Refer to Understanding Controllable Asset Types for more information.

You can access the **Asset Type** portal by selecting **Admin**, select **Asset Management**, then select **Asset Type**. You use the **Asset Type Query** portal and **Asset Search** zone to search for the controllable asset type you wish to view. To search for controllable asset types, select "Controllable Asset" from the **Asset Business Object** drop-down list. When you select the controllable asset type you wish to view, it opens in the **Controllable Asset Type** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Type**: This zone provides information about the selected controllable asset type.

The following zones may appear as part of the portal's **Device Type** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Device Type**: This zone provides information about the device type associated to the selected controllable asset type.
- **Device Type Characteristics**: This zone lists characteristics and values for the device type associated to the selected controllable asset type.
- **Valid Device Characteristic Types**: This zone lists valid characteristic types for the device type associated to the selected controllable asset type.

Defining Device Location Types

You use the **Device Location Type** portal to display and maintain device location types.

Refer to Understanding Device Location Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Device Location Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Location Type List**: This zone lists all device location type records. Broadcast a record to display the details of the selected record.
- **Device Location Type**: This zone provides information about the selected device location type.

Defining Head End Systems

You use the **Head End System** portal to display and maintain Head End Systems.

Refer to Understanding Head End Systems for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Head End System**.

The following zones may appear as part of the portal's **Main** tab page:

- **Head End System List**: This zone lists all head end system records. Broadcast a record to display the details of the selected record.
- **Head End System**: This zone provides information about the selected head end system.
- **Processing Method List**: This zone provides the list of processing methods defined for the head end system.
- **Translation Methods List**: This zone provides the list of translation methods defined for the head end system.
- **Inbound BOs Send By Service Provider**: This zone lists inbound business objects that are sent by this head end system. The identification is driven by the Business Object having a Business Object Option of type "Sent By Service Provider" that references the current Head End System.

Defining DER Aggregators

You use the **DER Aggregator** portal to display and maintain Distributed Energy Resource (DER) Aggregators.

Refer to Understanding DER Aggregators for more information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **DER Aggregator**.

The following zones may appear as part of the portal's **Main** tab page:

- **DER Aggregator List**: This zone lists all DER aggregator records. Broadcast a record to display the details of the selected record.
- **DER Aggregator**: This zone provides information about the selected aggregator.
- **Processing Method List**: This zone provides the list of processing methods defined for the aggregator.

Defining Metered Service Types

You use the **Metered Service Type** portal to display and maintain metered service types.

Refer to Understanding Metered Service Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Metered Service Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Metered Service Type List**: This zone lists all metered service type records. Broadcast a record to display the details of the selected record.
- **Metered Service Type**: This zone displays the selected metered service type.
- **Attributes**: This zone displays a list of characteristic types that may be defined for metered service records of the selected metered service type.

Defining Metered Service Point Types

You use the **Metered Service Point Type** portal to view metered service point types.

Refer to Understanding Metered Service Point Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Metered Service Point Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Metered Service Point Type List**: This zone lists all metered service point type records. Broadcast a record to display the details of the selected record.
- **Metered Service Point Type**: This zone provides information about the selected metered service point type.

Defining Program Control Score Calculation Groups

You use the **Program Control Score Calculation Group** portal to display and maintain program control score calculation groups.

Refer to Understanding Program Control Score Calculation Groups for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Control Score Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Control Score Calculation Group**: This zone displays basic information about program control score calculation group.
- **Program Control Score Calculation Rules List**: This zone lists the program control score calculation rules belonging to the group.

Defining Program Control Score Calculation Rules

You use the **Program Control Score Calculation Rule** portal to display and maintain program calculation rules.

Refer to Understanding Program Control Score Calculation Rules and Understanding Program Control Score Calculation Groups for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Control Score Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Control Score Calculation Rule**: This zone displays the program control score calculation rule, including parameters used when executing the rule.

Program Control Score Calculation Calculation Rules

The following is a list of the program control score calculation program calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Program Calculation Rule Name	Purpose
Score Calculation Rule	This rule is used to calculate control scores for controllable devices based on parameters and values defined in a the device's program rules. This rule uses pre-defined variables and base package algorithms to derive values from the program's rules. Multiple instances of this rule can be chained in a sequence within a program calculation group such that the first rule (or rules) sets values for a number of variables that are then used in calculations in subsequent rules.

Variables

The Program Variables (X1-ProgramVariablesLookup) extendable lookup includes a number of pre-defined and standard variables that can be used in program calculation rules. The table below lists the variables provided with the base package.

Variable	Description
AC	Activation Cost (AC)
AV	Availability (AV)
BR	Burn Rate (BR)
BRn	Burn Rate Factor (BRn)
CA	Successful Event Participation Limit (CA)

Copyright © 2026, Oracle and/or its affiliates.

Variable	Description
CU	Successful Event Participation Count (CU)
DE	Days Elapsed Since Last Event Participation (DE)
DP	Days in a Cycle (DP)
SC	Score (SC)
SNn	Score Factor (SNn)

The following variables are used to populate specific values on controllable assets:

- **Score (SC)**: When Program Calculation Rules return a value for this variable (defined as the **Result Variable** on the rule), the score is saved to the controllable asset (along with the date and time at which the score was calculated).
- **Availability (AV)**: If Program Calculation Rules return a value of 0 for this variable, the **Asset Availability** flag on the controllable asset is set to "No". If Program Calculation Rules return a non-zero value for this variable or the rule doesn't return a value for this variable, the **Asset Availability** flag on the controllable asset is set to "Yes".

Algorithms

Program calculation rules can leverage base package algorithms to derive values from programs and program rules for use in calculations. Each of these is typically paired with a pre-defined variable (see Variables, above). The table below lists the base package algorithms and associated variables.

Algorithm	Associated Variable
Activation Cost (X1-GETAC)	AC
Burn Rate Factor (X1-GETBRF)	BRn
Days in a Cycle (X1-GETDCP)	DP
Days Elapsed Since Last Event Participation (X1-GETDE)	DE
Successful Event Participation Limit (X1-GETEPA)	CA
Successful Event Participation Count (X1-GETEPC)	CU
Score Factor (X1-GETSF)	SNn

- Score Calculation Rule

Score Calculation Rule

This rule is used to calculate control scores for controllable devices based on parameters and values defined in a the device's program rules. This rule uses pre-defined variables and base package algorithms to derive values from the program's rules. Multiple instances of this rule can be chained in a sequence within a program calculation group such that the first rule (or rules) sets values for a number of variables that are then used in calculations in subsequent rules.

Calculation Details

The **Calculation Details** section is used to define the formulas used in the rule's calculations. Simple formulas support basic math operations using variables defined in the **Variables** section. Complex formulas allow for more complex operations, comparisons, and conditions

Copyright © 2026, Oracle and/or its affiliates.

upon which certain variables are set to certain values. Refer to the embedded help in this section for details about configuring formulas.

Variables

The **Variables** section is used to define one or more variables that will be used in formulas defined in the **Calculation Details** section. The variable's **Variable Type** determines how the variable is populated:

- **Algorithm**: The value is returned by a specified algorithm.
- **Transaction Value**: The value is based on a variable defined in a previous rule and stored in the program transaction's **Calculation Results**.

Variables must be defined in the Program Variables (X1-ProgramVariablesLookup) extendable lookup. In addition to the pre-defined variables provided, implementations can created their own variables using this extendable lookup.

When deriving values from the program rule's **Score Calculation Factors** section, values are defined per **Cycle**, which is the number of times the device has successfully participated in an event during the current cycle (week or month), plus 1. A device's event participation is considered successful in the following circumstances:

- The Event Participation has an **Outcome** of "Successful Participation".
- The Event Participation has an **Outcome** of "Canceled". This happens when a device has been reserved but the event stopped halfway through due to reasons such as, target load control reached. It is considered as a success in customer perspective, since it is the utility that stopped the event which resulted to a cancelled outcome.
- The Event Participation has a blank **Outcome** and the device was not preemptively opted out of the event (**Preemptive Opt-Out** is "No" or blank). Posting of an event's outcome may be delayed, and until then the event participation of a reserved device is considered as a success as long as the customer did not opt out before the event transpires.

For instance, if a device has never participated in an event, the algorithms would use the values in the Cycle 1 row. If a device has successfully participated in 3 previous events, the algorithms would use the values in the Cycle 4 row.

Results

The **Results** section is used to define result of the program calculation rule.

To save the control score to the controllable asset, set the Result Variable on the final rule in the program calculation group to "SC" (the Score variable).

To mark a device as unavailable under certain conditions (such as if specific values are missing from the Program Rule), configure the rule to return a value of "0" for the AV (Availability) variable and set the **Result Variable** to "AV" (the Availability variable).

Notes and Recommendations

Additional detail on the logic of this rule can be found in the Detailed Description of the X1-SCRCALC Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

X1-ScoreCalculationRule

Example Scenarios

Copyright © 2026, Oracle and/or its affiliates.

Below are some example scenarios that can be achieved based on configuration of this rule.

In this example, the first rule calculates a burn rate (BR) variable that is used in calculating the control score (SC) in the second rule.

Rule 1: Calculate Burn Rate

- **Sequence**: 10
- **Description**: Calculate Burn Rate
- **Calculation Details**:
  - **Formula Source**: Conditional Formula
  - **Conditional Formula**:

Seq.	Operand 1	Criteria Operator	Operand 2	True Action	True Formula	False Action	False Formula
10	CU	=	CA	Apply True Formula	99999	Check Next Condition	
20	DE	<=	DBC	Apply True Formula	99999	Apply False Formula	((CU/CA) / (DE/DP)) * BRF

- **Variables**:

Variable Name	Variable Type	Algorithm
BRF	Algorithm	Burn Rate Factor
CA	Algorithm	Event Participation Allowed
CU	Algorithm	Event Participation Count
DE	Algorithm	Days Elapsed Since Last Event Participation
DP	Algorithm	Days in a Cycle

- **Results**:
  - **Variable Name**: BR

Rule 2: Calculate Score

- **Sequence**: 20
- **Description**: Calculate Score
- **Calculation Details**:
  - **Formula Source**: Simple Formula
  - **Simple Formula**: BR * AC * SCF
- **Variables**:

Variable Name	Variable Type	Algorithm
AC	Algorithm	Activation Cost
BR	Transaction Value	
SCF	Algorithm	Score Factor

- **Results**:

Copyright © 2026, Oracle and/or its affiliates.

- **Variable Name**: SC

Defining Program Initiatives

You use the **Program Initiative** portal to display and maintain program initiatives.

Refer to Understanding Program Initiatives for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Initiative**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Initiative**: This zone displays the selected program initiative.
- **Initiative Eligibility Criteria Overview**: This zone lists criteria used to determine a customer's eligibility in a program. Click **Add Eligibility Criteria** to add a criteria. Click the **Edit** or **Delete** icons to edit or delete (respectively) an existing criteria.

Defining Program Rules

You use the **Program Rule** portal to display and maintain program rules.

Refer to Understanding Program Rules for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Rule**.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Rule List**: This zone lists all program rule records. Broadcast a record to display the details of the selected record.
- **Program Rule**: This zone displays the selected program rule.

Defining Program Contact Types

You use the **Contact Type** portal to display and maintain program contact types.

Refer to Understanding Program Contact Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Contact Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact Type List**: This zone lists all program contact type records. Broadcast a record to display the details of the selected record.
- **Contact Type**: This zone displays the selected program contact type.

Defining Program Subscription Types

You use the **Program Subscription Type** portal to display and maintain program subscription types.

Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding Program Subscription Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Subscription Type List**: This zone lists all program subscription type records. Broadcast a record to display the details of the selected record.
- **Program Subscription Type**: This zone displays the selected program subscription type.
- **Attributes**: This zone lists haracteristics that can be defined for program subscriptions of this type.

Defining Program Types

You use the **Program Type** portal to display and maintain a program types.

Refer to Understanding Program Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Type List**: This zone lists all program type records. Broadcast a record to display the details of the selected record.
- **Program Type**: This zone displays the selected program type.

Defining Program Settlement Calculation Groups

You use the **Program Settlement Calculation Group** portal to display and maintain program settlement calculation groups.

Refer to Understanding Program Settlement Calculation Groups for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Settlement Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Settlement Calculation Group**: This zone displays basic information about program settlement calculation group.
- **Program Settlement Calculation Rules List**: This zone lists the program settlement calculation rules belonging to the group.

Defining Program Settlement Calculation Rules

You use the **Program Settlement Calculation Rule** portal to display and maintain program settlement calculation rules.

Refer to Understanding Program Settlement Calculation Rules and Understanding Program Settlement Calculation Groups for more information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Program Settlement Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Program Settlement Calculation Rule**: This zone displays the program settlement calculation rule, including parameters used when executing the rule.
- **Eligibility Criteria List**: This zone lists eligibility criteria for the current calculation rule. Eligibility criteria are configured criteria used to determine whether to execute a specific calculation rule.

Program Settlement Calculation Rules

The following is a list of the program settlement calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Program Settlement Calculation Rule Name	Purpose
End of Event Program Settlement Rule	This rule is used to calculate program event settlement transactions that represent credits that will be applied to a customer's bill based on participation in an individual event. This rule uses multiplies "kWh avoided" or "kW drop" interval values captured on program event settlement transactions by interval prices defined in an interval price set. Interval prices can be defined as specific interval price sets or can be derived from more than one interval price set by an algorithm. When program event settlement transactions are created from data received from an external analytics application, "kWh avoided" or "kW drop" interval values are populated when the transaction is created. When program event settlement transactions are created via batch processing "kWh avoided" or "kW drop" interval values are calculated via baseline calculations, performed via a combination of a Baseline Calculation rule and Vector and Service Quantity Math rule to calculate reductions in consumption or demand.

> **Note**
>
> The base package does not include algorithms for deriving prices for use in program settlement calculations. Implementations must configure their own algorithms if they wish to derive prices from more than one interval price set.

Copyright © 2026, Oracle and/or its affiliates.

Program Settlement Calculation Rule Name	Purpose
End of Season Program Settlement Rule	This rule is used to calculate "Combined kW Drop" program event settlement transactions that represent credits that will be applied to a customer's bill based on participation in an individual event. This rule averages values captured on "kW Drop" program event settlement transactions and multiplies the result by a price. Prices can be defined in the rule configuration or can be derived by an algorithm. The average "kW Drop" value (captured as the **Settlement Quantity** on the "Combined kW Drop" program event settlement transaction) is calculated from the "Max Drop" values from the "kW Drop" program event settlement transactions that fall into the "top" percentage of values defined by the **Demand Drop Percentage (%)** parameter. See About Program Event Settlement Transactions for more information about the distinction between "kW Drop" and "Combined kW Drop" program event settlement transactions. When program event settlement transactions are created from data received from an external analytics application, "kWh avoided" or "kW drop" interval values are populated when the transaction is created. When program event settlement transactions are created via batch processing "kWh avoided" or "kW drop" interval values are calculated via baseline calculations, performed via a combination of a Baseline Calculation rule and Vector and Service Quantity Math rule to calculate reductions in consumption or demand. **Note** The base package does not include algorithms for deriving prices for use in program settlement calculations. Implementations must configure their own algorithms if they wish to derive prices from more than one interval price set.
Incentive and Fee Calculation Rule	This rule is used to calculate incentive- and fee-based program event settlement transactions that represent credits or changes that will be applied to a customer's bill based on incentives and/or fees configured for a program to which the customer subscribes.

Copyright © 2026, Oracle and/or its affiliates.

Program Settlement Calculation Rule Name	Purpose
Baseline Calculation	This rule is used to calculate estimated baseline consumption for a metered service point that represents an estimate of the customer's consumption or demand for an event period had they not participated in the event.
Vector and Service Quantity Math	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations. This can be used to calculate reductions in consumption or demand by subtracting actual consumption or demand from baseline consumption or demand calculated using the Baseline Calculation rule.
Execute Usage Calculation Group	This rule is used to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. For example, this can be used to group a Baseline Calculation rule and Vector and Service Quantity Math rule to calculate reductions in consumption or demand, which can then be passed to an End of Event Program Settlement Rule or End of Season Program Settlement rule.

- End of Event Program Settlement Rule
- End of Season Program Settlement Rule
- Incentives and Fee Rule
- Baseline Calculation Rule
- Vector and Service Quantity Math
- Execute Usage Calculation Group

End of Event Program Settlement Rule

This rule is used to calculate program event settlement transactions that represent credits that will be applied to a customer's bill based on participation in an individual event. This rule multiplies "kWh avoided" or "kW drop" interval values captured in program event settlement transactions by interval prices defined in an interval price set. Interval prices can be defined as specific interval price sets or can be derived by an algorithm.

> **Note**
>
> The base package does not include algorithms for deriving prices for use in program settlement calculations. Implementations must configure their own algorithms if they wish to derive prices from more than one interval price set.

Calculation Details

The **Calculation Details** section is used to define the interval values and interval prices used in the rule's calculations.

Copyright © 2026, Oracle and/or its affiliates.

- **UOM**: The unit of measure used to define the interval values to be created in the program settlement transaction.
- **TOU**: The time of use period used to define the interval values to be created in the program settlement transaction.
- **SQI**: The service quantity identifier used to define the interval values to be created in the program settlement transaction.
- **Set Function**: The function used to calculate consumption for the program event settlement transaction. Applicable only if the select UOM is one that measures peak. Options include:
  - **Average**: Use the average value
  - **Maximum**: Use the maximum value
  - **Minimum**: Use the minimum
- **Interval Size**: The size of the interval values to be created in the program settlement transaction represented as hours:minutes:seconds (HH:MI:SS). This should be the same interval size are defined on interval price set types that will be used with the rule. See **Understanding Interval Price Set Types** for more information.
- **Line Description**: A text string that describes the program settlement transaction
- **Defer Amount Calculation**: A flag that indicates (Yes or No) whether or not calculation of the program event settlement transaction should be deferred.
- **Total Consumption UOM**: The unit of measure for the total consumption calculated by the program event settlement transaction.
- **Total Consumption TOU**: The time of user period for the total consumption calculated by the program event settlement transaction.
- **Total Consumption SQI**: The service quantity identifier for the total consumption calculated by the program event settlement transaction.
- **Price Source**: The source of interval price data used by the rule. Options include:
  - **Algorithm**: A custom algorithm that derives interval prices from a set of interval price sets
  - **Specific Price**: A specific interval price set to be used by the rule

SQ Rounding Details

The **SQ Rounding Details** section defines the manner in which service quantities are rounded. These parameters allow configuration of the **Rounding Method** by which quantities are rounded (Up, Down, or Nearest) and the number of **Decimal Positions** retained after rounding.

Notes and Recommendations

Additional detail on the logic of this rule can be found in the **Detailed Description** of the X1-EVTSETCAL Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

X1-EventSettlementCalcRule

Example Scenarios

Copyright © 2026, Oracle and/or its affiliates.

Scenario 1: Calculate program settlement for hourly interval values based on hourly market prices

- Sequence: 10
- **Description**: Thermostat Program Settlement (Hourly)
- **Calculation Details**:
  - **UOM**: Kilowatt Hours
  - **TOU**:
  - **SQI**:
  - **Interval Size**: 01:00:00
  - **Line Description**: Thermostat Program Settlement - Hourly
  - **Price Source**: Specific Price
    - **Specific Price**: Market Price - Hourly

End of Season Program Settlement Rule

This rule is used to calculate "Combined kW Drop" program event settlement transactions that represent credits that will be applied to a customer's bill based on participation in an individual event. This rule averages values captured on "kW Drop" program event settlement transactions and multiplies the result by a price. Prices can be defined in the rule configuration or can be derived by an algorithm. The average "kW Drop" value (captured as the **Settlement Quantity** on the "Combined kW Drop" program event settlement transaction) is calculated from the "Max Drop" values from the "kW Drop" program event settlement transactions that fall into the "top" percentage of values defined by the **Demand Drop Percentage (%)** parameter. See About Program Event Settlement Transactions for more information about the distinction between "kW Drop" and "Combined kW Drop" program event settlement transactions.

Note: The base package does not include algorithms for deriving prices for use in program settlement calculations. Implementations must configure their own algorithms if they wish to derive prices.

Calculation Details

The **Calculation Details** section is used to define how the average "kW Drop" value is determined and the price used in the rule's calculations.

- **UOM**: The unit of measure used to define the average value to be created in the program event settlement transaction.
- **TOU**: The time of use period used to define the average value to be created in the program event settlement transaction.
- **SQI**: The service quantity identifier used to define the average value to be created in the program event settlement transaction.
- **Interval Size**: The size of the average value to be created in the program event settlement transaction represented as hours:minutes:seconds (HH:MI:SS).
- **Line Description**: A text string that describes the program event settlement transaction. This can include the following variables:
  - %SQ = Quantity
  - %UP = Unit Price
- **Price Source**: The source of price data used by the rule. Options include:

Copyright © 2026, Oracle and/or its affiliates.

  - **Algorithm**: A custom algorithm that derives the price.
  - **Unit Price**: A specific price to be used by the rule
- **Price**: The price that will be multiplied by **Settlement Quantity** on the "Combined kW Drop" program event settlement transaction to determine the "Combined kW Drop" program event settlement transaction's **Settlement Amount**.
- **Demand Drop Percentage**: The top percentage of "kW Drop" program event settlement transactions that will be averaged. The rule uses this value to select the top n% of "Maximum Drop" values from the "kW Drop" program event settlement transactions. For example, if there are 10 "kW Drop" program event settlement transactions and this parameter is set to "70%", the rule would calculate the average for the top 7 values.

SQ Rounding Details

The **SQ Rounding Details** section defines the manner in which service quantities are rounded. These parameters allow configuration of the **Rounding Method** by which quantities are rounded (Up, Down, or Nearest) and the number of **Decimal Positions** retained after rounding.

Notes and Recommendations

Additional detail on the logic of this rule can be found in the **Detailed Description** of the X1_CALAVDPAP Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

X1-EventSetCalcRuleDemandBased

Example Scenarios

Scenario 1: Calculate program settlement for demand reductions

- **Sequence**: 10
- **Description**: Demand Program Settlement
- **Calculation Details**:
  - **UOM**: Kilowatts
  - **TOU**:
  - **SQI**:
  - **Interval Size**: 01:00:00
  - **Line Description**: Event Participation Settlement: %SQ - %UP
  - **Price Source**: Unit Price
  - **Unit Price**: $12.00
  - **Demand Drop Percentage (%)**: 70

Incentives and Fee Rule

This rule is used to calculate incentive- and fee-based program event settlement transactions that represent credits or changes that will be applied to a customer's bill based on incentives and/or fees configured for a program to which the customer subscribes.

Variables

Copyright © 2026, Oracle and/or its affiliates.

The **Variables** section is used to define one or more variables that will be used in formulas defined in the **Calculation Details** section. The variable's **Variable Type** determines how the variable is populated:

- **Algorithm**: The value is returned by a specified algorithm.
- **Transaction Value**: The value is based on a variable defined in a previous rule and stored in the program transaction's **Calculation Results**.

Variables must be defined in the Incentive and Fees Variables (X1-IncentiveFeeVariablesLookup) extendable lookup. In addition to the pre-defined variables provided, implementations can created their own variables using this extendable lookup.

Note: The base package does not include algorithms for deriving prices for use in incentives and fees calculations. Implementations must configure their own algorithms if they wish to derive prices and/or values from other data.

Calculation Details

The **Calculation Details** section is used to define the formulas used in the rule's calculations. Simple formulas support basic math operations using variables defined in the **Variables** section. Complex formulas allow for more complex operations, comparisons, and conditions upon which certain variables are set to certain values. Refer to the embedded help in this section for details about configuring formulas.

Results

The Results section is used to define result parameters of the incentives and fees calculation rule, including:

- **Variable Name**: The variable that holds the result of the calculation.
- **FCPO**: Checkbox that indicates if the result will be used for calculation purpose only. This is most often used with results that are prices or quantities and that will be Transaction Values passed to other rules. This checkbox should be unchecked for rules that calculates the final incentive or fee amount.
- **Line Description**:A description of the calculated value. It is relevant only to the rule that calculates the incentive or fee amount.
- **Result Container**: indicates if the result is a quantity, price or an amount.

Notes and Recommendations

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

X1-IncentiveCalculationRule

Example Scenario

Below is an example scenario that can be achieved based on configurations of this rule.

In this example, the first rule calculates a Service Charge variable that is used in calculating a Penalty Charge in the second rule.

Rule 1: Calculate Service Charge

- **Sequence**: 10
- **Description**: Calculate Service Charge
- **Variables**:

Copyright © 2026, Oracle and/or its affiliates.

Variable Name	Variable Type	Algorithm
Early Termination Fee (ET)	Derive Price Algorithm	Derive Termination Fee (custom)
Penalty Fee (PF)	Derive Quantity Algorithm	Derive Penalty (Custom)

- **Calculation Details**:
  - **Formula Source**: Conditional Formula
  - **Conditional Formula**:

Seq.	Operand 1	Criteria Operator	Operand 2	True Action	True Formula	False Action	False Formula
10	PF	>	ET	Apply True Formula	PF+ET	Apply False Formula	PF-ET

- **Results**:
  - **Variable Name**: Service Charge (SC)
  - **FCPO**: Yes (checked)
  - **Result Container**: Price

Rule 2: Calculate Penalty

- **Sequence**: 20
- **Description**: Calculate Penalty Charge
- **Variables**:

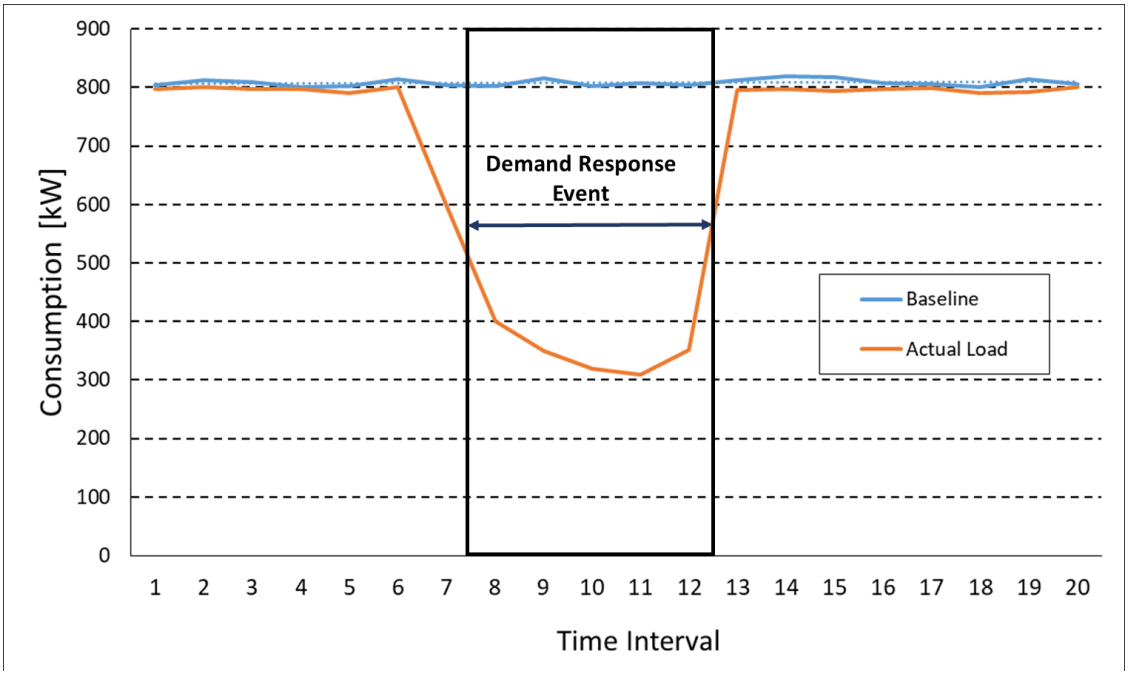
Variable Name	Variable Type	Algorithm
Service Charge (SC)	Transaction Value	

- **Calculation Details**:
  - **Formula Source**: Simple Formula
  - **Simple Formula**: SC
- **Results**:
  - **Variable Name**: Penalty (PN)
  - **FCPO**: No (unchecked)
  - **Line Description**: Penalty
  - **Result Container**: Amount

Baseline Calculation Rule

This rule is used to estimate the electric consumption a customer would have used had they not participated in a program event, which is used to calculate reductions in consumption or demand for use in program event settlement transactions that represent credits or changes that will be applied to a customer's bill based on participation in an individual event. This done by calculating a "Demand Response Customer Baseline" or baseline that represents this estimated consumption. The figure below shows a baseline compared to actual meter data and depicts how they can be compared during an event to calculate the reduction.

Copyright © 2026, Oracle and/or its affiliates.



The Baseline Calculation rule uses an "X of Y Calculation Method", a statistical methodology that considers Y eligible days (referred to as "Qualified" or "Qualifying" days) and selects the X days (referred to as "Selected" days) with the load to calculate the baseline on the basis of the most recent meter readings for consumption levels within those days. Qualified and Selected days can be determined based on Load, Temperature, or Net values (Consumed - Generated). The **X Selection Parameter** defines if the rule will use Load or Temperature, and the presence of "Generated" UOM, TOU, and SQI values indicate that the rule should use Net values.

> **Note**
>
> Using "Temperature" as the **X Selection Parameter** requires weather stations be defined for metered service points.

The **Baseline Calculation Details** section is used to define how the baseline consumption is calculated.

- **UOM**: The unit of measure used to define the interval values to be created in the program event settlement transaction.
- **TOU**: The time of use period used to define the interval values to be created in the program event settlement transaction.
- **SQI**: The service quantity identifier used to define the interval values to be created in the program event settlement transaction.
- **UOM Generated**: The unit of measure used to define generated interval values used in the program event settlement transaction calculation. Used to calculate Net consumption (Consumed - Generated).
- **TOU Generated**: The time of use period used to define generated interval values used in the program event settlement transaction calculation. Used to calculate Net consumption (Consumed - Generated).

Copyright © 2026, Oracle and/or its affiliates.

- **SQI Generated**: The service quantity identifier used to define generated interval values used in the program event settlement transaction calculation. Used to calculate Net consumption (Consumed - Generated).
- **Work Calendar**: A work calendar used to define weekdays, weekends and holidays. See Defining Work Calendar for more information.
- **Lookback**: The number of days preceding an Event Day (the day on which a Program Event took place) to be evaluated when identifying Qualified days. Examples could include the last 10 non-holiday weekdays, the 10 most recent program-eligible non-event days, the10 most recent program-eligible days beginning 2 days before the event, last 45 calendar days, or the previous year.
- **X of Y Type**: The type of X days to be selected. Valid options include:
  - **High**: Excludes the lowest Y-X values
  - **Low**: Excludes the highest Y-X values
  - **Middle**: Uses the middle values of the Y quantities
- **X Selection Parameter**: The criteria for the selection of (X) Selected days from the (Y) Qualifying Days. Valid options include:
  - **Load**: The baseline is calculated by taking the highest/lowest/middle consumption from the X highest/lowest/middle consumption days from the Y Qualified days
  - **Temperature**: The baseline is calculated by taking the highest/lowest/middle consumption from the X highest/lowest/middle temperature days from the Y Qualified temperature days.
- **UOM Temperature**: The unit of measure used to define the temperature values used in the calculation.
- **TOU Temperature**: The time of use period used define the temperature values used in the calculation.
- **SQI Temperature**: The service quantity identifier used to define the temperature values used in the calculation.
- **Value of X**: The number of Selected Days
- **Value of Y**: The number of Qualified Days
- **Adjustment Type**: The type of adjustment applied to the baseline, used if or when conditions on the day of a Program Event may differ from the conditions over the previous days which were used for the calculation of baseline. Valid options include:
  - **Additive**: Adds or subtracts a numerical value to each hour of the event. This value is calculated by:
    * Averaging the 4 hours prior to the event excluding the hour immediately prior to the event start; and,
    * Define the difference between the average load for these three hours relative to the same 3-hour average of the CBL calculation data set.
    * The difference is then applied to each counterfactual hour of the event and is capped at both a 20% increase and a 20% decrease.

> **Note**
>
> This is modeled after the "PJM CBL Addition Adjustment" approach.

Copyright © 2026, Oracle and/or its affiliates.

- **Multiplicative**: Applies a multiple to each hour of the event. The multiplier is calculated by:
  - Averaging the 4 hours prior to the event excluding the hour immediately prior to the event start; and,
  - Define the ratio of the average load for these three hours relative to the same 3-hour average of the CBL calculation data set.
  - The same multiplier is then applied to each counterfactual hour of the event and is capped at both a 20% increase and a 20% decrease.

> **Note**
>
> This is modeled after the "CAISO CBL Multiplication Adjustment" approach.

- **None**: the standard X of Y method selected will be employed, utilizing the X and Y values configured to compute the baseline

- **Adjustment Window Start**: The number of (15-minute) intervals which are selected to calculate the fixed quantity for adjustment of the baseline. Typically, the adjustment window start begins no more than four hours prior to an event's start. Applicable only if the Adjustment Type is "Additive" or "Multiplicative".
- **Adjustment Window End**: The number of (15-minute) intervals which are selected to calculate the fixed quantity for adjustment of the baseline. Typically, the adjustment window end ends no more than an hour prior to the event start .Applicable only if the Adjustment Type is "Additive" or "Multiplicative".
- **Adjustment Cap**: A limit on how much the baseline can be adjusted upwards or downwards during an event, typically expressed as a percentage of the original baseline, used to prevent extreme adjustments and ensure fairness in the measurement of demand reduction. For example, if a customer's normal baseline load is 100kW and the adjustment cap is set at 20%, then the maximum adjusted baseline could only be 120kW (100kW + 20% increase) or 80kW (100kW - 20% decrease), regardless of how much their load deviates during the event. Applicable only if the Adjustment Type is "Additive" or "Multiplicative".
- **Interval Size**: The size of the interval values to be created in the program event settlement transaction represented as hours:minutes:seconds (HH:MI:SS).

Notes and Recommendations

Additional detail on the logic of this rule can be found in the **Detailed Description** of the D2-BASELINEC Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-BaselineCalc

Example Scenarios

Scenario 1: Calculate a baseline based on the Highest 3 (X) of 5 (Y) days within 30 days prior to the program event, with no adjustment type

- **Sequence**: 10
- **Description**: Baseline - Highest 3 of 5

Copyright © 2026, Oracle and/or its affiliates.

- **Baseline Calculation Details**:
  - **UOM**: Kilowatt Hours
  - **TOU**:
  - **SQI**:
  - **Work Calendar**: US 2025 Work Calendar
  - **Lookback**: 30
  - **X of Y Type**: Highest
  - **X Selection Parameter**: Load
  - **Value of X**: 3
  - **Value of Y**: 5
  - **Adjustment Type**: None
  - **Interval Size**: 01:00:00
- **Results**:
  - **Qualified Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Qualified
  - **Selected Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Selected
  - **Skipped Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Skipped
  - **Pre Event Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Pre-Event
  - **Pre Event Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Pre-Event Selected
  - **Raw Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Raw

Scenario 2: Calculate a baseline based on the Highest 3 (X) of 5 (Y) days within 30 days prior to the program event using net consumption, with no adjustment type

- **Sequence**: 10
- **Description**: Baseline - Highest 3 of 5
- **Baseline Calculation Details**:
  - **UOM**: Kilowatt Hours - Consumed
  - **TOU**:

Copyright © 2026, Oracle and/or its affiliates.

- **SQI**:
- **UOM Generated**: Kilowatt Hours - Generated
- **TOU Generated**:
- **SQI Generated**:
- **Work Calendar**: US 2025 Work Calendar
- **Lookback**: 30
- **X of Y Type**: Highest
- **X Selection Parameter**: Load
- **Value of X**: 3
- **Value of Y**: 5
- **Adjustment Type**: None
- **Interval Size**: 01:00:00

- **Results**:
  - **Qualified Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Qualified
  - **Selected Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Selected
  - **Skipped Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Skipped
  - **Pre Event Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Pre-Event
  - **Pre Event Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Pre-Event Selected
  - **Raw Baseline Measurements**
    - **Unit of Measure**: Kilowatt Hours
    - **Service Quantity Identifier**: Baseline Raw

Vector and Service Quantity Math

This rule is designed to facilitate configuration of complex vector and scalar quantity calculations. It is based on a series of underlying services, including Axis Conversion, Apply Formula (most importantly), and Apply TOU Map. It can be configured to accept as input up to five vectors - where a single vector can represent the combination of all measuring components with linkages to the Usage Subscription with the configured UOM/SQI. It can also

Copyright © 2026, Oracle and/or its affiliates.

accept a list of scalar variables that can be taken in as factor values, usage transaction service quantities, set functions on a vector, or numeric values.

Vectors can be combined using a simple formula expression, or using condition formula expressions. Both interval values and interval conditions can be referenced. The result is interval-by-interval processing of the vector formula. The resulting final vector can be stored, TOU mapped, and/or subjected to a final set function (such as sum or max).

This rule stores service quantities of type "other" - meaning that the service quantities created by this rule do not have linkages to a specific Service Point or measuring component, given the potentially diverse sources of the data taken as input by this rule that lead to the final quantity.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-MATH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-Math

Vector and Service Quantity Math Usage Calculation Rules and Anchor Time

The Vector and Service Quantity Math usage calculation rule supports usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

Execute Usage Calculation Group

This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. It is especially useful in cases where repeating the same rule over and over would be hard to maintain.

For example, you may want to calculate a straightforward kWh sum for every usage calculation group. Creating a separate CALC_KWH_SUM usage calculation group for this one calculation allows you to isolate the configuration points. Then the CALC_KWH_SUM usage calculation group can be referenced in every other usage calculation group that needs this sort of service quantity.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AUSGR-RFG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Copyright © 2026, Oracle and/or its affiliates.

Business Object

D1-UsageRuleReferredUsageGroup

Configuring Usage Transaction Exception Types

You ue the **Usage Transaction Exception Type** portal to display and maintain usage transaction exception types .

Refer to Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Transaction Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Transaction Exception Type List**: displays all of the Usage Transaction Exception Types so you can choose the one you want to display in more detail
- **Usage Transaction Exception Type**: shows the specific configuration for the selected Usage Transaction Exception Type

Defining Interval Price Set Types

You use the **Interval Price Set Type** portal to display and maintain interval price set types.

Refer to **Understanding Interval Price Set Types** for more information./

You can access the portal by selecting **Admin**, select **Interval Price Set**, then select **Interval Price Set Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set Type**: This zone displays basic information about interval price set type.

Defining Customer Questions

You use the **Question** portal to display and maintain customer questions.

Refer to Understanding Customer Questions for more information.

You can access the portal by selecting **Admin**, **General**, **Customer Question**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Question**: This zone displays the selected question.

Defining Service Areas

You use the **Service Area** portal to display and maintain service areas.

Refer to Understanding Service Areas for more information.

Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, **Customer Program Management**, **Service Area**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Area List**: This zone lists all service areas defined in the system. Broadcast a record to display the details of the selected record.
- **Service Area**: This zone displays the selected service area.

Defining Service Classes

You use the **Service Class** portal to display and maintain service areas.

Refer to Understanding Service Classes for more information.

You can access the portal by selecting **Admin**, **Customer Program Management**, **Service Class**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Class List**: This zone lists all service classes defined in the system. Broadcast a record to display the details of the selected record.
- **Service Class**: This zone displays the selected service class.

Defining Service Codes

You use the **Service Code - CCB** portal to display and maintain service areas.

Refer to Understanding Service Codes for more information.

To access the portal select **Admin**, select **Field Work**, then select **Service Code - CCB**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Code List**: This zone lists all service codes defined in the system. Broadcast a record to display the details of the selected record.
- **Service Code**: This zone displays the selected service code.

Defining Service Types

You use the **Service Type** portal to display and maintain service types.

Refer to Understanding Service Types for more information.

You can access the portal by selecting **Admin**, **Customer Program Management**, **Service Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Type List**: This zone lists all service types defined in the system. Broadcast a record to display the details of the selected record.
- **Service Type**: This zone displays the selected service type.

Defining Units of Measure

You use the **Unit of Measure** portal is used to display and maintain units of measure.

Copyright © 2026, Oracle and/or its affiliates.

Refer to **Understanding Units of Measure** for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Unit Of Measure**.

The following zones may appear as part of the portal's **Main** tab page:

- **Unit of Measure List:** This zone lists all units of measure. Broadcast a record to display the details of the selected record.
- **Unit of Measure:** This zone provides information about the selected unit of measure.

Defining Time of Use Periods

You use the **Time of Use** portal is used to display and maintain time of Use periods.

Refer to **Understanding Time of Use Periods** for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Time Of Use**.

The following zones may appear as part of the portal's **Main** tab page:

- **Time Of Use List:** This zone lists all time of use periods records. Broadcast a record to display the details of the selected record.
- **Time Of Use:** This zone provides information about the selected time of use period.

Defining Service Quantity Identifiers

You use the **Service Quantity Identifier** portal is used to display and maintain service quantity identifiers.

Refer to **Understanding Service Quantity Identifiers** for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Service Quantity Identifier**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Quantity Identifier List:** This zone lists all service quantity identifiers. Broadcast a record to display the details of the selected record.
- **Service Quantity Identifier:** This zone provides information about the selected service quantity identifier.

Defining Divisions

You use the **Division** portal to display and maintain divisions.

Refer to Understanding Divisions for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **Division**.

The following zones may appear as part of the portal's **Main** tab page:

- **Division List**: This zone lists all divisions defined in the system. Broadcast a record to display the details of the selected record.

Copyright © 2026, Oracle and/or its affiliates.

- **Division**: This zone displays the selected division.

Entity Groups

You use the **Entity Group** portal to display and maintain entity groups.

Refer to Understanding Entity Groups for more information.

You can access the portal by selecting **Admin**, select **General**, then select **Entity Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Entity Group**: This zone displays basic information about entity group.
- **Eligible Devices for this Group**: This zone shows a list of devices that are eligible for assignment to this entity group. The list is based on the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more devices to the group, select the devices and click **Assign**. Applicable only for "Device" entity groups.
- **Devices in this Group**: This zone shows a list of devices that have been assigned to this entity group. To delete the relationship between the device and the entity group, select the device and click **Delete**. To expire the relationship between the device and the entity group, select the device and click **Expire**. Applicable only for "Device" entity groups.
- **Eligible Service Points for this Group**: This zone shows a list of service points that are eligible for assignment to this entity group. The list is built from the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more service points to the group, select the service points and click **Assign**. Applicable only for "Service Point" entity groups.
- **Service Points in this Group**: This zone shows a list of service points that have been assigned to this entity group. To delete the relationship between the service point and the entity group, select the device and click **Delete**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. Applicable only for "Service Point" entity groups.
- **Customer Address Search for Grouping**: This zone lists service points that are eligible for assignment to this entity group (and provides search parameters to filter the results). To assign one or more service points to the group, select the service points from the search results and click **Assign**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".
- **Service Points**: This zone shows a list of service points that have been assigned to this SP Measurement Extract entity group. To unassign a service point from the entity group, select the device and click **Unassign**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. To create an extract request for the group, click **Add**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".

The following zones may appear as part of the portal's **Requests** tab page:

- **Requests**: This zone lists the requests made for the entity group, such as requests for measurement consumption extract. To a request for the entity group, click **Add** in the zone title bar.

Batch Processing

The Device Group Evaluation (D1-DGEV) and Service Point Group Evaluation (D1-SPGEV) batch controls are used to identify devices or service points (respectively) that are eligible for a

Copyright © 2026, Oracle and/or its affiliates.

particular group and assigns them to a specified group, or expires relationships between devices or service points (respectively) and a specified group.

Devices or service points that are not currently assigned to an entity group for a specified category will be assigned to an eligible group. Only one group for the category may be assigned. If the device or service point is currently assigned a group for that category, it will be evaluated for eligibility, and if the current group is no longer eligible, the group is expired. If the same device becomes eligible for another group, that group is assigned to the device or service point.

Parameters:

- **Group Code Category**: A group category used to identify eligible entities. Required if the **Mode** is "Evaluate Group Code Category"
- **Group Code**: A group used to identify eligible entities, Required if the **Mode** is "Expire Device Group Code", "Expire Service Point Group Code" or "Evaluate Group Code", and is optional if the **Mode** is "Evaluate Group Code Category".
- **Mode** (Required): The mode by which the process evaluates entities. Options include:
  - **Expire Device Group Code (D1ED)**: Used to expire device entity group relationships for the specified Group Code
  - **Expire Service Point Group Code (D1ES)**: Used to expire service point entity group relationships for the specified Group Code
  - **Evaluate Group Code (D1EG)**: Used to search entities based on Group Code, and assigns eligible devices or service points to the specified Group Code
  - **Evaluate Group Code Category (D1GC)**: Used to search entities based on Group Code Category (and optionally a Group Code). This mode assigns devices or service points eligible for a group of the specified category (and Group Code if provided), and expires relationships between devices or service points and groups of the specified category (and Group Code is provided).

Note that these batch controls are used with groups whose Group Assignment Option is either "System" or "System or User".

Weather Stations

You use the **Weather Station** portal to display and maintain weather stations.

Refer to Understanding Weather Stations for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Weather Station**.

The following zones may appear as part of the portal's **Main** tab page:

- **Weather Station List**: This zone lists all weather stations defined in the system. Broadcast a record to display the details of the selected record.
- **Weather Station**: This zone displays the selected weather station.
- **Weather Station Characteristics**: This zone displays characteristics for the selected weather station.

Copyright © 2026, Oracle and/or its affiliates.

Importing Weather Stations

Weather stations can be imported along with other facility and network location data such as transformers, feeders, and substations. Refer to Importing Facility and Network Location Data from a Distributed Energy Resource System for more information.

Creating Weather Meters

Weather meters, including device configurations and measuring components, are created automatically when a weather station is created, based on the Device Type selected when creating the weather station.

This process requires the following data be set up prior to creating weather stations.

Measuring Component Type(s)

One or more measuring components used to capture weather data. Each should be defined as follows:

- **Business Object**: Interval Channel Type - Physical (D1-IntervalChannelTypePhysical)
- **Measuring Component Business Object**: Interval Channel (D1-IntervalChannel)
- **Measurement Business Object**: Measurement (D1-Measurement)
- **Service Type**: The service type associated with how weather data stored by measuring components of this type will be used (typically "Electric").
- **Interval /Scalar**: Interval
- **Consumptive/Subtractive**: Consumptive
- **Interval Size**: The size of the intervals for weather data values stored by measuring components of this type, represented as hours:minutes:seconds (HH:MI:SS)
- **Value Identifiers**: One or more value identifiers for the weather data captured by measuring components of this type (Degrees F, Degrees C, and so on).

Device Configuration Type

A device configuration type that defines the types of measuring components to be created for the weather meter. This should be defined as follows:

- **Business Object**: Device Configuration Type (D1-DeviceConfigurationType)
- **Device Configuration Business Object**: Device Configuration (D1-DeviceConfiguration)
- **Service Type**: The service type associated with how weather data stored by devices of this device configuration type will be used (typically "Electric") This must match the service type defined for the measuring component type.
- **Valid Measuring Component Types**: One or more measuring components (defined above) used to capture weather data. Each should be marked as "Required".

Device Type

A device type that defines the type of weather meter to be created. This should be defined as follows:

- **Business Object**: Weather Meter Type (D1-WeatherMeterType)
- **Device Business Object**: Weather Meter (D1-WeatherMeter)

Copyright © 2026, Oracle and/or its affiliates.

- **Service Type**: The service type associated with how weather data stored by devices of this type will be used (typically "Electric"). This must match the service type defined for the device configuration type and measuring component type.
- **Valid Device Configuration Types**: A device configuration type that defines the measuring component types to be created for the weather meter.
- **Characteristics**: Characteristics that define the type of data stored by the device (such as temperature)

Batch Processing

The Map Service Points to corresponding Weather Stations (D1-MAPWS) batch process can be used to map service points (including metered service points) to weather stations when and as needed.

The Initial Data Export D1-WTHRSTN (D1WSTNX) batch process can be used to export weather station data when and as needed.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Importing Weather Data

Weather data can be imported into the system using the Adapter Development Kit (ADK) in the same manner as used for measurement data. When importing weather data, the weather station ID should be used as the Device ID (the Serial Number on Weather Meters is based on the Weather Station ID). In addition, this requires that appropriate Unit of Measure (UOM), Time of Use (TOU), and Service Quantity Identifiers (SQI) values are defined to represent the weather data used in your implementation, and corresponding UOM translation mapping values are defined in the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup.

See About Measurements and Measurement Processing in the *Business User Guide*, and Configuring Payload Processing for more information.

Postal Code Defaults

You use the Postal Code Default portal to display and maintain postal code defaults.

Refer to Understanding Postal Code Defaults for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Postal Code Default - MDM**.

The following zones may appear as part of the portal's **Main** tab page:

- **Postal Code Default List**: This zone lists all postal code defaults defined in the system. Broadcast a record to display the details of the selected record.
- **Postal Code Default**: This zone displays the selected postal code default.
- **Postal Code Default Characteristics**: This zone displays characteristics for the selected postal code default.

Batch Processing

The Data Export D1_POSTAL_DFLT (D1POSTTX) batch process can be used to export postal code default data when and as needed. Use the **Batch Control** portal to view the parameters used by this batch process.

Copyright © 2026, Oracle and/or its affiliates.

Configuring M&V Options

Use the information in this section when defining M&V options used with Customer Program Management.

- Defining M&V Subscription Types
- Defining M&V Calculation Groups
- Defining M&V Calculation Rules

Defining M&V Subscription Types

You use the **M&V Subscription Type** portal to display and maintain M&V subscription types.

Refer to Understanding M&V Subscription Types for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **M&V Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **M&V Subscription Type List**: This zone lists all M&V subscription type records. Broadcast a record to display the details of the selected record.
- **M&V Subscription Type**: This zone displays the selected M&V subscription type.

Defining M&V Calculation Groups

You use the **M&V Calculation Group** portal to display and maintain Measurement and Verification calculation groups.

Refer to Understanding M&V Calculation Groups for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **M&V Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **M&V Calculation Group**: This zone displays basic information about M&V calculation group.
- **M&V Calculation Rules List**: This zone lists the M&V calculation rules belonging to the group.

Defining M&V Calculation Rules

You use the **M&V Calculation Rule** portal to display and maintain Measurement and Verification calculation rules.

Refer to Understanding M&V Calculation Rules and Understanding M&V Calculation Groups for more information.

You can access the portal by selecting **Admin**, select **Customer Program Management**, then select **M&V Calculation Rule**. You are brought to a query portal with options for searching.

Copyright © 2026, Oracle and/or its affiliates.

Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **M&V Settlement Calculation Rule**: This zone displays the M&V calculation rule, including parameters used when executing the rule.
- **Eligibility Criteria List**: This zone lists eligibility criteria for the current calculation rule. Eligibility criteria are configured criteria used to determine whether to execute a specific calculation rule.

M&V Calculation Rules

The following is a list of the calculation rules provided as part of base product that are used with M&V calculations. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Program Settlement Calculation Rule Name	Purpose
Vector and Service Quantity Math (D2-Math)	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations.

Defining Customer Program Service Task Types

You use the **Service Task Type** portal to display and maintain customer program service task types.

Refer to Understanding Customer Program Service Task Types for more information.

You can access the portal by selecting **Admin**, select **General**, then select **Service Task Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Task Type List**: This zone lists all service task types defined in the system. Broadcast a record to display the details of the selected record.
- **Service Task Type**: This zone displays the selected service task type.

Defining Customer Program Service Order Management Options

Service order management options used with customer program management include the following:

- Service Order Field Activity Types
- Field Task Types
- Customer Program Service Task Types
- Communicating with the Field Work System

Service Order Field Activity Types

A single service order field activity type must be configured to support communication with external field work systems such as Oracle Fusion Field Service. Service order activity types should be assigned to the “Field Activity” Activity Type Category and should be created using the Field Activity Type (D1-FieldActivityType).

Copyright © 2026, Oracle and/or its affiliates.

See Understanding Service Order Field Activity Types and Configuring Service Order Field Activity Types for more information.

Field Task Types

A service order field activity's field task type defines details about the type of task to be performed and how the system will process the activity.

Field task types are defined using the Field Task Type (D1-FieldTaskTypeLookup) extendable lookup.

Field task types are used in the **Field Task Type** drop-down in the **Setup Action Details** section of the service task type when using service order management (see Understanding Customer Program Service Task Types).

The following pre-configured field task types are used with Customer Program Management:

- Exchange Controllable Device (X1-ExchangeControllableDevice)
- In Home Energy Assessment (X1-InHomeEnergyAssessment)
- Install Controllable Device (X1-InstallControllableDevice)
- Investigate Controllable Device (X1-InvestigateControllableDevice)
- Remove Controllable Device (X1-RemoveControllableDevice)

Where applicable, these field task types have completion events defined as **Completion Events When Successful**, meaning those completion events are created and processed is the field work is completed successfully.

See Understanding Field Task Types and Configuring Field Task Types for more information.

Note: Customer Program Management does NOT use Service Order Activities (also referred to Orchestration Activities such as Enable Service, Disable Service and so on). Instead customer program management service tasks create service order field activities.

Customer Program Service Task Types

When using Service Order Management for customer program field activities, customer program service task types should be configured to use the following settings:

- **Setup Action**: Create Activity
- **Activity Type**: The service order field activity type created for service order management (see **Service Order Field Activity Types**).
- **Field Task Type**: The specific field task type, based on the service task (see **Field Task Types**).

See Understanding Customer Program Service Task Types for more information.

Communicating with the Field Work System

Communication with the field work is supported via outbound messages and inbound web services.

Outbound Messages

Outbound messages are created by the Create Outbound Message for Activity Add (D1-COUTMSG) algorithm on the "Awaiting Response" state of the outbound communications. This uses the Field Activity processing role defined on the external application. These outbound messages contain the customer and program information from the service order field activity that created the outbound communication.

Copyright © 2026, Oracle and/or its affiliates.

See Outbound Messages and Service Order Management External Applications for more information.

Inbound Web Services

Upon completion of the field work, the field work system can use the Mobile Activity Completion (X1-MblActCom) inbound web service to send an inbound communication containing details of the completed work. The message sent to the inbound web service must contain the customer and program information sent in the original outbound communication and message.

See Inbound Web Services for more information.

Defining Enrollment Options

This section outlines the configuration needed to support enrollment and unenrollment processes.

Create Service Task Types

You must create one or more service task types for different types of enrollment and unenrollment processes. The number of service task types needed is based on the number of programs, program subscription types, and device location types you have configured.

See Understanding Customer Program Service Task Types and Defining Customer Program Service Task Types for more information about working with customer program service task types.

Create Action Method

You must also configure an Action Method to enroll or unenroll customers using the **Process Flow** portal.

Use the following procedure to create an action method used with customer program enrollment or unenrollment processing.

1. Select **Admin**, select **System**, then select **Action Method**, **Add**.
2. Select "Customer Program Enrollment Action Method" or "Customer Program Unenrollment Action Method" from the **Action Method Business Object** drop-down list on the **Select Business Object** screen and click **OK**.
3. Enter the following on the **Action Method** screen:
   - **Action Method**: A code for your action method
   - **Action Method Role**: Customer Program Enrollment (or Customer Program Unenrollment)
   - **Description**: A description of your request type
   - **Default Details**:
     - **Process Flow Type**: Customer Program Enrollment Process (or Customer Program Unenrollment Process)
     - **Enable Bring Your Own**: Indicates if the "Bring Your Own" option is available on the Device Setup panel of Enrollment process flows. Only available for action methods based on the "Customer Program Enrollment Action Method" business object.

Copyright © 2026, Oracle and/or its affiliates.

- **Appointment Resource Option**: Indicates if available appointments should be based on an Activity Type or Template Activity (default). If set to "Template Activity", appointment slots are selected based on labor requirements defined in the Template Activity specified in the Enrollment/Unenrollment service task type as well as the selected crew shift type and crew.
- **Appointment Slot Days**: The number of days added to the selected appointment date to build a date range for the list of available appointment slots on the Device Setup panel of Enrollment or Unenrollment process flows.
- **Show Application Date**: Indicates whether or not the enrollment or unenrollment date should be displayed on Enrollment or Unenrollment process flow panels.

4. Click **Save**.

See Defining Action Methods for more information about creating Action Method records.

Defining Program Service Call Options

This section outlines the configuration needed to support service call processes.

Create Service Task Types

You must create one or more service task types for different types of service call processes. The number of service task types needed is based on the number of programs, program subscription types, and device location types you have configured.

See Understanding Customer Program Service Task Types and Defining Customer Program Service Task Types for more information about working with customer program service task types.

Create Action Method

You must also configure an Action Method to create service calls for customers using the Process Flow portal.

Use the following procedure to create an action method used with customer program service call processing.

1. Select **Admin**, select **System**, then select **Action Method**, **Add**.
2. Select "DA Program Service Call Action Method" from the **Action Method Business Object** drop-down list on the **Select Business Object** screen and click **OK**.
3. Enter the following on the **Action Method** screen:
   - **Action Method**: A code for your action method
   - **Action Method Role**: Program Service Call
   - **Description**: A description of your request type
   - **Default Details**:
     - **Process Flow Type**: Program Service Call Process
     - **Appointment Resource Option**: Indicates if available appointments should be based on an Activity Type or Template Activity (default). If set to "Template Activity", appointment slots are selected based on labor requirements defined in the Template Activity specified in the program call service task type as well as the selected crew shift type and crew.

Copyright © 2026, Oracle and/or its affiliates.

- **Appointment Slot Days**: The number of days added to the selected appointment date to build a date range for the list of available appointment slots on the Device Setup panel of program call process flows (maximum value of 9 days)

4. Click **Save**.

See Defining Action Methods for more information about creating Action Method records.

Defining Manual Work Activity Completion Options

This topic outlines the configuration needed to support manual work order activity completion processes.

Create Action Method

You must configure an Action Method to manually complete work activities using the Process Flow portal.

Use the following procedure to create an action method used with manual work order activity completion processing.

1. Select **Admin**, select **System**, select **Action Method**, then select **Add**.
2. Select "Work Activity Completion Action Method" from the **Action Method Business Object** drop-down list on the **Select Business Object** screen and click **OK**.
3. Enter the following on the **Action Method** screen:
   - **Action Method**: A code for your action method
   - **Action Method Role**: Work Activity Completion
   - **Description**: A description of your request type
   - **Process Flow Type**: Work Activity Completion Process
4. Click **Save**.

See Defining Action Methods for more information about creating Action Method records.

Configuring the Customer Program Summary Portal

This topic describes the process for configuring the **Customer Program Summary** portal.

Refer to the Using the Customer Program Summary Portal section of the *Business User Guide* for more information on this functionality.

The zones on the **Customer Program Summary** portal leverage a number of insight types and insight groups. Insight types and insight classes are provided with the base package, but you must create insight groups that contain specific insight types in order for these zones to display the correct information.

The table below lists the zones in the **Customer Program Summary** portal, along with details regarding the insight group that must be created for each zone, including the insight class, the visible structure, and the insight type to be included in the insight group.

Copyright © 2026, Oracle and/or its affiliates.

Zone	Insight Group	Insight Class	Valid Visual Structures	Insight Type
Current Enrollment	Current Enrollment	Program Enrollment	Card Insight	Customer Current Program Enrollment (X1-DACSCURPE)
Customer Insights	Customer Insights	Customer Insights	Card Insight	In Progress Service Tasks (X1-DACSINPRGSVCTASKS)
Customer Program Activity	Customer Event Participation	Customer Participation	Inline Insight	Customer Event Participation (X1-DACSCSPRT)
Eligible Programs	Eligible Programs	Eligible Programs	Card Insight	Eligible Customer Program (X1-DACSELPGM)

Refer to Contextual Insights, Insight Groups, and Defining Insight Groups for more information about configuring insight groups.

Configuring the Program Dashboard

This section describes the process for configuring the **Program Dashboard**.

Refer to the Using the Program Dashboard section of the *Business User Guide* for more information on this functionality.

This dashboard leverages a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on statistics snapshots.

Configuring the **Program Dashboard** includes the following:

Master Configuration

To access the Master Configuration portal select **Admin**, select **General**, then select **Master Configuration**.

Select the Program Dashboard Configuration master configuration.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

This master configuration includes the following options:

- **Color Options**: Allows configuration of the colors used for Enrolled, Unenrolled, and Active data on the graph.
- **Chart Options**: The **Stacked Bars** option specifies if bar graphs for enrollments and unenrollments should display as stacked bars (checked) or as a part of side by side bars (unchecked).

Configure Statistic Control and Statistic Snapshots

Statistic controls are the source for capturing the numbers of enrolllments, unenrollments, and active program subscriptions.

Copyright © 2026, Oracle and/or its affiliates.

Statistic snapshots capture the numbers for each metric. Statistic controls are configuration records that define the specific types of metrics to capture.

Statistic snapshots are captured periodically via the F1-STATS (Statistics Control Monitor) batch program.

Configure the following Statistics Control records:

Statistics Control	Description	Category	Status	Snapshot Frequency Hours	Retention Mode	Snapshot Business Object
MONTHLY	Program Dashboard Statistics Control - Monthly	Operational	Active	24.00	Retain All Snapshots	Subscription Trend Snapshot - Monthly (X1-SubscriptionTrendMonthly)
WEEKLY	Program Dashboard Statistics Control - Weekly	Operational	Active	24.00	Retain All Snapshots	Subscription Trend Snapshot - Weekly (X1-SubscriptionTrendWeekly)

See Capturing Statistics for more information about statistics.

Configure Subscription Trend Snapshot Business Objects and Algorithms (Optional)

By default, the Subscription Trend Snapshot business objects are configured to capture 13 months of data per snapshot. This can be changed by adjusting the value of the “Period Bucket Limit” parameters on the Create Subscription Trend Snapshot algorithms (defined as Enter algorithms on the Complete state of the snapshot business objects).

The table below lists the snapshot business objects and their respective algorithms.

Snapshot Business Object	Create Subscription Trend Snapshot Algorithm
Subscription Trend Snapshot - Monthly (X1-SubscriptionTrendMonthly)	Create Subscription Trend Snapshot - Monthly (X1-CRESTS)
Subscription Trend Snapshot - Weekly (X1-SubscriptionTrendWeekly)	Create Subscription Trend Snapshot - Weekly (X1-CRESTSWKY)

Configure Statistics Control Business Object

The Statistics Control (F1-Statistics) business object should be configured to specify the "Statistics Control Monitor" batch control in the **Monitor Process** drop-down list on the "Active" state.

To configure this business object, do the following:

1. Open the Statistics Control (F1-Statistics) business object in the **Business Object** portal.
2. Navigate to the **Lifecycle** tab, and expand the "Active" state.
3. Select "Statistics Control Monitor" from the **Monitor Process** drop-down list.
4. Click **Save**.

Copyright © 2026, Oracle and/or its affiliates.

Batch Processing

The **Program Dashboard** dashboard is configured to be refreshed regularly (as frequently as daily). To refresh the data used in the dashboard, the F1-STATS batch process should be processed in sequence after other regularly schedule batch processing is complete to generate statistic snapshots based on statistic controls.

Use the following parameters when running this batch process:

- Maintenance Object (maintenanceObject): F1-STATS
- Restrict by Business Object (restrictToBusinessObject): F1-Statistics
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Adding Custom Fields to Customer Program Process Flows

The Customer Program process flows provided with the base package, including enrollment and unenrollment, are based on predefined process flow types that, among other things, specify the data area used by each panel in the process flow. Implementations can add custom fields to process flow panels by extending the data areas used by each panel. This involves creating new data areas that reference the existing data areas associated with each panel as an "Extended Data Area".

To extend a process flow:

- Identify the fields to be added. If needed, create the new fields and other related data, such as lookups, extendable lookups, and so on. Refer to Fields for more information about fields.
- Create a new data area.
  - Specify the data area you wish to extend in the **Extended Data Area** field.
  - Define the new fields in the schema of your new data area.
  - Refer to Data Areas for more information about creating data areas.

Process Flow Types and Data Areas

This section provides details around the data areas used with the Digital Asset process flow types.

The process flow types provided with the base package include:

- Program Enrollment Process (X1-DIGITALASSETENROLLMENT)
- Program Unenrollment Process (X1-DIGITALASSETUNENROLLMENT)
- Program Program Service Call Process (X1-DAPROGRAMSERVICECALL)

The tables below list the data areas defined in these process flow types. These are the data areas that can be extended as needed. For example, to add custom fields to the Program Enrollment panel, you would extend the Digital Asset Enrollment - Program (X1-DAEnrollEligibility) data area.

Enrollment Process Flow Data Areas

The table below lists the base package data areas used by each panel for the Digital Asset Enrollment Process (X1-DIGITALASSETENROLLMENT) process flow type.

Copyright © 2026, Oracle and/or its affiliates.

Process Flow Panel	Data Area
Customer	Digital Asset Enrollment - Customer (X1-DAEnrollCustomer)
Program Enrollment	Digital Asset Enrollment - Program (X1-DAEnrollEligibility)
Related Program Enrollment	Digital Asset Enrollment - Related Program (X1-DAEnrollRelatedProgram)
Device Setup	Digital Asset Enrollment - Device Setup (X1-DAEnrollDeviceSetup)
Summary	Digital Asset Enrollment - Summary (X1-DAEnrollSummary)

Unenrollment Process Flow Data Areas

The table below lists the base package data areas used by each panel for the Digital Asset Unenrollment Process (X1-DIGITALASSETUNENROLLMENT) process flow type.

Process Flow Panel	Data Area
Customer	Digital Asset Unenrollment - Customer (X1-DAUnenrollCustomer)
Program Enrollment	Digital Asset Unenrollment - Program (X1-DAUnenrollProgram)
Device Setup	Digital Asset Unenrollment - Device Setup (X1-DAUnenrollDeviceSetup)
Summary	Digital Asset Unenrollment - Summary (X1-DAUnenrollSummary)

Program Service Call Process Flow Data Areas

The table below lists the base package data areas used by each panel for the Digital Asset Program Service Call Process (X1-DAPROGRAMSERVICECALL) process flow type.

Process Flow Panel	Data Area
Customer	Digital Asset Program Service Call Customer (X1-DAProgramServiceCallCust)
Trouble Report	Digital Asset Program Service Call Action (X1-DAProgramServiceCallAction)
Summary	Digital Asset Program Service Call Summary (X1-DAProgramServiceCallSummary)

Customer Program Management Integrations

This section describes options for integrating Customer Program Management with other systems.

- Customer Program Management Inbound Web Services
- Grid Edge Distributed Energy Resources Management System Integrations
- Oracle Utilities Customer Cloud Service Integrations
- Oracle Field Service Integrations
- Customer Information System Integrations

Copyright © 2026, Oracle and/or its affiliates.

- Analytics Application Integrations
- Importing Data from External Systems
- Extracting M&V Data for use in External Systems

Customer Program Management Inbound Web Services

Customer Program Management includes inbound web service REST APIs that can be used for customer enrollment and unenrollment, managing event participation, and other processes. Refer to Inbound Web Services for general information about inbound web service options.

Use the **Inbound Web Service** portal to view additional details about these web services and their supported operations.

Program Enrollments

The Program Enrollments (X1-ProgramEnrollment) inbound web service can be used to enable submission of program enrollment and unenrollment requests from external systems such as a customer information system. This web service includes the following operations:

- Create Program Enrollment (enrollProgram): Used to create an enrollment request to enroll a customer in a program
- Create Program Unenrollment (unenrollProgram): Used to create an unenrollment request to unenroll (remove) a customer from a program
- Enroll Additional Device (enrollDevice): Used to create a service task to add a device to an existing program enrollment
- Get Program Enrollment (getProgramEnrollment): Used to retrieve enrollment information
- Get Program Unenrollment (getProgramUnenrollment): Used to retrieve unenrollment information
- Post Acknowledgement (postAcknowledgement): Used to send an acknowledgement containing a list of devices and their corresponding Head End Registration Status.

See About Enrollment and Unenrollment in the *Business User Guide* for more information about the processes used to enroll and unenroll customers in programs.

Program Event

The Program Event (X1-ProgramEvent) inbound web service can be used to create and update program events from external systems such as an advanced distribution management system (ADMS) such as Oracle Utilities Network Management System or a supervisory control and data acquisition (SCADA) system. This web service includes the following operations:

- Create Program Event (createProgramEvent): Used to create program events. Program events begin in the "Scheduled" state (Created in Pending, but auto-transitions to "Scheduled")
- Action Program Event (actionProgramEvent): Used to transition event to different states, such as In Progress, Complete, or Cancelled.

See See About Program Events in the *Business User Guide* for more information about program events.

Program Event Participation

The Program Event Participation (X1-ProgramEventParticipation) inbound web service can be used to send event participation data from external systems such as an advanced distribution

Copyright © 2026, Oracle and/or its affiliates.

management system (ADMS) such as Oracle Utilities Network Management System or a supervisory control and data acquisition (SCADA) system. This web service includes the following operations:

- Reserve Device (reserveDevice): Used to "reserve" devices for an upcoming event by creating records in the Event Participation table. This process can also create an event if it doesn't already exist.
- Post-Event Update (postEventOutcome): Used to update device event participation records with the outcome of the event

See About Event Participation in the *Business User Guide* for more information about event participation.

Program Eligibility and Qualification Questions

The Program Eligibility and Qualification Questions (X1-ProgramEligibilityAndQualification) inbound web service can be used to retrieve the eligible programs for a customer, each program's questions that will be asked to the customer during the enrollment process, and expected answers from a customer in order to qualify in the program. This web service includes the following operations:

- Get Eligible Programs (getEligiblePrograms): Used to retrieve the eligible programs for a customer
- Get Informational Questions (getInformationalQuestions): Used to retrieve a program's informational questions used in the enrollment process, and expected answers from a customer.
- Get Qualification Questions (getQualifyingQuestions): Used to retrieve a program's qualification questions used in the enrollment process, and expected answers from a customer in order to qualify in the program

See About Programs in the *Business User Guide* for more information about program qualification and eligibility.

Controllable Device Status

The Controllable Device Status (X1-ControllableDeviceStatus) inbound web service can be used to receive the status of a controllable device from an external application. This web service includes the following operations:

- Post Controllable Device Status Check (postControllableDeviceStatus): Used by an external application to send the status of a controllable device

See About Controllable Assets and Devices in the *Business User Guide* for more information about controllable devices.

Current Program Enrollment

The Current Program Enrollment (X1-CurrentProgramEnrollment) inbound web service can be used to retrieve the current program enrollment for the specified customer. This web service includes the following operations:

- Retrieve Program Enrollment (retrieveProgramEnrollment): Used to retrieve the current program enrollment for the specified customer

See About Enrollment and Unenrollment in the *Business User Guide* for more information about the processes used to enroll and unenroll customers in programs.

Copyright © 2026, Oracle and/or its affiliates.

Program Context

The Program Context (X1-ProgramContext) inbound web service can be used to retrieve customer context information comprising name value pairs, which can help indicate various programs in which the customer may be enrolled. This web service includes the following operations:

- Get Program Context (getProgramContexts): Used to retrieve the program context for a customer

See About Enrollment and Unenrollment in the *Business User Guide* for more information about the processes used to enroll and unenroll customers in programs.

Appointment Tasks

The Appointment Tasks (X1-AppointmentTasks) inbound web service can beused to request and track appointments related to service order field activities for the customer via self-service portals. This web service includes the following operations:

- Book Appointment (bookAppointment): Used to book an appointment
- Cancel Appointment (cancelAppointment): Used to cancel an appointment
- Retrieve Pending Tasks (retrievePendingTasks): Used to retrieve a list of pending tasks
- Search Appointment Slots (searchAppointmentSlots): Used to search for available appointment slots
- Search For Tasks (searchForTasks): Used to search for tasks

Work Activity Appointment Tasks

The Work Activity Appointment Tasks (X1-WorkActivityAppointmentSlots) inbound web service can be used to request and schedule appointment slots related to work activities. This web service includes the following operations:

- Get Work Activity Appointment Slots (getAppointmentSlots): Used to get available appointment slots for a work activity
- Schedule Activity (scheduleAppointment): Used to schedule an appointment for a work activity

DER Device Management

The DER Device Management (X1-DERDeviceManagement) inbound web service can be used to perform registration and/or enrollment of Distributed Energy Resource (DER) devices. This web service includes the following operations:

- Register DER Device (registerDERDevice): Used by an external application to register and enroll a DER device. This operation creates a DER asset and device, as well as an enrollment service task to enroll the device in a program specified in the inbound web service request, or if a program is not provided based on the specification defined for the asset. The enrollment service task is created in the "Pending" state. The service task should be processed either manually or via the Service Task Monitor (Deferred) (F1-STKDF) batch process.
- DER Interconnection (interconnectDER): Used by an external application to send interconnection attributes of a DER asset from an external system that handles the interconnection process for the utility. The incoming data is validated against existing asset manufacturer and model combinations, leading to the creation of an interconnection service task to allow the initial creation of the DER asset. Enrollment for assets that require

Copyright © 2026, Oracle and/or its affiliates.

interconnection attributes must be performed separately via the Register DER Device operation. The interconnection service task is created in the "Pending" state. The service task should be processed either manually or via the Service Task Monitor (Deferred) (F1-STKDF) batch process.

Registration of DER devices includes populating various attributes of the device and related specifications. Specification attribute and attribute values must be defined for the specification prior to registration of DER devices related to specification.

See About Controllable Assets and Devices, Maintaining Controllable Assets and Device, and Maintaining Specifications in the *Business User Guide* for more information about controllable devices and where asset and specification attribute values are maintained.

Device Verification

The Device Location Verification (D1-DeviceVerificationService) SOAP inbound web service can be used to perform verification of equipment associated with a meter or service point. This service performs the following validations:

- **Device Validation:**
  - The device must exist for the given Serial Number and NIC ID.
  - The device must not currently be installed at any program location.
- **Program Location Validation**
  - The program location must exist.
  - The program location must be associated with the provided service point.

Grid Edge Distributed Energy Resources Management System Integrations

The Oracle Utilities Customer Program Management Cloud Service Integration to Oracle Utilities Distributed Energy Resources Management System supports integration with Oracle Utilities Network Management System and other components of the Oracle Utilities DERMS solution. This integration provides the following functionality:

- **Enrollment Synchronization**: As customers are enrolled in programs, enrollment data is synchronized with Oracle Utilities Network Management System.
- **Enrollment Request**: When customers enroll controllable devices in a program through a head end system provider or aggregator/vendor, enrollment requests are sent to Oracle Utilities Live Energy Connect, which then sends the enrollment request to Customer Program Management.
- **Device Data Synchronization**: As controllable devices are created and updated, device data is synchronized with Oracle Utilities Network Management System. When Distributed Energy Resource (DER) devices are registered and created, device data includes asset and specification attributes and values. When device data is synchronized, Oracle Utilities Network Management System sends an acknowledgement containing a list of devices and their corresponding Head End Registration Status and the appropriate devices are updated accordingly. If devices are grouped via Entity Groups, grouping information is included in the device data synchronized with Oracle Utilities Network Management System.
- **Device Population Synchronization**: As device population groups are created and updated, device population data is synchronized with Oracle Utilities Network Management System. When device population data is synchronized, Oracle Utilities Network Management System sends an acknowledgement and the device population group is updated to reflect successful synchronization.

Copyright © 2026, Oracle and/or its affiliates.

- **Customer and Device Unavailability**: If customers and their controllable devices are deemed unavailable to participate in program events via Customer Unavailability events, customer unavailability information is sent to Oracle Utilities Network Management System. If changes are later made to a customer unavailability period, those changes are sent to Oracle Utilities Network Management System.
- **DER Aggregator Data Synchronization**: As DER Aggregators are created and updated, aggregator data is synchronized with Oracle Utilities Network Management System. DER Aggregator data synchronization includes contact and program offering information. Note that deletion of DER Aggregators does **not** trigger data synchronization.
- **Control Score Extract**: Device control scores are extracted and sent to Oracle Utilities Network Management System on a daily basis. DERMS operators use this control score information (in addition to other data) to identify sets of devices that should participate in program events. Note that devices associated with programs whose program type has a **Program Category** of "DER Registration" are excluded from control score calculations and extract.
- **Event Participation**: Oracle Utilities Network Management System sends device reservation information to create event participation records (and events if they don't already exist in the system). If a customer opts out of an event, a notification is sent to Oracle Utilities Network Management System to exempt the device from the event.
- **Event Outcome - Device Score Calculation**: Oracle Utilities Grid Edge Distributed Energy Resources Management System sends event outcome details after scheduled events. This ensures that the actual program event participation outcome for each controllable device is correctly updated and can be factored into the controllable device score calculation.
- **Program Rule Synchronization:** When a Device Score rule is added to program, or an existing Device Score Program Rule associated with a program is edited, program rule data is synchronized with Oracle Utilities Network Management System.

Refer to the **Oracle Utilities Customer Program Management Cloud Service Integration to Oracle Utilities Distributed Energy Resources Management System** documentation for more information about this integration.

Oracle Utilities Customer Cloud Service Integrations

The Oracle Utilities Customer Cloud Service Integration to Oracle Utilities Customer Program Cloud Service supports integration with Oracle Utilities Customer Cloud Service. This integration provides the following functionality:

- Initial and incremental extract of customer data including persons, service points, and service agreements from Oracle Utilities Customer Cloud Service (or Oracle Utilities Customer to Meter or Oracle Utilities Customer Care and Billing). See **Customer Data Extract** in the *Oracle Utilities Customer Cloud Service User Guides* for more information about this process.
- Initial and incremental import of customer data into Oracle Utilities Customer Program Management Cloud Service. See Importing Customer-Related Data from a Customer Information System for details regarding this process. Customer Program Management Cloud Service can be configured to send an acknowledgement to Customer Cloud Service upon import of customer data.

Refer to the Oracle Utilities Customer Cloud Service Integration to Oracle Utilities Customer Program Cloud Service documentation for more information about this integration.

Copyright © 2026, Oracle and/or its affiliates.

Oracle Field Service Integrations

The Oracle Utilities Customer Program Management Cloud Service Integration to Oracle Field Service supports integration with Oracle Field Service. This integration provides the following functionality:

- **Work Activity Request**: Customer Program Management creates work orders and associated work activities as part of the device setup step within the enrollment or unenrollment process for utility-owned controllable devices. Work activities can also be created for supporting investigation work as part of troubleshooting. These work activities contain customer contact information, information about any related programs, and key information related to the type of work, the work window, its duration, required crew, and details to be captured during the work. Work activities are sent to Oracle Field Service so that crews can be assigned as appropriate.
- **Work Activity Completion**: When the work activities are completed, Oracle Field Service sends work completion details back to Customer Program Management, which in turn continues processing of the enrollment or unenrollment request.

Refer to the **Oracle Utilities Customer Program Cloud Service Integration to Oracle Field Service** documentation for more information about this integration.

Customer Information System Integrations

This section describes integrations with customer information systems.

- Sending Customer Settlement Transaction Requests
- Downloading Customer Settlement Transactions

Sending Customer Settlement Transaction Requests

Customer settlement transactions are created upon requests from a customer information system (CIS). Customer settlement transaction requests are sent in a file which is processed by a batch process which creates the individual customer settlement transactions. Customer settlement transaction requests include the customer contact or account or metered service, the program, the start and end date for the requested time period, and the type of request.

See About Customer Settlement Transactions for more information.

Uploading Customer Settlement Transaction Requests

Customer settlement requests are uploaded using the Customer Settlement Transaction Upload (X1-CSTUP) batch process. This batch process uploads data from a .tab-delimited file. For each data row in the file, this process creates a Customer Settlement Transaction record.

This batch processes use the following parameters:

- **File Name**: The name of the file to upload. File names can use wildcards (Required)
- **Flat Path**: The file location where the upload file is located. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* for more information. (Required)
- **Processed File Extension**: The text appended to the file name after the file has been processed. Default is "PROCESSED".

Copyright © 2026, Oracle and/or its affiliates.

- **Error On No Files Found**: Indicates whether or not an error is issued if no files are found matching the given name in the given path. Valid values are "true" (default) and "false".

Use the **Batch Control** portal to learn more about these and other parameters used by these batch process.

Customer Settlement Request Upload File Format

Customer settlement request upload files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Settlement Request ID	Required	An id for the customer settlement request from the CIS application.
External Contact ID*	Required*	The external id of the customer for which the Customer Settlement will be created. For requests from Customer Care and Billing or Customer to Meter, this is the Person ID.
External Account ID*	Required*	The external id of the account for which the Customer Settlement will be created. For requests from Customer Care and Billing or Customer to Meter, this is the Account ID.
External Metered Service ID*	Required*	The external id of the metered service for which the Customer Settlement will be created. For requests from Customer Care and Billing or Customer to Meter, this is the Service Agreement ID.
External Program ID	Required	The external ID of the program for which the Customer Settlement will be created. This is used for filtering in case the customer is enrolled in multiple programs.
Start Date	Required	The date on which the customer settlement period starts. This assumes a start time at the start of the day.
End Date	Required	The date on which the customer settlement period ends. This assumes a start time at the end of the day.
Customer Settlement Request Type	Required	A flag that indicates the type of customer settlement request. Valid values are values for the Customer Settlement Request Type (EVT_SETT_REQ_TYPE_FLG) lookup, and include: • D1FL (Final) • D1PC (Periodic) • D1RL (Rebill) • D1UT (Unenrollment)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Included Transactions	Required	A flag that indicates the types of transactions included in the customer settlement. Valid values are values for the Included Transaction (INCLUDED_TRANSACTION_FLG) lookup, and include: • X1BO (Both) • X1EV (Event Settlement) • X1IF (Incentive and Fee)

* Only 1 of the fields markers with an asterisk (*) can be included:

- If the External Contact ID is provided, the system evaluates all Metered Services for the contact.
- If the External Account ID is provided, the system evaluates all Metered Services for the account.
- If the External Metered Service ID is defined as a Settlement Group on a Program Subscription, the Customer Settlement is run with the **Settlement Aggregation Method** set to "Grouped". See About Customer Settlement Transactionsfor more information.

Sample Record

Below is a sample file:

```
Tmp5    MBFEXT0605_1    PRG002    2023-04-06    2023-04-07    D1FL    X1EV
```

Downloading Customer Settlement Transactions

Customer settlement transactions can be downloaded for import into a customer information system via a batch process which creates a delimited file containing customer settlement transaction information.

See About Customer Settlement Transactions for more information about customer settlement transactions.

Batch Control

Customer settlement transactions are downloaded using the Customer Settlements Extract (X1-CSEXT) batch process. This batch process downloads data into a delimited file, the details of which are specified by the batch control parameters.

This batch processes use the following parameters:

- **File Integration Type**: The file integration type (see below) used to extract customer settlement transactions (see below).
- **File Name**: The name of the downloaded file. File names can use wildcards (Required)
- **File Path**: The file location to which the file will be downloaded. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* the for more information. (Required)
- **File Format**: The extract file format. Valid values are CSV (delimited) FIXD (fixed position output) or XML (XML document).

Copyright © 2026, Oracle and/or its affiliates.

- **File Delimiter**: The delimiter to use for the CSV file format
- **Start Date (optional)**: A date that specifies the start of the date range for which customer settlement transactions are extracted. Used with the End Date parameter.
- **End Date (optional)**: A date that specifies the end of the date range for which customer settlement transactions are extracted. Used with the Start Date parameter.
- **External Account Id (optional)**: An external account ID used to restrict the customer settlements transactions extracted to those for the specified account.
- **Extract Option**: Flag used to re-extract previously extracted customer settlements. Valid values include:
  - **Y**: Used to Re-extract only previously extracted customer settlement transactions that match other criteria. This should only be used when specifying a Start/End Date.
  - **N**: Not Re-Extract (default)

Use the **Batch Control** portal to view additional details about these parameters.

File Integration Type

The customer settlement transaction extract process uses the Customer Settlements Extract (X1-CustomerSettlementExtract) file integration type, defined as a parameter on the Customer Settlements Extract (X1-CSEXT) batch process.

Implementations that require a different extract format can create a custom file integration type that defines their own extract format and process. This includes creating a custom extract algorithm and a custom extract data area.

Refer to Extract Using File Integration for more information about the role of file integration types.

Extract Algorithm Parameters

The Customer Settlements Extract (X1-CustomerSettlementExtract) file integration type uses the Customer Settlements Extract (X1CUSTSETEXT) algorithm to extract customer settlement transactions. This algorithm includes a "Line Charge Option" parameter with the following options:

- **A** (Aggregated): The algorithm outputs a single line charge from aggregated program settlements
- **D** (Detailed): The algorithm outputs lines charges for each program settlement

The default setting for this parameter is "D", but this can be overridden if needed by creating a new algorithm version and changing the value of this parameter.

File Format

The file format of the extract file is defined by the Customer Settlements Extract (X1-CustomerSettlementExtract) data area (specified on the Customer Settlement Extract (X1-CustomerSettlementExtract) file integration record on the Customer Settlements Extract (X1-CustomerSettlementExtract) file integration type.

Customer settlement transaction extract files use the following format:

Field Name	Mapping	Notes
Start Date	Customer Settlement Start Date	
End Date	Customer Settlement End Date	

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Mapping	Notes
Account Id	Metered Service External Account Id	
External Id	Customer Settlement Id	
Description		This will be derived message: Customer Settlement for Period (%1 - %2)
Program Code		
Saved Consumption		
Actual Consumption		
Number of Events		
Line Charges		This column contains the list of program event settlement and incentives and fees settlement amounts. Each settlement row is delimited by a semi-colon (;) and each row column is delimited by comma (,). Depending on the extract algorithm, this list will contain 1 row for Aggregated extract option or many for Detailed extract option (default).

Field Name	Mapping
Sequence	Generated
Line Description	Program Settlement Line Description
Amount	Program Settlement Amount * -1
Line Charge External Id	Event Settlement Id

Note that this format is patterned after the format used by the Billable Charge Upload Staging Table used by Oracle Utilities Customer Care and Billing, Oracle Utilities Customer to Meter, Oracle Utilities Customer Cloud Service, and Oracle Utilities Billing Cloud Service.

Analytics Application Integrations

Integration with external analytics applications provide the following functionality:

- Program Settlement Creation: The external analytics application sends files which contain the actual consumption or demand reductions achieved for the duration of the event for each customer. These records are used to determine financial settlement data. There are two types of these files:
  - "KWH Avoided" files contain consumption reduction values. See Processing KWH Avoided Files for more information.
  - "KW Drop" files contain demand reduction values. See Processing KW Drop Files for more information.
- Processing KWH Avoided Files
- Processing KW Drop Files

Processing KWH Avoided Files

An external analytics application sends "kWh avoided" files containing the actual energy reduction achieved for the duration of the event for each customer, which can be used to determine financial settlement data. These files consist of comma separated records for each participating metered service point which are sent to Oracle Cloud Object Storage. A batch

Copyright © 2026, Oracle and/or its affiliates.

process imports these files and creates business flags that capture the data for each metered service point. This section describes how the application processes these "kWh avoided" files.

KWH Avoided File Format

"kWh avoided" files use the following comma values format:

Field	Description	Example
EventId	ID of the program event	4011
EventType	The program event type	Economic
ProgramId	The Program ID for the event	1234512345
SPId	The Metered Service Point ID	1122334455
ActualStartTime	Start time of the event	2022-12-13T14:30:00-05:00
ActualEndTime	End time of the event	2022-12-13T16:30:00-05:00
IntervalSize(Seconds)	The interval size of the interval values included, represented in seconds.	900
TotalkWh	The total kWh for the event period	75.00
TotalKwhSavedForPeriod	The total kWh saved for the event period	33.00
KwhSaved<n>	Individual interval values that comprise the total kWh saved for the event period. The number of interval values is based on the duration of the event divided by the interval size.	1: 5.00 2: 3.00 3: 2.00 4: 2.00 5: 3.00 6: 8.00 7: 6.00 8: 4.00

Below is a sample file, based on the example data above:

```
EventId,EventType,ProgramId,SPId,ActualStartTime,ActualEndTime,IntervalSize(Se
conds),TotalKwh,TotalKwhSavedForPeriod,KwhSaved1,KwhSaved2,KwhSaved3,KwhSaved4
,.....
4011,Economic,1234512345,1122334455,2022-12-13T14:30:00-05:00,2022-12-13T16:30
:00-05:00,900,75.00,33.00,5.00,3.00,2.00,2.00,3.00,8.00,6.00,4.00
```

Batch Processing

"kWh avoided" files are processed using the kWh Avoided Business Flag Upload (X1-KWHAV) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be scheduled to run on a regular basis.

This batch processes uses the Create kWh Avoided Business Flag (X1-KWHAVDBUS) algorithm (defined for the "File Upload" **System Event** on the **Algorithms** tab of the **Batch Control** portal). This algorithm uses the following parameters:

- **Business Flag Standard Name**: The Standard Name (X1-KWHAVOIDED) used to identify the Business Flag Type for business flags created by the batch process. See Understanding Business Flags and Standard Name for more information about business flag standard names.

Copyright © 2026, Oracle and/or its affiliates.

- **Interval Data Condition Value**: The usage transaction service quantity condition code for the interval data values captured in the business flag, from the Usage Transaction SQ Interval Condition (D2-UTSQIntvlConditionLookup) extendable lookup

Note that these parameter values cannot be changed. Implementations wishing to use alternative values for these parameters must create their own version of this algorithm and add the customer algorithm to the batch control.

KWH Avoided Business Flags

The kWh Avoided Business Flag Upload (X1-KWHAV) batch process creates business flags for each metered service point in the "kWh avoided" files sent from an external analytics application. These business flags are created using the kWh Avoided Business Flag (X1-KWhAvdBusinessFlag) business object.

"kWh avoided" business flags include the following:

- **Business Flag Type**: The business flag type on which the business flag is based. Business flag types specify details for business flags of a certain type.
- **Status**: The current status of the business flag
- **Metered Service Point ID**: The metered service point associated with the business flag
- **Business Flag Date/Time**: The date and time the business flag was created. Note that this is not related to the time period of the interval data values contained in the record.
- **Actual Start Date/Time**: The start date and time of the event
- **Actual End Date/Time**: The end date and time of the event
- **Initiating System**: the system that created the business flag. When blank it indicates the business flag was created internally.
- **External Information**: Identifiers received from the system that initiated the business flag
  - **Event External Identifier**: External ID for the event
  - **Program External ID**: The program's external ID
  - **Source File ID**: An ID used to group business flags that were received in the same file
  - **External Reference System**: an alternate reference for the external system. This is used to identify the initiating system.
  - **External Service Point ID**: the identifier for the service point in the initiating system
- **Details**: Details of the program event that resulted in the actual energy reduction, including:
  - **Program Settlement Processed**: A flag that indicates if an program settlement transaction has been created from this business flag. Note that this is only visible when editing the business flag.
  - **Program Event ID**: ID for the program event
  - **Program Event Category**: The event's category
  - **Status**: The status of the data import. Can be one of the following:
    - **Success**: The import contained a Total KWh value and all interval values
    - **Partial**: The import contained a Total kWh value and some interval values
    - **Failure**: The import contained a Total kWh value but no interval values
  - **Total kWh**: The service point's total energy (in kWh) for the event.

Copyright © 2026, Oracle and/or its affiliates.

- **Total kWh avoided**: The service point's total energy reduction (in kWh) for the event. If the Status is "Success" this value should equal the total of the interval values listed below.
- **Interval Size**: The interval size of the interval values included, represented as hh:mm:ss (hours:minutes:seconds)
- **Interval Data**: A table listing the individual interval values that comprise the energy reduction. Each interval value is defined by:
  - **Sequence**: The sequence number for the interval value. The number of interval values should equal the duration of the event divided by the interval size. For example, for an event that last 2 hours with an interval size of 15 minutes, there would be 8 interval values.
  - **Date/Time**: The date and time of the interval value. Note that the first interval value should have a **Date/Time** equal to the **Actual Start Date/Time**.
  - **Quantity**: The interval value, in kWh. The total of these values should equal the **Total kWh avoided** value above.
  - **Interval Condition Value**: The usage transaction service quantity condition assigned to the value (from the **Interval Data Condition Value** parameter on the Create kWh Avoided Business Flag (X1-KWHAVDBUS) algorithm).

KWH Avoided Business Flag Processing

When "kWh avoided" business flags enter the "Additional Processing" state, the Program Settlement Creation (X1-EVTSETCR) algorithm creates a program settlement transaction based on the details contained in the business flag. If the Interval Size (Seconds) on the business flag is different from the Interval Size defined on the business flag type, the interval values are converted to match the interval size defined on the business flag type.

For example, a business flag contains the following 15-minute interval values (Interval Size of 900 seconds):

Date/Time	Value
2022-12-13T14:00:00	5.00
2022-12-13T14:15:00	3.00
2022-12-13T14:30:00	2.00
2022-12-13T14:45:00	2.00
2022-12-13T15:00:00	3.00
2022-12-13T15:15:00	8.00
2022-12-13T15:30:00	6.00
2022-12-13T15:45:00	4.00

If the Interval Size defined on the business flag type is 1-hour intervals (Interval Size of 01:00:00), the 15-minute interval values would be summed to create the following 1-hour interval values.

Date/Time	Value
2022-12-13T14:00:00	12 (the sum of the first four 15-minute interval values)
2022-12-13T15:00:00	21 (the sum of the second four 15-minute interval values)

KWH Avoided Business Flag Types

Copyright © 2026, Oracle and/or its affiliates.

You must create a business flag type that references the business flag Standard Name and business flag business object to be used when creating business flags, as well as the unit of measure, time of use period, and service quantity identifier combination used to define the interval usage values on program settlement transactions created from business flags. This should be creating using the kWh Avoided Business Flag Type (X1-KWhAvdBusinessFlagType) business object, and should contain the following:

- **Business Flag Type**: A code for the business flag type.
- **Standard Name**: The business flag standard name for business flags based on this type. This should be kWh Avoided (X1-KWHAVOIDED).
- **Related Transaction BO**: The business object used for business flags based on this type. This defaults to kWh Avoided Business Flag (X1-KWhAvdBusinessFlag) and should not be changed unless your implementation has created its own "kWh avoided" business flag business object.
- **Default Priority**: The default priority assigned to business flags based on this type. This is used to prioritize processing of business flags based on this type relative to other business flags.
- **Status**: The status of the business flag type. Business flags are created only from "Active" business flag types.
- **Description**: A description of the business flag type. This is used in information strings for the business flag type.
- **UOM/TOU/SQI**: Parameters that define how the "kWh avoided" interval values will be represented on program settlement transactions created from business flags of this type
  - **UOM**: The unit of measure used to define how the "kWh avoided" interval values will be represented on program settlement transactions
  - **TOU**: The time of use period used to define how the "kWh avoided" interval values will be represented on program settlement transactions
  - **SQI**: The service quantity identifier used to define how the "kWh avoided" interval values will be represented on program settlement transactions
  - **Interval Size**: The size of the "kWh avoided" interval values created on program settlement transactions, represented as hours:minutes:seconds (HH:MI:SS). This should match the interval size used for interval price sets.

See Understanding Business Flags for more information about business flags and business flag types.

Processing KW Drop Files

An external analytics application sends "kW drop" files containing the demand reductions achieved for the duration of an event for each customer, which can be used to determine financial settlement data. These files consist of comma separated records for each participating metered service point which are sent to Oracle Cloud Object Storage. A batch process imports these files and creates business flags that capture the data for each metered service point. This section describes how the application processes these "kW drop" files.

KW Drop File Format

"kW drop" files use the following comma values format:

Field	Description	Example
EventId	ID of the program event	5011

Copyright © 2026, Oracle and/or its affiliates.

Field	Description	Example
ProgramId	The Program ID for the event	1234512345
SPId	The Metered Service Point ID	1122334455
ActualStartTime	Start time of the event	2022-12-13T14:30:00-05:00
ActualEndTime	End time of the event	2022-12-13T16:30:00-05:00
IntervalSize(Seconds)	The interval size of the interval values included, represented in seconds.	900
MaxKwDrop	The maximum kW drop value of the KwDrop values.	20.00
KwDrop<n>	Individual interval values that indicate the kW drop values during the event period. The number of interval values is based on the duration of the event divided by the interval size.	1: 5.00 2: 3.00 3: 20.00 4: 2.00 5: 6: 8.00 7: 8:

Below is a sample file, based on the example data above:

```
EventId,ProgramId,SPId,ActualStartTime,ActualEndTime,IntervalSize(Seconds),MaxKwDrop,KwDrop1,KwDrop2,KwDrop3,KwDrop4,.....
5011,1234512345,1122334455,2022-12-13T14:30:00-05:00,2022-12-13T16:30:00-05:00,900,20.00,5.00,3.00,20.00,2.00,,8.00,,
```

Batch Processing

"kW drop" files are processed using the kW Drop Business Flag Upload (X1-KWDRP) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be scheduled to run on a regular basis.

This batch processes uses the Create kW Drop Business Flag (X1-KWDROPBUS) algorithm (defined for the "File Upload" **System Event** on the **Algorithms** tab of the **Batch Control** portal). This algorithm uses the following parameters:

- **Business Flag Standard Name**: The Standard Name (X1-KWDROP) used to identify the Business Flag Type for business flags created by the batch process. See Understanding Business Flags and Standard Name for more information about business flag standard names.
- **Interval Data Condition Value**: The usage transaction service quantity condition code for the interval data values captured in the business flag, from the Usage Transaction SQ Interval Condition (D2-UTSQIntvlConditionLookup) extendable lookup

Note that these parameter values cannot be changed. Implementations wishing to use alternative values for these parameters must create their own version of this algorithm and add the customer algorithm to the batch control.

KW Drop Business Flags

The kW Drop Business Flag Upload (X1-KWDRP) batch process creates business flags for each metered service point in the "kW drop" files sent from an external analytics application. These business flags are created using the KW Drop Business Flag (X1-KWBusinessFlag) business object.

Copyright © 2026, Oracle and/or its affiliates.

"kW drop" business flags include the following:

- **Business Flag Type**: The business flag type on which the business flag is based. Business flag types specify details for business flags of a certain type.
- **Status**: The current status of the business flag
- **Metered Service Point ID**: The metered service point associated with the business flag
- **Business Flag Date/Time**: The date and time the business flag was created. Note that this is not related to the time period of the interval data values contained in the record.
- **Actual Start Date/Time**: The start date and time of the event
- **Actual End Date/Time**: The end date and time of the event
- **Initiating System**: the system that created the business flag. When blank it indicates the business flag was created internally.
- **External Information**: Identifiers received from the system that initiated the business flag
  - **Event External Identifier**: External ID for the event
  - **Program External ID**: The program's external ID
  - **Source File ID**: An ID used to group business flags that were received in the same file
  - **External Reference System**: an alternate reference for the external system. This is used to identify the initiating system.
  - **External Service Point ID**: the identifier for the service point in the initiating system
- **Details**: Details of the program event that resulted in the actual energy reduction, including:
  - **Program Settlement Processed**: A flag that indicates if an program settlement transaction has been created from this business flag. Note that this is only visible when editing the business flag.
  - **Program Event ID**: ID for the program event
  - **Program Event Category**: The event's category
  - **Status**: The status of the data import. Can be one of the following:
    - **Success**: The import contained a Total KWh value and all interval values
    - **Partial**: The import contained a Total kWh value and some interval values
    - **Failure**: The import contained a Total kWh value but no interval values
  - **Max kW Drop**: The maximum drop (in kW) for the event. This equals the maximum **Quantity** value from the values in the **Interval Data** section.
  - **Interval Size**: The interval size of the interval values included, represented as hh:mm:ss (hours:minutes:seconds)
  - **Interval Data**: A table listing the individual interval values that represent the demand reduction at each interval. Each interval value is defined by:
    - **Sequence**: The sequence number for the interval value. The number of interval values should equal the duration of the event divided by the interval size. For example, for an event that last 2 hours with an interval size of 15 minutes, there would be 8 interval values.
    - **Date/Time**: The date and time of the interval value. Note that the first interval value should have a **Date/Time** equal to the **Actual Start Date/Time**.

Copyright © 2026, Oracle and/or its affiliates.

    * **Quantity**: The interval value, in kW. The maximum of these values should equal the **Max kW Drop** value above.
    * **Interval Condition Value**: The usage transaction service quantity condition assigned to the value (from the **Interval Data Condition Value** parameter on the Create kWh Avoided Business Flag (X1-KWHAVDBUS) algorithm).

KW Drop Business Flag Processing

When "kW drop" business flags enter the "Additional Processing" state, the KW Drop Program Settlement Creation (X1-KWEVTSTCR) algorithm creates a program settlement transaction based on the details contained in the business flag. If the Interval Size (Seconds) on the business flag is different from the Interval Size defined on the business flag type, the interval values are converted to match the interval size defined on the business flag type.

KW Drop Business Flag Types

You must create a business flag type that references the business flag Standard Name and business flag business object to be used when creating business flags, as well as the unit of measure, time of use period, and service quantity identifier combination used to define the interval usage values on program settlement transactions created from business flags. This should be creating using the KW Drop Business Flag Type (X1-KWDropBusinessFlagType) business object, and should contain the following:

- **Business Flag Type**: A code for the business flag type.
- **Standard Name**: The business flag standard name for business flags based on this type. This should be kW Drop (X1-KWDROP).
- **Related Transaction BO**: The business object used for business flags based on this type. This defaults to KW Drop Business Flag (X1-KWBusinessFlag) and should not be changed unless your implementation has created its own "kW drop" business flag business object.
- **Default Priority**: The default priority assigned to business flags based on this type. This is used to prioritize processing of business flags based on this type relative to other business flags.
- **Status**: The status of the business flag type. Business flags are created only from "Active" business flag types.
- **Description**: A description of the business flag type. This is used in information strings for the business flag type.
- **UOM/TOU/SQI**: Parameters that define how the "kW drop" interval values will be represented on program settlement transactions created from business flags of this type
  - **UOM**: The unit of measure used to define how the "kW drop" interval values will be represented on program settlement transactions
  - **TOU**: The time of use period used to define how the "kW drop" interval values will be represented on program settlement transactions
  - **SQI**: The service quantity identifier used to define how the "kW drop" interval values will be represented on program settlement transactions
  - **Interval Size**: The size of the "kW drop" interval values created on program settlement transactions, represented as hours:minutes:seconds (HH:MI:SS).

See Understanding Business Flags for more information about business flags and business flag types.

Importing Data from External Systems

Copyright © 2026, Oracle and/or its affiliates.

This section describes how data from external systems can be imported. This includes customer-related data such as contacts, metered service points, and metered services, as well as device and enrollment data including controllable devices, device locations, enrollments, and event participation records.

- Importing Customer-Related Data from a Customer Information System
- Importing Device and Program Enrollment Data from a Demand Response Management System (DRMS)
- Importing Facility and Network Location Data from a Distributed Energy Resource System

Importing Customer-Related Data from a Customer Information System

Customer-related data includes contacts, metered service points, and metered services, and is usually imported from a legacy customer information system. Data to be imported is sent to a tab-delimited flat file, which is uploaded into the system via batch processing which creates inbound data synchronization (sync) requests. These sync requests are processed and validated by a monitor batch process, and then a separate batch process creates master data records from each.

Import Process Overview

This section provide an overview of the import process and outlines the various system objects used for each type of data (contacts, metered service points, and metered services). The overall process is as follows:

1. Data to be imported is exported from the CIS into a tab-delimited values file.
2. An "Upload" batch process is used to create inbound sync requests in the Pending state based on object-specific initial synchronization business objects. A separate inbound sync request is created for each row in the import file.
3. The inbound sync requests are processed using the Sync Request Monitor (F1-SYNRQ) batch process.
4. An "Initial Sync Request - Load" batch process is used to create individual master data records from each inbound sync request.

Initial Load vs. Incremental Uploads

Customer data can be uploaded in one of two modes: Initial Load or Incremental. The upload mode is defined by the Initial Load Sync Flag value in the upload file.

Initial Load uploads are used when setting up the system to import contact, metered service point, and metered service data from a customer information system such as Oracle Utilities Customer to Meter or Oracle Utilities Customer Cloud Service.

Incremental uploads are used to update customer data as customer data changes in the customer information system. Incremental uploads do NOT use "Import Sync Request - Load" batch processes. Records are added or updated automatically once validated.

Upload Batch Processes

Import files are uploaded using object-specific "Upload" batch processes. See Import Process Object Summary below for the specific "Upload" batch process used for each type of customer data. These should be run as needed for the initial import, as well as subsequent incremental updates.

These processes use the following parameters:

Copyright © 2026, Oracle and/or its affiliates.

- **File Name**: The name of the file to upload. File names can use wildcards (Required)
- **Flat Path**: The file location where the upload file is located. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* for more information. (Required)
- **Processed File Extension**: The text appended to the file name after the file has been processed. Default is "PROCESSED".
- **Error On No Files Found**: Indicates whether or not an error is issued if no files are found matching the given name in the given path. Valid values are "true" (default) and "false".
- **External System**: The external system that created the upload file (Required)

Use the **Batch Control** portal to learn more about these and other parameters used by these batch processes.

Inbound Sync Request Business Objects

Inbound sync requests are created by the "Upload" batch processes, and are based on "Initial Sync Request" business objects. See Import Process Object Summary below for the specific "Initial Sync Request" business objects used for each type of customer data.

The table below outlines the stages of the inbound sync request lifecycle. Use the **Business Object** portal to learn more about these business objects and the details of each lifecycle state.

State	Notes
Pending	Inbound sync requests begin in this state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition sync requests in this state to the Transform/Validate state.
Transform/Validate	In this state the data is validated and transformed as needed for import. In this state values defined by lookups are derived from the data in the upload file via extendable lookups. Sync requests move from this state to either the Validated or Add/Update states, depending on the type of upload. See Initial Load vs. Incremental Uploads for more information.
Validation Error	Sync requests enter this state if an error occurs during validation in the Transform/Validate step. If a sync request needs to be updated manually, it can be put into this state via the **Correct** action in the **Record Actions** section.
Validated	Initial Load sync requests enter this state following validation in the Transform/Validate step. If corrections need to be made, users can use the **Correct** action in the **Record Actions** section to transition the record to the Validation Error state.
Add/Update	Incremental sync requests enter this state following validation in the Transform/Validate step, and records are either added or updated as appropriate.
Loaded	Initial Load sync requests enter this state when a record has been successfully loaded,

Copyright © 2026, Oracle and/or its affiliates.

State	Notes
Synchronized	Incremental sync requests enter this state when a record has been successfully added or updated. The Create Acknowledgement (X1-CRESRACK) algorithm, defined as an Enter algorithm on this state, can be configured to send an acknowledgement to the system from which the imported data originated. This requires defining the **Outbound Message Type** parameter on the algorithm. The base package includes the Send Sync Request Acknowledgement (X1-SYNREQACK) outbound message type which can be used for this.
Discarded	The sync request enters this state if the sync request is discarded from either the Validated or Validation Error state,

Initial Sync Request Load Batch Processes

Master data records are created from inbound sync requests using "Initial Sync Request - Load" batch processes. See below for the specific "Initial Sync Request - Load" batch processes used for each type of customer data. These should be run as needed for the initial import, as well as subsequent incremental updates.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Import Process Object Summary

The table below lists the batch controls and business objects used by each type of customer data.

Import Order	Object Type	Upload Batch Process	Inbound Sync Request Business Object	Initial Sync Request Load Batch Process
1	Contact	Contact File Upload (X1-COUPL)	Contact Initial Sync Request (X1-DASyncReqContact)	Initial Sync Request - Load Contacts (X1-SILCO)
2	Metered Service Point	Metered Service Point File Upload (X1-MSPUP)	Metered Service Point Sync Request BO (X1-SyncReqMeteredSP)	Initial Sync Request - Load Metered SP (X1-SLMSP)
3	Metered Service	Metered Service File Upload (X1-MSUPL)	Metered Service Inbound Sync Request (X1-DASyncReqMeteredService)	Initial Sync Request - Load Metered Service (X1-SILMS)

> **Note**
>
> Customer data **must** be imported in the order listed above.

- Importing Contacts
- Importing Metered Service Points
- Importing Metered Services

Copyright © 2026, Oracle and/or its affiliates.

Importing Contacts

This topic provides additional details related to importing contacts from a customer information system for use with Customer Program Management.

Contact Import File Format

Contact import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Reference Id	Optional	The contact's reference ID in the CIS application
Change Date Time	Optional	The date and time on which the contact was changed. Applicable only for Incremental Uploads Will be used to sequence processing in case there are multiple changes to the same contact.
Initial Load Sync Flag	Required	A value that indicates the type of upload. Valid values are: • F1IL - Initial Load • F1IC - Incremental
External Id	Required	The contact's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Contact External ID.
Contact Type	Required	The type of contact. When importing data from Customer to Meter or Customer Cloud Service this will be either Person or Business. If the provided contact type has not been defined as a Program Contact Type, an extendable lookup is used to map this value to the Contact Type used.
Name	Required	The contact's name. For format for this is <FIRST_NAME> <LAST_NAME> (note the space between them.
Home Phone	Optional	A home phone number associated with person-type contacts.
Mobile Phone	Optional	A mobile phone number associated with person-type contacts.
Phone	Optional	A phone number associated with business-type contacts.
Email	Optional	An email address for the contact.
Life Support Sensitive Load	Required	A flag that indicates if the contact requires a Life Support Sensitive Load. Valid values are "true" and "false" An extendable lookup is used to map this value to the Life Support Sensitive Load flag used in Customer Program Management.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Life Support Sensitive Load Description	Optional	An optional description for Life Support Sensitive Load.
Is Erased	Optional	A flag that indicates if the contact has been obfuscated (if supported) in the CIS application. Valid values are Y/N (Default) Applicable only to Incremental Uploads This should be set to "N" if Object Erasure is not supported in the CIS application.

Sample Record

Below is a sample file:

```
EXTREFCON11    2023-06-21T13:15:00-08:00    F1IC    EXTCONTACT1    P    John
Smith    (510)777-8888    (510)777-9090    (510)556-4556
johnsmith@gmail.com    false    None    N
```

Upload Batch Process

Contact import files are uploaded using the Contact File Upload (X1-COUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process creates inbound sync requests in the "Pending" state (see below).

The Contact File Upload (X1CONTUPLD) algorithm used by this batch process uses the following extendable lookups to map data in the import file to values defined in Customer Program Management:

- CCB Contact Type Mapping (D1-CCBContactTypeLookup)
- CCB Life Support Sensitive Load Mapping (D1-CCBLifeSupSensitiveLdLookup)

Values must be defined for these extendable lookups before import files can be uploaded and used with Customer Program Management.

Inbound Sync Request Business Object

Contact inbound sync requests are based on the Contact Initial Sync Request (X1-DASyncReqContact) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import. What happens next is based on the type of sync request:

- Initial Load sync requests then move to the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below) .
- Incremental sync requests move to the "Add/Update" state in which algorithms add or update records as appropriate and then to the "Synchronized" state.
- The Create Acknowledgement (X1-CRESRACK) algorithm on the "Synchronized" state can be configured to send an acknowledgement to the system from which the imported data originated. This requires specifying the "Send Sync Request Acknowledgement" (X1-

Copyright © 2026, Oracle and/or its affiliates.

SYNREQACK) outbound message type for the **Outbound Message Type** parameter on the algorithm.

Initial Sync Request Load Batch Process

Contacts imported as part of an Initial Load upload are created from Initial Load inbound sync requests using the Initial Sync Request - Load Contacts (X1-SILCO) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process transitions the inbound sync requests to the "Loaded" state.

Importing Metered Service Points

This topic provides additional details related to importing metered service points from a customer information system.

Metered Service Point Import File Format

Metered service point import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Reference Id	Optional	The metered service point's reference ID in the CIS application.
Change Date Time	Optional	The date and time on which the metered service point was changed. Applicable only for Incremental Uploads Will be used to sequence processing in case there are multiple changes to the same metered service point.
Initial Load Sync Flag	Required	A value that indicates the type of upload. Valid values are: • F1IL - Initial Load • F1IC - Incremental
External Id	Required	The metered service point's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Service Point External ID.
Metered SP Type	Required	The type of metered service point. When importing data from Customer to Meter or Customer Cloud Service this comes from the Service Point Type. If the provided metered service point type has not been defined as a Metered Service Point Type an extendable lookup is used to map this value to the Metered Service Point Type used with Customer Program Management.
SP Source Status	Required	The metered service point's source status. Valid values are values for the Source Status (SP_SRC_STAT_FLG) lookup, and include: • D1CN (Connected) • D1DC (Disconnected)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Division	Optional	The metered service point's division. If the provided division has not been defined as a Division, an extendable lookup is used to map this value to the Division used with Customer Program Management.
External Premise Id	Optional	The ID of the premise associated with the metered service point in the CIS application.
Time Zone	Required	The time zone in which the metered service point is located.
Country	Optional	The country in which the metered service point is located.
Postal	Optional	The metered service point's postal code.
Address 1	Optional	The metered service point's address.
Address 2	Optional	Additional details regarding the metered service point's address (if applicable).
Address 3	Optional	Additional details regarding the metered service point's address (if applicable).
Address 4	Optional	Additional details regarding the metered service point's address (if applicable).
City	Optional	The city in which the metered service point is located.
County	Optional	The county in which the metered service point is located.
State	Optional	The state in which the metered service point is located.
Longitude	Optional	The metered service point's longitude.
Latitude	Optional	The metered service point's latitude.
Life Support Sensitive Load Flag	Required	A flag that indicates if the metered service point requires a Life Support Sensitive Load. Valid values are "true" and "false" An extendable lookup is used to map this value to the Life Support Sensitive Load flag used with Customer Program Management.
Life Support Sensitive Load Description	Optional	An optional description for Life Support Sensitive Load.
Status	Required	The current state of the metered service point. Valid values are "ACTIVE" and "INACTIVE".

Sample Record

Below is a sample file:

```
EXTREFMSP11    2023-06-21T14:01:00-08:00    F1IC    EXTMSP1    ZZ-
EXTMSPTYPE    D1CN    ZZEDV    EXTPREM1    USPACIFIC    USA    94705    7610
Block Derby St    Addr2    Addr3    Addr4    Berkley    Alameda    CA
-81.697494    41.501120    false        ACTIVE
```


Copyright © 2026, Oracle and/or its affiliates.

Upload Batch Process

Metered service point import files are uploaded using the Metered Service Point File Upload (X1-MSPUP) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process creates inbound sync requests in the "Pending" state (see below).

The Metered Service Point File Upload (X1MSPUPL) algorithm used by this batch process uses the following extendable lookups to map data in the import file to values used with Customer Program Management:

- CCB Service Point Type Mapping (D1-CCBSPTypeLookup)
- CCB Life Support Sensitive Load Mapping (D1-CCBLifeSupSensitiveLdLookup)
- CIS Division Mapping (D1-CCBDivisionLookup)

Values must be defined for these extendable lookups before import files can be uploaded.

Inbound Sync Request Business Object

Metered service point inbound sync requests are based on the Metered Service Point Sync Request BO (X1-SyncReqMeteredSP) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import. What happens next is based on the type of sync request:

- Initial Load sync requests then move to the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below) .
- Incremental sync requests move to the "Add/Update" state in which algorithms add or update records as appropriate and then to the "Synchronized" state.
- The Create Acknowledgement (X1-CRESRACK) algorithm on the "Synchronized" state can be configured to send an acknowledgement to the system from which the imported data originated. This requires specifying the "Send Sync Request Acknowledgement" (X1-SYNREQACK) outbound message type for the **Outbound Message Type** parameter on the algorithm.

Initial Sync Request Load Batch Process

Metered service points imported as part of an Initial Load upload are created from Initial Load inbound sync requests using the Initial Sync Request - Load Metered SP (X1-SLMSP) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process transitions the inbound sync requests to the "Loaded" state.

Importing Metered Services

This topic provides additional details related to importing metered services from a customer information system.

Copyright © 2026, Oracle and/or its affiliates.

Metered Services Import File Format

Metered service import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
External Reference Id	Optional	The metered service's reference ID in the CIS application.
Change Date Time	Optional	The date and time on which the metered service was changed. Applicable only for Incremental Uploads Will be used to sequence processing in case there are multiple changes to the same metered service point.
Initial Load Sync Flag	Required	A value that indicates the type of upload. Valid values are: • F1IL - Initial Load • F1IC - Incremental
External Id	Required	The metered service's ID in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Service Agreement External ID.
Metered Service Type	Required	The type of metered service. When importing data from Customer to Meter or Customer Cloud Service this comes from the Service Agreement Type. If the provided metered service point type has not been defined as a Metered Service Type, an extendable lookup is used to map this value to the Metered Service Type used with Customer Program Management.
External Contact ID	Optional	The ID of the contact associated with the metered service in the CIS application. When importing data from Customer to Meter or Customer Cloud Service this is the Contact External ID.
External Account ID	Optional	An ID used to identify the account associated with the metered service record in the CIS application.
Start Date Time	Required	The date and time on which the metered service record started.
End Date Time	Optional	The date and time on which the metered service record ended or will end (if applicable).
Division	Optional	The metered service's division. If the provided division has not been defined as a Division, an extendable lookup is used to map this value to the Division used with Customer Program Management.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
SA Status	Optional	The metered service's status. If the provided status has not been defined as an Status, an extendable lookup is used to map this value to a Status defined for Customer Program Management.
Rate Schedule List	Optional	A concatenated list of rate schedules and effective dates associated with the metered service. Each row in this list is composed of the following: • Rate Schedule • Effective Date Each element in the row will be delimited by comma (,). Each row in the list will be delimited by semi-colon (;). For example: ERES-1,2022-01-01ERES-2,2021-01-01
Metered SP List	Optional	A concatenated list of metered service points associated with the metered service. Each row in this list is composed of the following: • External Metered SP Id (Required) • Start Date Time (Required) • End Date Time (Optional) Each element in the row will be delimited by comma (,). Each row in the list will be delimited by semi-colon (;). Example: 6800401571,2022-12-13T14:30:00-05:00,1703304223,2022-12-13T14:30:00-05:00,
Is Erased	Optional	A flag that indicates if the customer has been obfuscated (if supported) in the CIS application. Valid values are Y/N (Default) Applicable only to Incremental Uploads This should be set to "N" if Object Erasure is not supported in the CIS application.

Sample Record

Below is a sample file:

```
EXTREFMS11    2023-06-15T13:02:00-08:00    F1IC    EXTMS1    CA!ZZMTRDST
EXTCONTACT1    EXTACCT1    2022-10-15T09:23:00-08:00
2023-06-14T13:45:00-08:00    ZZEDV    20    ERESRBOH,2022-11-09;
EXTMSP1,2022-11-09T13:14:55-08:00,2023-04-14T08:25:44-08:00;    Y
```

Upload Batch Process

Metered service import files are uploaded using the Metered Service File Upload (X1-MSUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch

Copyright © 2026, Oracle and/or its affiliates.

process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process creates inbound sync requests in the "Pending" state (see below).

The Metered Service File Upload (X1_LEGMSUPLD) algorithm used by this batch process uses the following extendable lookups to map data in the import file to values used with Customer Program Management:

- CCB SA Type Mapping (D2-CCBSATypeLookup)
- CCB SA Status Mapping (D2-CCBSAStatusLookup)
- CIS Division Mapping (D1-CCBDivisionLookup)

Values must be defined for these extendable lookups before import files can be uploaded.

Inbound Sync Request Business Object

Metered service inbound sync requests are based on the Metered Service Inbound Sync Request (X1-DASyncReqMeteredService) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import. What happens next is based on the type of sync request:

- Initial Load sync requests then move to the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below) .
- Incremental sync requests move to the "Add/Update" state in which algorithms add or update records as appropriate and then to the "Synchronized" state.
- The Create Acknowledgement (X1-CRESRACK) algorithm on the "Synchronized" state can be configured to send an acknowledgement to the system from which the imported data originated. This requires specifying the "Send Sync Request Acknowledgement" (X1-SYNREQACK) outbound message type for the **Outbound Message Type** parameter on the algorithm.

Initial Sync Request Load Batch Process

Metered services imported as part of an Initial Load upload are created from Initial Load inbound sync requests using the Initial Sync Request - Load Metered Service (X1-SILMS) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import, as well as subsequent incremental updates.

This process transitions the inbound sync requests to the "Loaded" state.

Importing Device and Program Enrollment Data from a Demand Response Management System (DRMS)

Device and program enrollment data includes controllable devices, device locations, enrollments, and event participation records, and is usually imported from a demand response management system (DRMS) such as Oracle Utilities Network Management System. Data to be imported is sent to a tab-delimited flat file, which is uploaded into the system via batch processing which creates inbound data synchronization (sync) requests. These sync requests are processed and validated by a monitor batch process, and then a separate batch process creates master data records from each.

Copyright © 2026, Oracle and/or its affiliates.

Device and program enrollment data uploads are used when setting up the system to import data from a demand response management system (DRMS) as Oracle Utilities Network Management System. Once the implementation is live, device and program enrollment information will created in in the application and sent to the demand response management system (DRMS),

> **Note**
>
> Device and program enrollment data cannot be imported until AFTER importing customer data (see Importing Customer-Related Data from a Customer Information System).

Import Process Overview

This section provide an overview of the import process and outlines the various system objects used for each type of data (controllable devices, device locations, enrollments, and event participation). The overall process is as follows:

1. Data to be imported is exported from the DRMS into a tab-delimited values file.
2. An "Upload" batch process is used to create inbound sync requests in the Pending state based on object-specific initial synchronization business objects. A separate inbound sync request is created for each row in the import file.
3. The inbound sync requests are processed using the Sync Request Monitor (F1-SYNRQ) batch process.
4. An "Initial Sync Request - Load" batch process is used to create individual master data records from each inbound sync request.

Import of some types of data uses additional batch processing. These are noted in the data-specific topics that follow.

Upload Batch Processes

Import files are uploaded using object-specific "Upload" batch processes. See Import Process Object Summary below for the specific "Upload" batch process used for each type of device and enrollment data. These should be run as needed for the initial import, as well as subsequent incremental updates.

These processes use the following parameters:

- **File Name**: The name of the file to upload. File names can use wildcards (Required)
- **Flat Path**: The file location where the upload file is located. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* for more information. (Required)
- **Processed File Extension**: The text appended to the file name after the file has been processed. Default is "PROCESSED".
- **Error On No Files Found**: Indicates whether or not an error is issued if no files are found matching the given name in the given path. Valid values are "true" (default) and "false".
- **External System**: The external system that created the upload file (Required)

Use the **Batch Control** portal to learn more about these and other parameters used by these batch process.

Copyright © 2026, Oracle and/or its affiliates.

Inbound Sync Request Business Objects

Inbound sync requests are created by the "Upload" batch processes, and are based on "Initial Sync Request" business objects. See Import Process Object Summary below for the specific "Initial Sync Request" business objects used for each type of customer data.

The table below outlines the stages of the inbound sync request lifecycle. Use the **Business Object** portal to learn more about these business objects and the details of each lifecycle state.

State	Notes
Pending	Inbound sync requests begin in this state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition sync requests in this state to the Transform/Validate state.
Transform/Validate	In this state the data is validated and transformed as needed for import. In this state values defined by lookups are derived from the data in the upload file via extendable lookups. Sync requests move from this state to either the Validated state.
Validation Error	Sync requests enter this state if an error occurs during validation in the Transform/Validate step. If a sync request needs to be updated manually, it can be put into this state via the **Correct** action in the **Record Actions** section.
Validated	Initial Load sync requests enter this state following validation in the Transform/Validate step. If corrections need to be made, users can use the **Correct** action in the **Record Actions** section to transition the record to the Validation Error state.
Loaded	Initial Load sync requests enter this state when a record has been successfully loaded,
Discarded	The sync request enters this state if the sync request is discarded from either the Validated or Validation Error state,

Initial Sync Request Load Batch Processes

Master data records are created from inbound sync requests using "Initial Sync Request - Load" batch processes. See Import Process Object Summary below for the specific "Initial Sync Request - Load" batch processes used for each type of device and enrollment data. These should be run as needed for the initial import, as well as subsequent incremental updates.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Import Process Object Summary

The table below lists the batch controls and business objects used by each type of device and enrollment data.

Copyright © 2026, Oracle and/or its affiliates.

Import Order	Object Type	Upload Batch Process	Initial Synchronization Business Object	Initial Synchronization Load Batch Process
1	Controllable Device	Legacy Controllable Device File Upload (X1-DVUPL)	Controllable Device Initial Sync Request (X1-DASyncReqControllableDevice)	Initial Sync Request - Load Controllable Device (X1-SILDV)
2	Device Location	Legacy Device Location File Upload (X1-DLUPL)	Device Location Initial Sync Request (X1-DASyncReqDeviceLocation)	Initial Sync Request - Load Device Location (X1-SILDL)
3	Enrollment	Legacy Enrollment File Upload (X1-ENUPL)	Enrollment Initial Sync Request (X1-DASyncReqEnrollment)	Initial Sync Request - Load Enrollment (X1-SILEN)
4	Event Participation	Legacy Participation Event File Upload (X1-PTUPL)	Event Participation Sync Request BO (X1-DASyncReqEventParticipation)	Initial Sync Request - Load Event Participation (X1-SLPAR)

> **Note**
>
> Device and program enrollment data **must** be imported in the order listed above.

- Importing Controllable Devices
- Importing Device Locations
- Importing Enrollments
- Importing Event Participation Records

Importing Controllable Devices

This topic provides additional details related to importing legacy controllable devices from a demand response management system (DRMS).

Controllable Device Import File Format

Legacy controllable device import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
Legacy Device Id	Required	An id for the controllable device in the DRMS application.
Device Type	Required	The controllable device's type. This must be a valid controllable device type. See Understanding Controllable Asset Types for more information.
Serial Number	Optional	The controllable device's serial number.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Badge Number	Optional	The controllable device's badge number.
NIC ID	Required	The controllable device's Network Identification Card (NIC) ID.
Head End Registration	Optional	A flag that indicates if the controllable device is registered in the head end system. Valid values are values for the Head-End Registration Status (HEAD_END_REGISTR_STATUS_FLG) lookup, and include: • D1RE - Registered • D1NR - Not Registered • D1DR - Deregistered
Legacy Device Date Time	Required	The controllable device's asset disposition date and time. This will be one of the following: • In-Store Date Time: Date and time when the asset came into the storeroom/ warehouse or when asset was removed from the location. • Retire Date Time: Date and time when the asset was expired. • Repair Date Time: Date and time when the asset came into the repair shop
Specification	Required	The specification associated with the controllable device. This must be a valid specification.
Asset Ownership	Optional	A flag that indicates the controllable device's ownership. Valid values are values for the Asset Ownership (ASSET_OWNERSHIP_FLG) lookup, and include: • W1CO - Customer Owned • W1OW - Utility Owned
Asset Status	Required	A value that indicates the current status of the controllable device. Valid values include: • INSTORE • INREPAIR • RETIRED

Sample Record

Below is a sample file:

```
EXTDVC3    ZZ-MUYU    SERIALNBR3    BADGENBR3    NICID3    D1NR
2023-04-23T12:22:00-05:00    ZZ-MUYU-CD-SPEC    W1CO    INSTORE
```

Upload Batch Process

Legacy controllable device import files are uploaded using the Legacy Controllable Device File Upload (X1-DVUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

Copyright © 2026, Oracle and/or its affiliates.

This process creates inbound sync requests in the "Pending" state (see below).

Inbound Sync Request Business Object

Controllable device inbound sync requests are based on the Controllable Device Initial Sync Request (X1-DASyncReqControllableDevice) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import and moved into the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below).

Initial Sync Request Load Batch Process

Legacy controllable devices are created from inbound sync requests using the Initial Sync Request - Load Controllable Device (X1-SILDV) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process transitions the inbound sync requests to the "Loaded" state.

Create Controllable Asset from Controllable Device Batch Process

The Create Controllable Asset from Controllable Device (X1-CRAST) batch control is used to create controllable assets from newly uploaded controllable devices. This process should be run as needed for the initial import.

This process uses the following algorithms:

- Select Records - Controllable Devices without an asset id (X1-SLNOASSET): This algorithm selects controllable devices where the legacy device ID is populated, but the Asset ID is empty.
- Process Record - Create Controllable Asset (X1CREATASSET): This algorithm creates controllable assets from data in the Controllable Device Initial Sync Request (X1-DASyncReqControllableDevice) inbound sync requests created for each controllable device.

This process uses the following parameters:

- **Default Out of Service Location**: The fallback out of service location used when creating controllable assets if other state-specific out of service locations are not defined. (Required) Note that controllable asset types must be configured with the appropriate device location types for each these states. See Understanding Controllable Asset Types for more information.
- **In Store Out of Service Location**: The out of service location when creating controllable assets in the "In Store" state.
- **InRepair Out of Service Location**: The out of service location when creating the controllable assets in the "In Repair" state.
- **Retired Out of Service Location**: The out of service location when creating the controllable assets in the "In Store", "In Repair" or "Retired" state.

Use the **Batch Control** portal to learn more about these and other parameters used by these batch process.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Because a controllable asset's disposition includes a device location, and device locations are imported after importing controllable devices, controlled devices that currently "Installed" are imported in the "In Store" state. Their status will be corrected when the device locations are imported.

Importing Device Locations

This topic provides additional details related to importing legacy device locations from a demand response management system (DRMS).

Device Location Import File Format

Legacy device location import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
Legacy Device Location Id	Required	The id for the device location in the DRMS application.
Device Location Type	Required	The device location's type. This must be a valid device location type. See Understanding Device Location Types for more information.
CIS Metered Service Point	Required	The id for the metered service point associated with the device location in the CIS application.
Ok To Enter	Optional	A value that indicates if it is okay for field service crews to enter the device location. Valid values are values defined for the Ok to Enter (D1-OkToEnterLookup) extendable lookup.
SP Warning	Optional	A value that indicates any warnings for field service crews when at the device location (such as a dog). Valid values are values defined for the SP Warning (D1-SPWarningLookup) extendable lookup.
SP Instruction	Optional	A value that provides any special instructions for field service crews when at the device location. Valid values are values defined for the SP Instruction (D1-SPInstructionLookup) extendable lookup.
SP Instruction Details	Optional	A text string containing any additional details related to the SP Instruction value.
Key	Optional	A value that indicates if a key is available for field service crews when at the device location. Valid values are values defined for the Key (D1-KeyLookup) extendable lookup.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Key Id	Optional	A text string containing any additional details related to a key.
Device Location	Optional	A value that indicates the location of the controllable device at the device location. Valid values are values defined for the Device Location (D1-DeviceLocationLookup) extendable lookup.
Device Location Details	Optional	A text string containing any additional details related to the device location.
Device List	Optional	A concatenated list of controllable devices current installed at the device location. Each row in this list is composed of the following: • Legacy Device Id (Required) • Attach Date Time (Required) • Remove Date Time (Optional) • Active/Inactive Indicator (D1YS/D1NO) (Required) Each element in the row will be delimited by comma (,). Each row in the list will be delimited by semi-colon (;). Example: DVC1,2022-12-13T14:30:00-05:00,,D1YS DVC2,,2022-12-13T14:30:00-05:00,D1YS ;

Sample Record

Below is a sample file:

```
EXTDVCLOC1    ZZ-MUYU    EXTMSP1    ZZNO    DOG    C    service point
instruction details    ZZ1    key1    LEFT    DeviceLoc1 details    ACTIVE
EXTDVC1,2022-11-09T09:23:00-05:00,2023-04-15T14:04:00-05:00,D1YS;
```

Upload Batch Process

Legacy device location import files are uploaded using the Legacy Device Location File Upload (X1-DLUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process creates inbound sync requests in the "Pending" state (see below).

The Legacy Device Location File Upload (X1LGDVLOCUPL) algorithm used by this batch process uses the following extendable lookups to map data in the import file to values used with Customer Program Management:

- Ok to Enter (D1-OkToEnterLookup)
- SP Warning (D1-SPWarningLookup)
- SP Instruction (D1-SPInstructionLookup)

Copyright © 2026, Oracle and/or its affiliates.

- Key (D1-KeyLookup)
- Device Location (D1-DeviceLocationLookup)

Values must be defined for these extendable lookups before device location import files can be uploaded.

Inbound Sync Request Business Object

Controllable device inbound sync requests are based on the Device Location Initial Sync Request (X1-DASyncReqDeviceLocation) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import and moved into the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below).

Initial Sync Request Load Batch Process

Legacy controllable devices are created from inbound sync requests using the Initial Sync Request - Load Device Location (X1-SILDL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process transitions the inbound sync requests to the "Loaded" state.

Create Asset Locations from Device Location Batch Process

The Create Asset Location from Device Location (X1-CRASL) batch control is used to create asset locations from newly uploaded device locations.

This process uses the following algorithms:

- Select Records - Device locations without an asset location (node) (X1SLNOASTLOC): This algorithm selects device locations where the legacy device location ID is populated, but the Asset service point ID is empty.
- Process Record - Create Asset Location (Node) (X1CRASSETLOC): This algorithm creates asset locations from data in the Device Location Initial Sync Request (X1-DASyncReqDeviceLocation) inbound sync requests created for each device location.

This process uses the following parameters:

- **Default Out of Service Location**: The out of service location used when creating controllable assets if state-specific out of service locations are not defined. (Required)

Use the **Batch Control** portal to learn more about these and other parameters used by these batch process.

Importing Enrollments

This topic provides additional details related to importing legacy program enrollments from a demand response management system (DRMS). Program enrollments are defined by program subscriptions in Customer Program Management.

Enrollment Import File Format

Legacy enrollment import files use the following tab-separated values format:

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Legacy Enrollment Id	Required	The id for the enrollment in the DRMS application.
CIS Metered Service	Required	The id for the metered service point associated with the enrollment in the CIS application.
Program Subscription Type	Required	The program subscription type. This must be a valid program subscription type. See Understanding Program Subscription Types for more information.
Start Date Time	Required	The date and time on which the program subscription record started.
End Date Time	Optional	The date and time on which the program subscription record ended or will end (if applicable).
Legacy Device Location Id	Required	The id for the device location in the DRMS application.
Legacy Program External Id	Required	The id for the program in the DRMS application.
Status	Required	The current status of the enrollment/ program subscription. Valid values include: • ACTIVE • INACTIVE

Sample Record

Below is a sample file:

```
EXTENR2     EXTMS2     ZZ_MZ_TEST     2022-10-24T08:00:00-05:00
2024-10-24T08:00:00-05:00     EXTDVCLOC2     LEGACYPRG1     ACTIVE
```

Upload Batch Process

Legacy enrollment import files are uploaded using the Legacy Enrollment File Upload (X1-ENUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process creates inbound sync requests in the "Pending" state (see below).

Inbound Sync Request Business Object

Enrollment inbound sync requests are based on the Enrollment Initial Sync Request (X1-DASyncReqEnrollment) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import and moved into the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below).

Copyright © 2026, Oracle and/or its affiliates.

Initial Sync Request Load Batch Process

Program subscriptions for legacy enrollments are created from inbound sync requests using the Initial Sync Request - Load Enrollment (X1-SILEN) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process transitions the inbound sync requests to the "Loaded" state.

Importing Event Participation Records

This topic provides additional details related to importing legacy event participation records from a demand response management system (DRMS).

Event Participation Import File Format

Legacy event participation import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
Legacy Event Participation ID	Required	The id for the event participation record in the DRMS application.
Reservation Start Date Time	Required	The start date and time of the event.
Reservation End Date Time	Required	The end date and time of the event.
Participation Start Date Time	Optional	The start date and time of the device's participation in the event, Some devices may participate in only a portion of an event. If not applicable, this field is blank.
Participation End Date Time	Optional	The end date and time of the device's participation in the event, Some devices may participate in only a portion of an event. If not applicable, this field is blank.
Legacy Controllable Device Id	Required	The id for the controllable device in the DRMS application.
Legacy Program External Id	Required	The id for the program in the DRMS application.
Legacy Program Subscription Id	Required	The id for the enrollment (program subscription) in the DRMS application.
Preemptive Opt Out	Optional	A flag that indicates if the event participation had opted-out of prior to the event. Valid values are values for the Preemptive Opt-Out (PREEMPTIVE_OPT_OUT_FLG) lookup, and include: • D1NO (No) • D1YS (Yes)

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
Event Outcome	Optional	A flag that indicates the outcome of the event. Valid values are values for the Outcome (OUTCOME_FLG) lookup, and include: • D1SP (Successful participation) • D1IO (In-flight Override) • D1ND (Non-responsive device) • D1ER (Error) • D1CN (Cancelled)
Event Participation Status	Required	A text string that specifies the status of the event participation. Valid values include: • PRE-EVENT • POST_EVENT • CLOSED • CANCELLED

Sample Record

Below is a sample file:

```
EXTEVTPART2    2022-09-01T08:00:00-05:00    2022-09-11T08:00:00-05:00
2022-09-01T08:00:00-05:00    2022-09-11T08:00:00-05:00    EXTDVC2
LEGACYPRG1    EXTENR2    D1YS    D1SP    PRE-EVENT
```

Upload Batch Process

Legacy event participation import files are uploaded using the Legacy Participation Event File Upload (X1-PTUPL) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process creates inbound sync requests in the "Pending" state (see below).

Inbound Sync Request Business Object

Event participation inbound sync requests are based on the Event Participation Sync Request BO (X1-DASyncReqEventParticipation) business object.

Inbound sync requests are created in the "Pending" state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition these to the "Transform/Validate" state where the data is validated and transformed as needed for import and moved into the "Validated" state (or "Validation Error" if an error is encountered). These are processed by an "Initial Sync Request Load" batch process (see below).

Initial Sync Request Load Batch Process

Legacy event participation records are created from inbound sync requests using the Initial Sync Request - Load Event Participation (X1-SLPAR) batch process. Use the **Batch Control** portal to view the parameters used by this batch process. This process should be run as needed for the initial import.

This process transitions the inbound sync requests to the "Loaded" state.

Copyright © 2026, Oracle and/or its affiliates.

Importing Facility and Network Location Data from a Distributed Energy Resource System

Facility and network location data includes service points and associated network entities such as transformers, feeders, substations, and weather stations and can be imported from a distributed energy resources system (DERMS) such as Oracle Utilities Grid Edge DERMS or Oracle Utilities Network Management System. Data to be imported is sent to a flat file, which is uploaded into the system via batch processing which creates inbound data synchronization (sync) requests. These sync requests are processed and validated by a monitor batch process, and then a separate batch process creates master data records from each.

Import Process Overview

This section provide an overview of the import process and outlines the various system objects used at each stage in the process. The overall process is as follows:

1. Data to be imported is exported from the DERMS into a flat file.
2. The Network Location File Upload (D1-NLUP) batch process is used to create inbound sync requests in the Pending state based on object-specific initial synchronization business objects. A separate inbound sync request is created for each row in the import file.
3. The inbound sync requests are processed using the Sync Request Monitor (F1-SYNRQ) batch process.
4. The Maintain Network Location (D1-MNLOC) batch process is used to create individual master data records from each inbound sync request.

Initial vs. Ongoing Uploads

Network location data can be uploaded both as part of the initial system set up, and as part of ongoing system maintenance. The process functions the same for both.

Initial uploads are used when setting up the system to import network location data including metered service points, transformers, feeders, substations, and weather stations. Ongoing uploads are used to update network location data as the data changes over time.

Network Location Import File Format

Network location import files use the following tab-separated values format:

Field Name	Optional/Required	Notes
CIS Service Point	Required	The ID of the metered service point in the customer information system.
External Transformer ID	Required	The external ID for the transformer. Used to populate the External Transformer ID when creating transformer records.
Transformer Name	Required	The name of the transformer. Used to populate the Description on transformer records.

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Optional/Required	Notes
External Feeder ID	Required	The external ID for the feeder. Used to populate the External Feeder ID when creating feeder records.
Feeder Name	Required	The name of the feeder. Used to populate the Description on feeder records.
External Substation ID	Required	The external ID for the substation. Used to populate the External Substation ID when creating substation records.
Substation Name	Required	The name of the substation. Used to populate the Description on substation records.
External Weather Station Name	Optional	The name of the weather station, Used to populate the Description on weather station records.

Sample Record

Below is a sample file:

```
CISSP1    EXTTRAN01    TRAN01    EXTFEED01    FEED01    EXTSUBST01    SUBST01
EXTWST01
```

Upload Network Location Batch Processes

Import files are uploaded using he Network Location File Upload (D1-NLUP) batch process. These should be run as needed for the initial upload, as well as subsequent ongoing updates.

This batch process uses the following parameters:

- **File Name**: The name of the file to upload. File names can use wildcards (Required)
- **Flat Path**: The file location where the upload file is located. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* for more information. (Required)
- **Character Encoding**: The character encoding to use for the extract file. The default is UTF-8.
- **Processed File Extension**: The text appended to the file name after the file has been processed. Default is "PROCESSED".
- **Error On No Files Found**: Indicates whether or not an error is issued if no files are found matching the given name in the given path. Valid values are "true" (default) and "false".
- **External System**: The external system that created the upload file (Required)

Use the **Batch Control** portal to learn more about these and other parameters used by this batch process.

SP Network Location Sync Request Business Object

Inbound sync requests are created by the upload batch process, and are based on the SP Network Location Sync Request (D1-SyncReqInSPNetworkLocation) business object.

Copyright © 2026, Oracle and/or its affiliates.

The table below outlines the stages of the inbound sync request lifecycle. Use the **Business Object** portal to learn more about this business object and the details of each lifecycle state.

State	Notes
Pending	Inbound sync requests begin in this state. The Sync Request Monitor (F1-SYNRQ) batch process is used to transition sync requests in this state to the Validate state.
Validate	In this state the data is validated and transformed as needed for import. In this state, system IDs are derived from the external IDs in the upload file.
Validation Error	Sync requests enter this state if an error occurs during validation in the Validate step. If a sync request needs to be updated manually, it can be put into this state via the **Correct** action in the **Record Actions** section.
Processed	Initial Load sync requests enter this state when a record has been successfully loaded,
Discarded	The sync request enters this state if the sync request is discarded from the Validation Error state,

Maintain Network Location Batch Process

Master data records are created from inbound sync requests using he Maintain Network Location (D1-MNLOC) batch process. This should be run as needed for the initial import, as well as subsequent ongoing updates.

This batch process uses the following parameters:

- **Override Low Identifier**: A custom low id to use in calculating a range of records to process for a thread. The parameter value can be an actual number between 0's (e.g. 000000000) and 9's (e.g. 9999999999) or it can be set to 'auto'. If auto is configured, it is set to the lowest current value on the database table associated with the background process.
- **Override High Identifier**: A custom high id to use in calculating a range of records to process for a thread. The parameter value can be an actual number between 0's (e.g. 000000000) and 9's (e.g. 9999999999) or it can be set to 'auto'. If auto is configured, it is set to the highest current value on the database table associated with the background process.

Use the **Batch Control** portal to view the parameters used by these batch processes.

Extracting M&V Data for use in External Systems

This section describes how M&V data can be extracted from for use in external systems. This includes event impact and statistics data, which could be used by analytics applications used to compare the impact details for a Treatment group and its corresponding Control group, or to analyze event statistics.

- Extracting M&V Event Impact Data - Treatment Groups
- Extracting M&V Event Impact Data - Control Groups
- Extracting M&V Event Statistics

Copyright © 2026, Oracle and/or its affiliates.

Extracting M&V Event Impact Data - Treatment Groups

M&V event impact data for Treatment groups (Device Population groups) can be extracted for import into an external system via a batch process which creates a file containing data from Service Point Event Outcome records.

See About Service Point Event Outcome Snapshots for more information.

Batch Control

M&V event impact data for Treatment groups is extracted using the Stage Impact Extract - Treatment Group (X1-SIETG) batch process. This batch process downloads data into a file, the details of which are specified by the batch control parameters.

This batch processes use the following parameters:

- **File Integration Type**: The file integration type (see below) used to extract M&V event impact data for Treatment groups (see below).
- **File Name**: The name of the downloaded file. File names can use wildcards (Required).
- **File Path**: The file location to which the file will be downloaded. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* the for more information (Required).
- **File Format**: The extract file format. Valid values are CSV (delimited) FIXD (fixed position output), XML (XML document), JSON (JSON document) or JSNL (JSON Lines).
- **File Delimiter**: The delimiter to use for the CSV file format.

Use the **Batch Control** portal to view additional details about these and other parameters.

File Integration Type

The M&V event impact extract process uses the Stage Impact Extract - Treatment Group (X1-SIETGExtract) file integration type, defined as a parameter on the Stage Impact Extract - Treatment Group (X1-SIETG) batch process.

Implementations that require a different extract format can create a custom file integration type that defines their own extract format and process. This includes creating a custom extract algorithm and a custom extract data area.

Refer to Extract Using File Integration for more information about the role of file integration types.

File Format

The file format of the extract file is defined by the Stage Impact - Treatment Group Record (X1-SIETGExtractRecord) data area, specified on the Stage Impact Extract - Treatment Group (X1-StageImpactTreatmentGroup) file integration record on the Stage Impact Extract - Treatment Group (X1-SIETGExtract) file integration type.

M&V event impact data for Treatment groups files use the following format:

Field Name	Notes
Event ID	This is the External Event ID in the Program Event

Copyright © 2026, Oracle and/or its affiliates.

Field Name		Notes
Stage ID		This is the External ID in the Program Event
Treatment Group		Entity Group of the Service Point Event Outcome
Service Point		Service Point of the Service Point Event Outcome
		This is the Customer's SP ID - external ID
Stage Impacts		
	Date Time	This is the Outcome Date Time from the Service Point Event Outcome Quantity
	Quantity	
	Quantity Description	Unit of Measure Description and Service Quantity Identifier Description
	Temperature	
	Temperature Description	This is the description on temperature quantity, for example Fahrenheit

Extracting M&V Event Impact Data - Control Groups

M&V event impact data for Control groups can be extracted for import into an external system via a batch process which creates a file containing data from Service Point Event Outcome records.

See About Service Point Event Outcome Snapshots for more information.

Batch Control

M&V event impact data for Control groups are extracted using the Stage Impact Extract - Control Group (X1-SIECG) batch process. This batch process downloads data into a file, the details of which are specified by the batch control parameters.

This batch processes use the following parameters:

- **File Integration Type**: The file integration type (see below) used to extract M&V event impact data for Control groups (see below).
- **File Name**: The name of the downloaded file. File names can use wildcards (Required).
- **File Path**: The file location to which the file will be downloaded. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* the for more information (Required).
- **File Format**: The extract file format. Valid values are CSV (delimited) FIXD (fixed position output), XML (XML document), JSON (JSON document) or JSNL (JSON Lines).
- **File Delimiter**: The delimiter to use for the CSV file format.

Copyright © 2026, Oracle and/or its affiliates.

Use the **Batch Control** portal to view additional details about these and other parameters.

File Integration Type

The M&V event impact extract process uses the Stage Impact Extract - Control Group (X1-SIECGExtract) file integration type, defined as a parameter on the Stage Impact Extract - Control Group (X1-SIECG) batch process.

Implementations that require a different extract format can create a custom file integration type that defines their own extract format and process. This includes creating a custom extract algorithm and a custom extract data area.

Refer to Extract Using File Integration for more information about the role of file integration types.

File Format

The file format of the extract file is defined by the Stage Impact - Control Group Record (X1-StgImpactExtControlGrp) data area, specified on the Stage Impact Extract - Control Group (X1-StageImpactControlGroup) file integration record on the Stage Impact Extract - Control Group (X1-SIECGExtract) file integration type.

M&V event impact data for Control groups files use the following format:

Field Name		Notes
Event ID		This is the External Event ID in the Program Event
Stage ID		This is the External ID in the Program Event
Control Group		Control Group linked the M&V Transaction's Subscription's Entity Group
Service Point		Service Point belonging to the Control Group
		This is the Customer's SP ID - external ID
Control Group Load		
	Date Time	
	Quantity	
	Quantity Description	Unit of Measure Description and Service Quantity Identifier Description
		OR
		Unit of Measure Description + 'Net'

Extracting M&V Event Statistics

M&V event statistics can be extracted for import into an external system via a batch process which creates a file containing M&V event statistics information from both Service Point Event Outcome records and M&V transactions.

Copyright © 2026, Oracle and/or its affiliates.

See About Service Point Event Outcome Snapshots and About M&V Transactions for more information.

Batch Control

M&V event statistics are extracted using the Event M&V Statistics Extract (X1-EVMVE) batch process. This batch process downloads data into a file, the details of which are specified by the batch control parameters.

This batch processes use the following parameters:

- **File Integration Type**: The file integration type (see below) used to extract M&V event statistics (see below).
- **File Name**: The name of the downloaded file. File names can use wildcards (Required).
- **File Path**: The file location to which the file will be downloaded. This can be a file location on a local server or a location in Object Storage. See Referencing URIs in the *Framework Administrative User Guide* the for more information (Required).
- **File Format**: The extract file format. Valid values are CSV (delimited) FIXD (fixed position output), XML (XML document), JSON (JSON document) or JSNL (JSON Lines).
- **File Delimiter**: The delimiter to use for the CSV file format.

Use the **Batch Control** portal to view additional details about these and other parameters.

File Integration Type

The M&V event statistics extract process uses the Event M&V Statistics Extract (X1-EVMVExtract) file integration type, defined as a parameter on the Event M&V Statistics ExtractX1-EVMVE) batch process.

Implementations that require a different extract format can create a custom file integration type that defines their own extract format and process. This includes creating a custom extract algorithm and a custom extract data area.

Refer to Extract Using File Integration for more information about the role of file integration types.

File Format

The file format of the extract file is defined by the Event M&V Statistics Extract Record (X1-EVMVExtractRecord) data area, specified on the Event M&V Statistics Extract (X1-EVMVExtract) file integration record on the Event M&V Statistics Extract (X1-EVMVExtract) file integration type.

M&V event statistics extract files use the following format:

Field Name	Notes
Event ID	This is the External Event ID in the Program Event.
Stage ID	This is the External ID in the Program Event
Treatment Group	Entity Group from the Service Point Event Outcome
Control Group	Control Group linked to the Treatment Group

Copyright © 2026, Oracle and/or its affiliates.

Field Name	Notes
Coefficient of Determination	From M&V Transaction
Multiple R	From M&V Transaction
Standard Error	From M&V Transaction
Number of Observations	From M&V Transaction
B0	From M&V Transaction Service Quantities
B0 Intercept Standard Error	From M&V Transaction
B0 Confidence Interval Low	From M&V Transaction
B0 Confidence Interval High	From M&V Transaction
Coefficient for Time (B1)	From M&V Transaction Service Quantities
B1	From M&V Transaction
B1 Confidence Interval Low	From M&V Transaction
B1 Confidence Interval High	From M&V Transaction
B2	From M&V Transaction Service Quantities
B2 Standard Error	From M&V Transaction
B2 Confidence Interval Low	From M&V Transaction
B2 Confidence Interval High	From M&V Transaction
B3	From M&V Transaction Service Quantities
B3 Standard Error	From M&V Transaction
B3 Confidence Interval Low	From M&V Transaction
B3 Confidence Interval High	From M&V Transaction

Defining Other Options

This section includes topics covering administrative options for other functionality used by Customer Program Management.

- General Options

Copyright © 2026, Oracle and/or its affiliates.

- Approvals
- Understanding Completion Events
- Defining Asset Completion Event Types
- Defining Completion Events

General Options

- Defining Installation Options
- Access Control
- E-Mail Configuration
- Three Decimals Places for Currency
- Defining Colors

Defining Installation Options

This section provides additional setup information related to your installation options record.

Refer to Installation Options - Main for more information.

- Formatted Address
- Address Geocoding

Formatted Address

Throughout the system address information is formatted for display based on standard address constituents.

> **Note**
>
> Logic to format an address resides in an "Address Information" algorithm on the installation options.

Address Geocoding

The scheduler has to be provided with the geocode of any location that may be included on a crew's route. These include locations of assets, service centers, substations, and so on.

Unless a geocode is provided, algorithms on the various business entities that capture an address attempt to geocode the address when the entity is added or when the address changes.

The minimum match quality level for address geocoding is configurable. By default, the base system requires an exact match, which is the maximum quality level. You can control the quality level of geocoding for any entity that has an address by overriding the match quality parameter of the algorithms on each entity with a different minimum quality level, based on your business needs.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Logic to geocode an address resides in an "Address Geocoding" algorithm on the installations options.

Geocoding, when used in conjunction with GIS, provides location-based mapping and analysis of assets and activities residing within your infrastructure.

- Map layers display the operational details for services provided by your company.
- Exact location of activities, work orders, and other objects are easily viewed and managed directly from the map.
- New assets created from the map are already plotted with Geocode coordinates.

Access Control

Access control provides a way for your organization to limit or classify record information by “plant,” varying work locations, or work functions. For example, you may have asset types that are only used at your northern location, so you would designate those asset types as restricted to that location. In the same way, you can leverage access control to prevent users in the southern location from accessing work orders related to the northern location (and vice versa). This is referred to as “row level security” in the application framework.

Each location represents an access group and objects in the application may be owned by a specific access group. Access groups are defined in the **Access Group** portal and are assigned to records as an **Owning Organization**.

Refer to Row Security in the **Oracle Utilities Application Framework** documentation for more information on data access roles and how they work with access groups.

> **Note**
>
> This functionality is optional and not enabled by default. If your implementation needs to restrict access by owning organization, you will need to explicitly enable it as described below.

The following points describe the functionality at a high level:

- Some entities are directly secured by being assigned an owning organization value on the entity itself. For example, the work order entity is directly secured and therefore its owning organization field is populated.
- Some entities are indirectly secured by association to another related entity that is directly secured. Entities secured this way are not assigned an owning organization. For example, work activities are secured indirectly by association to their work order. If a user is allowed to view a work order they are also allowed to view all its work activities. There is no owning organization data security for individual activities.
- All other entities do not support this type of security. For example, operation and technical entities are not used by business users and as such should not be subject to data restrictions. A system administrator needs to be able to review all the data in these entities, unfiltered by owning organization, to properly address issues and exceptions.

The following sections further describe concepts related to data security by owning organization.

Copyright © 2026, Oracle and/or its affiliates.

Which Entities Are Secured?

Not all entities support this type of data security. The following business entities directly support security by owning organization:

- Asset
- Compatible Unit
- Crew
- Location (This includes all classes of locations stored in the "Node" maintenance object, for example, asset locations, storerooms, and so on.(
- Purchase Order Header
- Purchase Requisition Header
- Template Work Order
- Work Design
- Work Location
- Work Order
- Work Request

The following business entities indirectly support security by owning organization via association:

- Work Activity. Secured via association to work order
- Activity Reconciliation. Secured via association to work order
- Design Element. Secured via association to work design.
- Expedite Purchase Order. Secured via association to Purchase Order Header.
- Invoice Header. Secured via association to Purchase Order Header.
- Material Request Header. Secured via association to Storeroom (Node).
- Material Return Header. Secured via association to Storeroom (Node).
- Measurement. Secured via association to Asset.
- Physical Inventory Header. Secured via association to Storeroom (Node).
- Warranty. Secured via association to Asset.

The following administrative entities directly support security by owning organization:

- Activity Type
- Approval Profile
- Asset Type
- Business Unit
- Checklist Type
- Cost Adjustment Type
- Cost Center
- Distribution Code
- Expense Code

Copyright © 2026, Oracle and/or its affiliates.

- Labor Earning Type
- Location Type. (This includes all classes of location types stored in the "Node Type" maintenance object, for example, asset location types, inventory location types, and so on.)
- Planner
- Service Category
- Service Class
- Service Code
- Service History Question
- Service History Type
- Specification
- Work Category
- Work Class

Shared Access

Security by owning organization is optional but when it is enabled the application assumes every record on a directly secured entity is assigned with an owning organization value.

When records are missing a value, the application cannot enforce data restriction and therefore lets these records be accessible to all. While the application treats these records as unrestricted, it assumes they are only temporarily unrestricted until a user or some default rule assigns them with a value.

The correct approach for intentionally sharing records across owning organizations is to designate one or more shared access groups and assign those to appropriate users. For example, if some entities are to be shared between the north and south owning organizations you may want to designate a “north-and-south” owning organization, assign shared entities to this owning organization, and allows users from both locations to have access to it.

Default Owning Organization Assignment Rules

When owning organization security is enabled, entities that are directly secured need to have their owning organization field populated either by the user or process that created them or in absence of an explicit value by a default rule. The application leverages an **Audit** algorithm configured on the entity’s business object to assign it with a default value. While a common default rule for administrative entities is to default the owning organization from the user’s default access group, more complex rules are called for when assigning values for master and transactional entities. To find the rule used for a directly secured entity, find the corresponding audit algorithm configured on its business object.

When creating a new administrative entity that is directly secured by owning organization, the user’s default access group is presented on the user interface prior to saving the record, allowing the user to change the value if needed before creating the record. Note that if the user deliberately clears the defaulted value the application still enforces the default rule and the record would be saved with the user’s default access group.

Default rules for master and transactional objects are more complex and as such need to support a way for your organization to customize them to meet your business requirements. Therefore, the user’s default access group is intentionally not defaulted on the user interface before creating such an entity, allowing the business object audit rule to populate the value as needed.

Copyright © 2026, Oracle and/or its affiliates.

By default, product provided algorithms for defaulting an entity's owning organization only resort to using the user's default access group when a record is created. However, they all support an algorithm parameter that you may configure to enforce defaulting from the user's default access group also when existing records with no owning organization value populated are updated.

> **Note**
>
> If after applying the default rules of any object an assignment has not been made, i.e. the record still has an empty owning organization field, the application is not able to restrict access to this record and therefore leaves it allowed to all. This should not be confused with a configuration that intentionally associates entities with a shared owning organization.

Explicit Owning Organization Assignment Rules

There are times when the appropriate owning organization value can not be determined by the entity's default rule. In these unique situations, the process that creates the secured entity pre-populates it with the appropriate owning organization value when adding it. The following are examples of such processes:

- When generating preventive maintenance work orders, the application inherits their owning organization from the asset they are created for. The asset is referenced on the activity and therefore a default rule for the work order would not have access to the asset at creation time.
- When generating or interfacing other direct charges, the application inherits their owning organization from their activity's work order. Here too, the activity is only referenced at the detail record and not available when the charge header record is created..
- When generating follow-up work orders based on service history type configuration, the application inherits their owning organization from the service history's activity's work order.

Refer to the relevant algorithms associated with such processes for more information.

Creating Entities via Integration or in Batch

Typically, the user used to process web service calls or execute batch processes is a common user record that is not affiliated with any owning organization. As such , entities created by this common user should not rely on the user record's default access group for a proper owning organization assignment. For that reason, these common users should not have any default access group defined for them, yet they should be allowed access to all access roles.

Directly secured objects created via web services or in batch are designed to have their value explicitly provided and not rely on the user's default value. For example, a user creates an asset online they can either explicitly populate the asset owning organization or let the application default it to the user's default access group. However, when assets are interfaced from an external system their owning organization should be explicitly populated.

Filtering By Owning Organization

Secured entities, either directly or indirectly, may support an owning organization filter on their respective query portals. The filter allows the user to review data that belongs to a specific owning organization, yet this should not be confused with data restriction. When security by owning organization is enabled, the application restricts the data the user can access to regardless of whether they have used the filter or not. If the user has access to more than one owning organization then a filter may be useful to narrow down the results to a selected group.

Copyright © 2026, Oracle and/or its affiliates.

However, when security by owning organization is not enabled, data is unrestricted and the user may access any record.

Enabling Data Security By Owning Organization

The following points highlight the overall steps needed to enable data security by owning organization customers:

- Design and configure access groups and access roles to meet your business requirements. If some entities should be shared across all owning organization, define a customer shared access group that is linked to all access roles.
- Associate users with access groups and roles:
  - All user records for individuals accessing the application must be assigned with a default access group and may belong to one or more access roles. This access group is used as the default value assigned to some secured objects by various business rules.
  - All user records used for integration and batch processes should not be associated with a default access group but must have access to all access roles. This way data created by these shared user records are not stamped with an arbitrary owning organization yet all data is accessible to these processes.
- For upgrading customers, make sure existing records in the primary tables of directly secured entities have the correct value in the owning organization field. Depending on the volume of each table and the number of records that may need correction in each table you may choose to make these updates online or via a custom plug-in driven batch process.
- Run the **Apply Access Group VPD Policy** (W1APLVPD) batch process to enforce row level security on master and transactional objects that are directly secured by owning organization. Refer to the batch control definition for more information about this batch process.

E-Mail Configuration

This section provides information about requirements to configure settings for emails sent from within the system regarding objects such as approvals, purchase orders, timekeeping, and so on.

> **Note**
>
> The “System Email From” Option Type used with email notifications is not supported for Oracle Utilities Cloud Services, including Oracle Utilities Work and Asset Cloud Service and Oracle Utilities Customer Program Management Cloud Service. Email configuration for "Cloud" customers is defined in "Service" properties.

To configure the system for sending emails complete the following:

1. Create or update a **Feature Configuration** for email configuration (EMAILCONFIG).

   The **Feature Type** should be set as **External Messages**

2. Add the **System Email From Option Type** with a valid from address in the **Value** field. This identifies the sender to use for system emails.

   Example:

Copyright © 2026, Oracle and/or its affiliates.



3. Create a Message Sender with the following:
   - **Invocation Type**: Real-time
   - **Message Class**: RTEMAILSNDR
4. Select the **Context** tab on the Message Sender and add the SMTP information.

   This includes the following:
   - **SMTP Host Name**: Example: inernal-mail-router.yourcompany.com
   - **SMTP Password**: Your SMTP password
   - **SMTP Username**: Example: admin@oracle.com

Refer to the Real-Time Message Configuration and the Message Sender - Context section in the **Oracle Utilities Application Framework** documentation for more information.

Three Decimals Places for Currency

The overall system uses the base currency defined in **Installation Options**, and all financial transactions use the base currency. However, the purchasing system supports other currencies which can be defined in the **Currency** portal. The currency code will appear as an option wherever a currency code is required on purchasing records, such as on a blanket contract or purchase order. Any currency codes used in the purchasing system which are different from the base currency are converted into the base when the system creates financial transactions.

Oracle Utilities Work and Asset Management supports up to three digits after the decimal for currency.

To configure the system for three decimal currency, set the number of decimal positions when defining currency codes.

Refer to the Defining Currency Codes section in the **Oracle Utilities Application Framework** documentation for more information.

Defining Colors

Colors can be used to represent transactions and other entities in various portals. For example, colors can be used to distinguish cost categories as they are represented in graphs and pie charts.

The **Color** portal is used to view and maintain colors.

You can access this portal from **Admin**, select **WAM General**, then select **Color**.

The following zones may appear as part of the portal's **Main** tab page:

Copyright © 2026, Oracle and/or its affiliates.

- **Color List**: This zone lists all the colors. Broadcast a record to display the details of a selected record.
- **Color**: This zone provides information about the selected color.

Approvals

- Understanding Approval Processing
- Defining Approval Profiles
- Understanding Change Request Types
- Defining Change Request Types

Understanding Approval Processing

One of the ways that the system simplifies and speeds up work is by routing records when they require approval. For example, purchase requests, work requests, employee leave, and other functions can be configured to require approvals.

Basic approval processing requires that the person creating or managing a record that requires approval must create an approval request. This is done by clicking the **Submit for Approval** button on the record when it is in a pending approval status and then selecting the appropriate approval profile for the document. Once a record is submitted for approval, the request is emailed to approvers, sent to their To Do List, or both.

- The **Approval Request** shows the related document, the person who submitted it and when, who has approved it so far, and which roles still need to provide an approval decision.
- Approval requests use an **Approval Profile** to determine the properties and requirements for the approval. A common structure defines the approval process, users, and limits for each given functional area that uses approval processing. The **Approval Profile** indicated on the approval request identifies the people who need to approve the record and their role. Sometimes there are multiple approvers needed depending on how the approval profile is set up and your business practices. Often records will have a set of approval profiles which are defined for all records of that type. For example, when a new work request is routed for approval, the options for approval profile are limited to those defined for work requests. Most users only need to know which **Approval Profile** to choose when they submit a record for approval. This selection will depend on your business processes and how your approval profiles are configured. For example, when a user submits an approval request for a timesheet, they should only see approval profiles available for a timesheet. See Approval Profiles for more information.
- Approvers can access information about approvals that require their attention using the **Approval Dashboard**. Also, the person (requestor) who submitted the record for approval can check the status of the approval from a zone on their home page or by checking the individual record. The requestor is notified by e-mail when the record has been approved or if the request times out.
- Most approvers also have access to the **Approval Request Action** zone on the side **Dashboard** where they can record and submit their decision to reject or approve the approval request. Access to this zone is controlled by your **Portal Preferences** in your user profile.
- The **Log** tab on the record being approved indicates who approved the record and when.

Refer to About Approvals for more information.

Copyright © 2026, Oracle and/or its affiliates.

- Approval Hierarchy
- No Approval Needed
- Example

Approval Hierarchy

Approval profiles can reference one or many **To Do Roles**. The way in which you structure your approval profiles depends on how regulated the structure needs to be for your organization.

Your business practices may be such that your approvals can go to a limited number of supervisor and manager roles, or they can be more advanced having detailed levels. To this end, the To Do Roles you create can be shared across document types or they can be kept discrete to business process and functionality. This is also true for the **No Approval** user group. There can be one role to be reused across your approval processing configuration, or you can set more roles to address different functions.

One type of record, such as work order, might require several different **Approval Profiles** to address different situations. For example,

- PM work might be “auto-approved” since the work needed has already been analyzed, planned and approved.
- Emergency work might bypass normal approval processing since it might come in at off hours or will otherwise be subject to different approvals than regular work.
- Regular work will likely require general approval processing.

Approval Types

Approval Types indicate whether the approval role is standard, mandatory, or notification only:

- **Standard**: A standard approver might appear on the approval hierarchy with other approvers with equal approval authority. Any standard approver can record the decision before the record moves on to the next approver in the hierarchy.
- **Mandatory**: Indicates that this approver must provide an approval decision on the record. Approval processing is not complete until this decision is recorded.
- **Notification Only**: Notification only approvers are only sent an email that the approval is pending. The system does not validate an amount limit and the approver does not have to respond in order for the record to be approved.

> **Note**
>
> If an approver with enough authority approves the request, and there are no other mandatory approvers, the approval request is complete. In this case, other *standard* and *notification only* approvers on the approval path are skipped. No action can be taken to approve or reject the approval request.

Additional Approval Profile Settings

- **Initial Notification Method**: Indicates whether the approver receives an e-mail, an alert in their To Do List, both, or none.

Copyright © 2026, Oracle and/or its affiliates.

- **Authorized to Approve**: (Non-Amount Based only) When multiple To Do Roles are added to the profile, this field identifies which is authorized to approve. At least one to do role must be added with this value set to Yes.
- **Amount Limit**: Provides the maximum amount that the approver in the indicated role is allowed to approve. If the value on the record in need of approval exceeds this limit for the role, the approval path must move on to the next approver or the requester is notified that no one on the approval profile has sufficient authority to approve the request.
- **Timeout Threshold**: Indicates the amount of time that the approver has to respond to the approval request, in days. For example, if the threshold is set to “2”, the approver has 2 days to approve requests.
  If the approver does not respond within the set time, another alert is sent. Additional alerts are sent up to the set number of retries.
- **Maximum Retry**: Indicates the number of times that the system will retry sending the approval request to approvers with the To Do Role before timing out the request. For example, if the Timeout Threshold is set to “2” and the Maximum Retry is set to “3” the system will try three times for a total of 6 days resending the alert or email every 2 days. If the approvers fail to respond, the request is routed to the next approver in the hierarchy and that person receives the email or alert.

No Approval Needed

User groups referenced on **No Approval Required** approval profiles indicate the user groups that can create an approval request in which the system automatically approves the record. Some examples of where a **No Approval Required** approval profile can be used might be for emergency type work which could be generated at anytime or for routine maintenance type work that has a lower impact on business processes.

Example

The following table provides examples of the types of roles you may need to create in order to support your approval hierarchy:

Admin Object	Example of Data to Create
To Do Types	These default types are delivered with the base package with the ToDo Type Usage set to **Automatic**: • Approval Type — Employee Unavailability • Approval Type — ODC • Approval Type — Project • Approval Type — Purchase Order • Approval Type — Purchase Requisition • Approval Type — Timesheet • Approval Type — Work Order • Approval Type — Work Request
To Do Role	• General Approver

Copyright © 2026, Oracle and/or its affiliates.

Admin Object	Example of Data to Create
	• General Director • General Manager • General Notification Only Approver • General Supervisor • General Work Management Supervisor • General Work Management Manager • General Purchasing Manager • General Purchasing Supervisor
User Groups	• Authorized for No Approval
Approval Profiles	• Approval Amount – General Purchasing Approver – High Value Purchasing Approver – General Work Request Approver – General Work Order Approver – High Value Work Order Approver – Project Approver – Other Direct Charges Approver • No Approval – Emergency Work Approval • No Amount – Employee Unavailability Approver – Timesheet Approver

Using a scenario of a moderate structure with a separate approval profile for each document type, refer to the following table for an example of the hierarchy:

Seq	To Do Role	Approval Type	Initial Notification Method	Amount Limit	Time Out Threshold	Maximum Retry
10	Supervisor	Standard	To Do List Only	$1000	2 (days)	1 (retry)
20	General Manager	Mandatory	Both (email and To Do List)	$5000	2 (days)	2 (retries)

This example establishes a hierarchy where the first role to receive the approval request is at the supervisor level with a minimal approval limit. This supervisor would approve the request only to inform the general manager that they approve, but since the General Manager has been set with a mandatory approval type, the General Manager must approve the request for the processing to be considered complete.

Copyright © 2026, Oracle and/or its affiliates.

Defining Approval Profiles

The **Approval Profile** portal is used to display and maintain approval profiles.

You can access the portal from the **Admin** menu, select **WAM General,** then select **Approval Profile**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Approval Profile**: This zone provides information about the approval profile.
- **Approval Information**: This zone provides additional details about the approval profile, such as the To Do to be used and the hierarchy to use when an approval record is submitted.
- **User Groups**: This zone provides information on what user groups can create the approval request.

Understanding Change Request Types

A change request type defines the related object, for example, Asset, that change requests of this type are for and the applicable approval profiles for such change requests.

Refer to About Change Requests for more information.

The related object is defined by the extendable lookup “Change Request Related Object”. The product comes with support to several key entities but your implementation may extend this configuration to support additional entities.

The following sections describe concepts and features related to change request configuration.

- Entity Specific Versus General Change Requests
- Initiating Change Requests
- Enabling Change Request For Another Entity

Entity Specific Versus General Change Requests

An "Entity Specific” change request type supports the tracking and approval of changes requested for an existing entity. A general change request type supports all other types of requests, for example:

- A request to update several entities where the request describes in text what should be changed.
- A request to add one of more entities that do not exist yet.

Initiating Change Requests

A dashboard zone allows users to view or create a change request for the entity currently being displayed when at least one active change request type exists for such entity. For example, if a change request type exists for an asset then the dashboard zone would appear when an asset is displayed.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Though the product comes with some “Change Request Related Object” extendable lookup values, you can still control which ones to enable by only creating the change request types for the ones you need.

Only one in-progress change request is allowed for a specific entity, for example a specific asset, therefore the button, when applicable, may only be available if such a request does not already exist. If the change request exists, a link to the in-progress request is provided, allowing any user to navigate to it and review the requested changes.

If there is no in-progress change request for that specific instance, a **Create Change Request** button is made available on the **Dashboard** for users with or without “Update” security access to that object.

By definition, this dashboard is not applicable to “General” change requests. Users need to use the menu to add such change requests.

Enabling Change Request For Another Entity

To enable change request for another object, you need to do the following:

- Add a new value to the “Change Request Related Object” extendable lookup.
  - Select the corresponding entity value. For example, Cost Center.
  - Select the corresponding FK reference, which is used to search for a specific instance of the related object. For example, a particular cost center.
- Create the change request types for the new related object.
- Associate the maintenance portal of that entity to the dashboard change request context sensitive zone as follows:
  - Find the maintenance portal by navigating to **Admin**, select **Portal**, and then search for the portal page.
  - Click the **Navigation Option** hyperlink to navigate to the portal’s navigation option.
  - Go to the **Target Transaction** and copy the navigation key.
  - Navigate to **Admin**, click **Context Sensitive Zone**, and then paste the navigation key into the **Navigation Key** field and click **Search**.
  - Add the **Change Request Dashboard** (W1–CGRQDASH) to the **Zone** column for this navigation key.
- If you want to have a nicely formatted UI on the change requests with separate fields for users to enter new values, you need to do the following:
  - Create a “mini” BO for that object.
  - Navigate to the entity’s maintenance object, select the **Options** tab, and select the change request “Valid BO Option Type” for the entity’s maintenance object options.
  - Define the “mini” BO as a BO option of **Option Type** “Change Request BO” on the base BO for that object. Please refer to those objects delivered as product values for the “Change Request Related Object”, except for “**General**” change requests. For example, if you go to BO “W1-Vendor”, you will see such BO options.
- Please Note:

Copyright © 2026, Oracle and/or its affiliates.

  - The mini BO should simply include a list of elements you would like to include in the change request process but without **Record Actions** and so forth. Please refer to our base “mini” BO’s for an example.
  - Without a “mini” BO, users can only describe the changes they would like to make in the detail description text field. As such, users need to manually make the updates in the system upon approval of the change request and then update the change request to an “Applied” status. Please note that change requests of a “**General**” type are not for any specific object thus, are not a “mini” BO by definition.
- If you want to have audit logs for changes to elements defined in the mini BO, please add the audit algorithm ‘W1GENCGRQLOG’ to the BO’s that the mini BO is added to as a BO option.
- Flush the online cache.
  - Replace the cis.jsp section of the url with flushAll.jsp (case sensitive) and refresh the url.
  - Once finished, it is traditional to close all of your browser windows and log in.

Defining Change Request Types

The **Change Request Type** portal is used to display and maintain change request types.

You can access the portal from **Admin**, select **Resource Management**, then select **Change Request Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Change Request Type List**: This zone lists all change type records. Broadcast a record to display the details of the selected record.
- **Change Request Type**: This zone provides general information about the selected change request type as well as valid approval profiles.

Refer to Understanding Change Request Types for more information.

Understanding Completion Events

Completion events are used to capture closeout information against activities received from an external application. For example, completion events may be configured for several types of asset processing such as updating worked assets, creating service histories, and creating operational read information.

All external activities have a set of required and optional completion events that can be executed to see the activity to completion. The **Completion** tab on the activity shows any other system objects that are associated to the completed object. For example, if you are looking at the completion event for an activity, the **Completion** tab shows the valid completion-related actions, service history, and measurement information that may be associated with the activity, asset, or location in context.

The **Relationship Information** section shows any other system objects that are associated to the completed object. For example, if you are looking at the completion event for an activity, the relationship information shows the asset, any components, the location, service history, and so on.

- Processing an Activity Completion Message
- Completion Event Workflow

Copyright © 2026, Oracle and/or its affiliates.

Processing an Activity Completion Message

Activity completion is captured in relation to an asset listed on the activity and includes the following details:

- Was any work performed on the asset?
- Was one or more measurements taken for this asset?
- Was there one or more service histories filled out for this asset?

The completion message is locally queued and when the device is connected, the mobile application automatically delivers it to the server.

Completion Event Workflow

Completion events are used to capture closeout information against activities received from an external application. The following is the typical workflow of a completion event:

Pending

This is the initial status of the completion event.

Executed

This status checks for the required completion event.

Discarded

This status indicates the completion event was discarded and is eligible for deletion.

Defining Asset Completion Event Types

You use the **Completion Event Type** portal to display and maintain asset completion event types.

Refer to Understanding Asset Completion Event Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Completion Event Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Completion Event Type** : This zone provides information about the selected completion event type, including as inbound communications and attachments.
- **Unlink Activity Types**: This zone enables the removal of an activity type from the completion event type. To remove an activity type select the activity type(s) to remove and click **Remove**. The selected activity types will no longer appear in the **Unlink Activity Types** zone.
- **Link Activity Types**: This zone enables adding an activity type to the completion event type. To link an activity type:
    1. Enter search criteria n the **Activity Type Description** field and click **Search**. The zone will display a list of activity types matching your search criteria.

Copyright © 2026, Oracle and/or its affiliates.

2. Select the activity type(s) to add and click **Link**. The selected activity types will now appear in the **Unlink Activity Types** zone.

Defining Completion Events

The **Completion Event** portal is used to maintain various completion events.

Refer to Understanding Completion Events for more information.

You can access this portal from **Admin**, select **Work Management**, then select **Completion Event**. The user is brought to a query portal with options for searching for a specific completion event. Once the completion event has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zone may appear as part of the portals **Main** tab page:

- **Completion Event**: This zone displays various details about the completion event, such as assets, components, locations and so on associated with the completion event, as well as any related objects and attachments.
- **Completion Event Related Objects**: This zone displays the related objects associated with the completion event.
- **Attachments**: This zone enables users to manage all types of attachments for the completion event.

Defining Oracle Utilities Rate Cloud Service Options

- Oracle Utilities Rate Cloud Service Admin Data
- Creating Oracle Utilities Rate Cloud Service Users
- Initiating Calculations from an External System

Oracle Utilities Rate Cloud Service Admin Data

Oracle Utilities Rate Cloud Service includes access to pages, portals, and zones used to create and maintain various types of administrative configuration data, including:

- Characteristic Types
- Currencies
- Distribution Codes
- Revenue Classes
- Item Device Types
- Frequencies
- Service Types
- Service Quantity Identifiers
- Time of Use codes

Copyright © 2026, Oracle and/or its affiliates.

- Units of Measure

Refer to the Defining General Options chapter in the *Administrative User Guide* for detailed information about setting up currencies and characteristic types.

Refer to the Defining Financial Transaction Options chapter in the *Administrative User Guide* for detailed information about setting up distribution codes and revenue classes.

Refer to the Defining Device Options chapter in the *Administrative User Guide* for detailed information about setting up device types.

Refer to the Rates chapter of the *Business User Guide* for detailed information about setting up frequency codes.

Refer to the Defining Device Measurement Options chapter in the *Administrative User Guide* for detailed information about setting up service types, units of measure, service quantity identifiers, and time of use codes.

Note: Some fields and options used by these types of configuration data are not applicable when using Oracle Utilities Rate Cloud Service when compared to using Oracle Utilities Customer Cloud Service.

Creating Oracle Utilities Rate Cloud Service Users

Oracle Utilities Rate Cloud Service includes a “Rating Services” user group that provides access to the specific pages, portals, and features included with the cloud service.

The “Rating Services” user group should be used as the basis for all new Oracle Utilities Rate Cloud Service users.

Refer to the Defining Security and User Options chapter in the *Administrative User Guide* for more information about creating new users.

Initiating Calculations from an External System

Oracle Utilities Rate Cloud Service includes an inbound web service that allows rate calculation to be initiated from an external system.

The `c1rateCalculation` inbound web service is an inbound REST web service that can be used to request a rate calculation. Some specifics regarding this web service include:

- Master data (such as person, account, service agreement, service point, device, or usage) is not required.
- Service quantities and characteristic types/values must be included with the web service request.
- Calculation results are consistent with online rate calculation using the Rate Check feature.
- Calculation results are returned to the external system that submitted the request.
- Calculation results are not stored in Oracle Utilities Rate Cloud Service.

Refer to the External Messages chapter in the *Administrative User Guide* for more information about inbound web services.

Use the **Inbound Web Service** portal to view details regarding the `c1rateCalculation` inbound web service.

Copyright © 2026, Oracle and/or its affiliates.

Defining Oracle Utilities Billing Cloud Service Options

- Oracle Utilities Billing Cloud Service Admin Data
- Creating Oracle Utilities Billing Cloud Service Users

Oracle Utilities Billing Cloud Service Admin Data

Oracle Utilities Billing Cloud Service includes access to pages, portals, and zones used to create and maintain various types of administrative configuration data. The menus provided with Oracle Utilities Billing Cloud Service display only those menu items that correspond to capabilities that are available with the cloud service.

> **Note**
>
> Not all fields and field values available in Oracle Utilities Customer to Meter associated with these menu items are used with Oracle Utilities Billing Cloud Service.

This page outlines the administrative areas available with Oracle Utilities Billing Cloud Service based on the **Admin** menu, including:

- Aggregation
- Billing
- Communication (Activity Type only)
- Conversion
- Customer
- Database
- Device
- Device Installation
- Financial
- General
- Geographic
- Implementation Tools
- Integration
- Market Transaction Management
- Open Market
- Rates
- Reporting
- Sales & Marketing (Terms and Conditions only)

Copyright © 2026, Oracle and/or its affiliates.

- Security
- System
- Umbrella Agreement
- Usage
- VEE
- Batch Operations
- Conversion Support

Refer to specific chapter and sections in the *Administrative User Guide* for more information about working with these above listed functional areas.

Creating Oracle Utilities Billing Cloud Service Users

Oracle Utilities Billing Cloud Service includes the following user groups:

- Billing Cloud Services: A user group that provides access to the specific pages, portals, and features included with the cloud service
- Billing Cloud Service Batch Services: A user group that enables users to execute batch processes used by the cloud service

The “Billing Cloud Services” user group should be used as the basis for all new Oracle Utilities Billing Cloud Service users.

Users who will be responsible for running batch processing should be part of the “Billing Cloud Service Batch Services” user group.

Refer to the Defining Security and User Options chapter in the *Administrative User Guide* for more information about creating new users.

Smart Grid Gateway Adapters

Oracle Utilities Smart Grid Gateway (SGG) Adapters support communication with various third-party head end systems to provide the following functionality:

- **Payload Processing**: Receiving usage and device event payloads from the head end system (also referred to as Measurement Data and Device Event Loading and Processing)
- **Smart Meter Command Processing**: Sending/receiving messages to/from third-party applications to initiate smart meter commands from Smart Grid Gateway

Smart Grid Gateway Implementations

Oracle Utilities Smart Grid Gateway Adapters can be implemented in one of two manners:

- **Middleware Implementations** use Oracle Service Bus (OSB) for payload processing and Oracle Business Process Execution Language (BPEL) to communicate with head end systems when initiating smart meter commands. Middleware implementations can be used **only** with on-premises implementations of Oracle Utilities Smart Grid Gateway.
- **Native Implementations** use batch processing for payload processing, and specific communication business objects to communicate with head end systems when initiating smart meter commands. Native implementations can be used with **either** on-premises implementations of Oracle Utilities Smart Grid Gateway, or with Oracle Utilities cloud

Copyright © 2026, Oracle and/or its affiliates.

services such as Oracle Utilities Meter Solution Cloud Service or Oracle Utilities Customer Cloud Service.

The following sections provide details about each type of implementation.

- Smart Grid Gateway Adapter Middleware Implementations
- Smart Grid Gateway Adapter Native Implementations

Smart Grid Gateway Adapter Middleware Implementations

> **Note**
>
> Unless otherwise noted, the contents of this section apply to middleware (on-premises) implementations of Oracle Utilities Smart Grid Gateway only, and do NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

Oracle Utilities Smart Grid Gateway (SGG) Adapters support communication with various third-party head-end systems. Smart Grid Gateway uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) to facilitate communication with the head-end systems. Most if not all of the functionality shown below is available in each SGG Adapter.

Measurement Data and Device Event Loading: Data parsing and transformation via Oracle Service Bus from head-end system format into the Oracle Utilities Service and Measurement Data Foundation unified format for measurement data and device events.

Measurement Data and Device Event Processing: Configurable mapping for head-end system status codes and device event names to Oracle Utilities Service and Measurement Data Foundation standard values.

Smart Meter Command Processing: Sending/receiving messages to/from third-party applications to initiate smart meter commands from SGG. Most head-end systems support the following types of commands and communications:

- **Device Status Check:** Business objects and BPEL processes to support issuing device status check commands.
- **Meter Commissioning**: Business objects and BPEL processes to support issuing meter commissioning commands (including registration and installation commands).
- **Meter Decommissioning**: Business objects and BPEL processes to support issuing meter decommissioning commands.
- **Meter Retirement:** Business objects and BPEL processes to support issuing meter retire (deregistration) commands.
- **On-Demand Read**: Business objects and BPEL processes to support issuing on-demand read commands.
- **Remote Connect**: Business objects and BPEL processes to support issuing remote connect commands.
- **Remote Disconnect**: Business objects and BPEL processes to support issuing remote disconnect commands.

Copyright © 2026, Oracle and/or its affiliates.

This information in this guide describes how to configure, extend, and test the Smart Grid Gateway Adapters for individual head-end systems. The following head-end systems each have a specific SGG Adapter that can be configured: Itron OpenWay, Landis+Gyr, Networked Energy Services, MV90 for Itron, Sensus RNI, and Silver Spring Networks. Customers also can create their own customized adapter for using Smart Grid Gateway with a specific head-end system. See Creating a Custom Adapter for Smart Grid Gateway for more information.

- Configuring Smart Grid Gateway Adapters
- Using Smart Grid Gateway Middleware Test Harnesses
- Creating a Custom Adapter for Smart Grid Gateway

Configuring Smart Grid Gateway Adapters

This section describes how to configure Oracle Utilities Smart Grid Gateway Adapters for each supported head-end system. Use the cross-references below to go to the adapter of interest.

Itron OpenWay

Landis+Gyr

Networked Energy Services

MV90 for Itron

Sensus RNI

Silver Spring Networks

> **Note**
>
> This section applies to middleware (on-premises) implementations of Oracle Utilities Smart Grid Gateway only. This section does NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

- Itron OpenWay
- Landis+Gyr
- MV90 for Itron
- Networked Energy Services
- Sensus
- Silver Spring Networks

Itron OpenWay

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports communication with the Itron OpenWay application, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read commands. The following table describes the attributes of the adapter:

Copyright © 2026, Oracle and/or its affiliates.

Attribute	Details
Currently Supported Versions	Itron OpenWay 3.70, 3.90, 4.1, 6.10 OpenWay Collection Manager 5.4
Protocol	Proprietary
Market(s)	Worldwide
Architecture	RF Mesh

- Itron OpenWay Adapter Processing
- Configuring an Itron OpenWay Head-End System
- Configuring Itron OpenWay Extendable Lookups
- Using the Itron OpenWay Test Harness

Itron OpenWay Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

Measurement and device event data must be requested from the Itron OpenWay head-end system via Scheduled Read commands. See Scheduled Read Commands for more information.

- Scheduled Read Commands
- Initial Measurements
- Device Events
- Base Package Business Objects

Scheduled Read Commands

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Itron OpenWay.

Copyright © 2026, Oracle and/or its affiliates.

Measurement and device event data must be requested from the Itron OpenWay head-end system via Scheduled Read commands. Schedule read commands are used to request meter reads and/or device events for a group of devices for a specified date and time. Scheduled read commands can be created such that recurring requests based on the same command parameters are sent to the head-end system at regular intervals. For example, a command could be configured to request daily (24 hour) scalar meter readings once per day for a specified group of devices. Another command could be configured to request weekly (168 hours) device events once per week for a group of devices.

Scheduled read commands should be created for all devices from which meter readings and/or device events are to be uploaded via Oracle Utilities Smart Grid Gateway. Separate scheduled read commands can be created for specific groups and for specific types of data. For example, for a given group of scalar devices, you might create separate commands to request scalar usage data and to request device events from the same group (both of which can be based on different recurrence patterns). The specifics concerning request specifics and recurrence patterns should be based on the requirements of the implementation.

Attributes used to define a schedule read command include the following:

- **Status:** the status of the activity
- **Schedule Read Type:** the type of scheduled read
- **Schedule Information:** information about the schedule for the read, including:
  - **One Time Read:** indicates if the request is a one-time request. One-time requests have a defined date/time range during which request and recurrences can take place.
  - **One Time Start Date Time:** specifies the start time for one-time requests (applicable only if **One Time Read** is set to “Yes”)
  - **One Time End Date Time:** specifies the end time for one-time requests (applicable only if **One Time Read** is set to “Yes”)
  - **First Daily Measurement Time**: Indicates starting time for measurements for each day. This field, along with the value from **IMD Length**, will be used to determine when interrogation requests should be made. For example, if the **First Daily Measurement Time** is set to 08:00AM and **IMD Length** is set to 12 hours (12:00:00) then there would be two requests made per day:
    - 8:00AM - 8:00PM
    - 8:00PM - 8:00AM
  - **Interrogation Buffer**: This parameter impacts when interrogation requests are sent. It is entered in the format of hours:minutes:seconds. For example, if set to 1 hour (01:00:00) for our above example, the first request of the day for 8:00AM - 8:00PM would be sent to Itron OpenWay at 7:00PM. The buffer time will decrease if the activity is monitored after 7:00PM, so if the monitor process is executed at 7:30PM the buffer time would only be 30 minutes and if the monitor process were not executed until after 8:00PM there would be no buffer time at all.
  - **IMD Length**: Defines the number of hours, minutes and seconds of data that should be retrieved for each request. It is entered in the format of hours:minutes:seconds. This number must be divisible into 24 since it will determine how many requests will be made per day (i.e. if set to 8 hours (08:00:00) there will be 3 requests per day).
  - **Disable Extended IMD Length Recovery**: By default the schedule read activity will attempt to "catch-up" when interrogation requests have been missed by requesting the entire period from the last interrogation request made. Setting this parameter to "Yes" will turn off this functionality and interrogation requests will only be made in for the exact amount of time defined in the **IMD Length**

Copyright © 2026, Oracle and/or its affiliates.

- **Range of Recurrence:** indicates how to define request recurrences. Valid values include “Maximum Recurrence”, “No End Date”, and “Recurrence End Date”
  - **Maximum Recurrence**: The activity will make a set number of requests, once the set number has been reached the activity will expire. When selected, **Maximum Recurrences** must also be provided.
  - **No End Date**: The activity will never expire and will continue to make requests until it is manually transitioned to inactive.
  - **Recurrence End Date**: The activity will continue to make requests until a specified date. When selected, **Recurrence End Date Time** must also be provided.

> **Note**
>
> When “No End Data” is selected, the recurrences of the request continue until a user manually changes the status of the command.

- **Maximum Recurrence:** defines the maximum number of recurrences before the command’s status is changed to “Inactive”.
- **Recurrence End Date Time:** The date after which no more recurrences are attempted. After this date and time, the command’s status is changed to “Inactive”.

- **Group:** the group of devices used by requests for the command
  - **Group Type:** indicates the type of device group to use for the command. Valid options include Application Group or Configuration Group.
  - **Application Group/Configuration Group Name:** defines the specific group (based on the group type).

> **Note**
>
> Groups referenced by Scheduled Read commands and the devices that belong to each are defined in Itron OpenWay head-end system. Each group to be used with these commands must also be defined in either the “Itron — Application Group Lookup” or “Asset Specification” extendable lookup. See the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay Configuration Guide for more information about these extendable lookups.

- **Default Read Parameters:** specifies details concerning the data to be requested. The read parameters define the exact data that will be requested from the meters. Please consult the Itron OpenWay documentation for the specifics of each parameter.
  - **Interrogation Window Hours:** the number of hours used to calculate when to submit requests. For example, to submit requests once per day, this should be set to 24. To submit requests once per week (7 days), this should be set to 168. The current date time (or the Latest Request End Date Time) plus this equals interrogation window end date time.
- **Override Read Parameters**: an alternative set of parameters that can be defined. To leverage this group each parameter must be provided, no parameter is allowed to be left blank. This set of parameters will be used when the request end time is equal to the **First Daily Measurement Time**. For example, a schedule read with a **First Daily Measurement Time** of 12:00AM and an **IMD Length** 12 hours (12:00:00) would make the following requests each day:
  - 12:00AM - 12:00PM

Copyright © 2026, Oracle and/or its affiliates.

- 12:00PM - 12:00AM

  The **Override Read Parameters** would be used for the 2nd request where the request end time of 12:00AM matches the **First Daily Measurement Time** of 12:00AM. If an extended request is made to "catch up" (i.e. it makes a request that is larger in duration than the configured **IMD Length**), and the request start and request end date time includes the **First Daily Measurement Time** then the **Override Read Parameters** will be used.

- **Outbound Communication Exception Handling Overrides**: parameters used to control retry attempts and expiration for the outbound communication. These parameters take precedence over their outbound communication counterparts
- **Last Read Details:** details concerning the most recent read request sent for the command. This information is used to calculate the time for the next request (based on the **Interrogation Window Hours** parameter).
  - **Latest Request Start Date Time:** the start date and time of the last request
  - **Latest Request End Date Time:** the end date and time of the last request
  - **Recurrence Count:** the number of recurrences of the last request

About Ad Hoc Requests

While scheduled read commands are typically configured to create recurring requests for meter reads and device events, users can also submit ad hoc requests for a specific date and time range based on the parameters of a currently active scheduled read command. This is useful if meter read data or device events are needed before the next request for a command would be submitted based on the command's recurrence settings.

Ad hoc requests are created as separate one-time request commands, with a start and end time based on parameters supplied by the user when initiating the ad hoc request. The group and read parameters for ad hoc requests are the same as those for the command used to send the ad hoc request.

Error Handling

If errors occur when processing scheduled read commands, errors are logged as follows:

- If there is a single error it will create a single log entry
- If there is more than one error, the log entry will perform a count of devices and create a single error message stating the number of devices that were in error.

Initial Measurements

The usage data received from the AMI head-end system as a file in Itron OpenWay XML format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D8-ITRONXML-BASE** contains components responsible for "actual" processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to SGG-D8-ITRONXML-CM.
2. **SGG-D8-ITRONXML-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D8-ITRONXML-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify "true" for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
populateRaw	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
destinationRootElementEvent	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false
filterUsage	Determines if usage should be filtered.	true false
filterRegisterSource	Determines if source register data should be filtered.	true false

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D8-ITRONXML-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Copyright © 2026, Oracle and/or its affiliates.

Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Itron - UOM Code to Standard UOM Mapping** extendable lookup (D8-HeadendUOMLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

Itron OpenWay can send real-time events such as meter outage alarms to Oracle Utilities Smart Grid Gateway. These events are referred to as exceptions in the Itron architecture, and are sent to the BPEL ExceptionSubscriberService service by Itron OpenWay without requiring an explicit request from Oracle Utilities Smart Grid Gateway. Exceptions are loaded as device events data and provided to the edge applications.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D8-EXCEPTION-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to the SGG-D8-EXCEPTION-CM project.
2. **SGG-D8-EXCEPTION-CM** allows the customization and simplifies the future upgrades.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the device event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Publishing Events

The Itron OpenWay adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D8-EXCEPTION-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to

Copyright © 2026, Oracle and/or its affiliates.

consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Itron - Device Event Mapping** extendable lookup (D8-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D8-EXCEPTION-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The Itron OpenWay adapter base package includes the following initial measurement and device event business objects:

Business Object Name	Description
D8-InterrogateByGroup	Itron - Interrogate By Group Performs a Schedule Read for a group of devices (either Application or Configuration) in the Itron OpenWay API.

Copyright © 2026, Oracle and/or its affiliates.

Business Object Name	Description
D8-InterrogateByGroupResult	Itron - Interrogate by Group Result This BO is the inbound asynchronous response to an Interrogate By Group (Schedule Read) outbound communication.
D8-InitialLoadIMDInterval	Itron - Initial Load IMD - Interval Used when loading Itron OpenWay interval measurements into the system for the first time.
D8-InitialLoadIMDScalar	Itron - Initial Load IMD - Scalar Used when loading Itron OpenWay scalar measurements into the system for the first time.
D8-ScheduleRead	Itron - Schedule Read Requests IMD and Event data from the Itron OpenWay Network. This data will later be returned and placed in files to be uploaded.

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Itron OpenWay command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	Itron - Reconnect Meter	Itron - Reconnect Meter Result	Connect Device Completion Event
Remote Disconnect	Itron - Disconnect Meter	Itron - Disconnect Meter Result	Disconnect Device Completion Event
Device Commissioning	Itron - Add Meter Definition		Device Commissioning Completion Event

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Decommissioning	Itron - Deregister Meter		Device Decommissioning Completion Event
On-Demand Read (Scalar)	Itron - Contingency Read (Scalar)	Itron - Contingency Read Result (Scalar)	Create IMD Completion Event
On-Demand Read (Interval)	Itron - Contingency Read (Interval)	Itron - Contingency Read Result (Interval)	Create IMD Completion Event
Scheduled Read (Scalar)	Itron - Interrogate by Group (Scalar)	Itron - Interrogate by Group Result (Scalar)	Create IMD Completion Event
Scheduled Read (Interval)	Itron - Interrogate by Group (Interval)	Itron - Interrogate by Group Result (Interval)	Create IMD Completion Event
Device Status Check	Itron - Ping by Endpoints (Status Check)	Itron - Get Ping by Endpoints Result	

Device Communication Base Package Business Objects

The Itron OpenWay Adapter base package includes the following communication business objects:

Business Object Name	Description
D8-AddMeterDefinitions	Itron - Add Meter Definition (Commission)
D8-DeregisterMeter	Itron - Deregister Meter (Decommission)
D8-DetectLoadSideVoltageByMtr	Itron - Detect Load Side Voltage
D8-DetLoadSideVoltageMtrRslt	Itron - Detect Load Side Voltage Result
D8-DisconnectMeter	Itron - Disconnect Meter (Remote Disconnect)
D8-DisconnectMeterResult	Itron - Disconnect Meter Result
D8-PingByEndpoints	Itron - Ping By Endpoints (Device Status Check)
D8-PingByEndpointsMDResponse	Itron - Multi-Device Ping Response
D8-PingByEndpointsMultiDevice	Itron - Ping By Endpoints Multi-Device
D8-PingByEndpointsResponse	Itron - Ping By Endpoints Response
D8-ReadDisconStateByMtr	Itron - Read Disconnect State
D8-ReadDisconStateMtrRslt	Itron - Read Disconnect State Result
D8-ReadInterval	Itron - Contingency Read (Interval)
D8-ReadResult	Itron - Contingency Read Result
D8-ReadScalar	Itron - Contingency Read (Scalar)
D8-ReconnectMeter	Itron - Reconnect Meter (Remote Connect)
D8-ReconnectMeterResult	Itron - Reconnect Meter Result

Itron OpenWay Event Data Mapping

The Itron OpenWay event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

Copyright © 2026, Oracle and/or its affiliates.

Itron OpenWay Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

Outbound Message Type	Description
D8ITRECO	D8 Itron Remote Connect

Copyright © 2026, Oracle and/or its affiliates.

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Itron OpenWay includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.
D1-DeviceStatusCheck	Device Status Check This service is invoked by the integration layer to instantiate a Device Status Check command.
D1-InitialLoadIMD	Used by OSB to instantiate an IMD The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.
D1-RemoteConnect	Remote Connect This service is invoked by the integration layer to instantiate a Remote Connect command.
D1-RemoteDisconnect	Remote Disconnect This service is invoked by the integration layer to instantiate a Remote Disconnect command.
D8-DetLoadSideVoltageMtrRslt	Itron - Detect Load Side Voltage by Meter Result
D8-DisconnectMeterResult	Itron - Disconnect Meter Result Service
D8-InterrogateByGroupResult	Itron - Interrogate By Group Result XIA
D8-PingByEndpointsMDResponse	Itron - Ping By Endpoints Multi-Device Response XAI Inbound
D8-PingByEndpointsResponse	Itron - Ping By End Response inbound web service

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service	Description
D8-ReadDisconStateMtrRslt	Itron - Read Disconnect State by Meter Result Service
D8-ReadResult	Itron - Contingency Read Result
D8-ReconnectMeterResult	Itron - Reconnect Meter Result Service

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Itron OpenWay includes the following message senders:

Message Sender	Description
D8-RemoteCon	D8 Remote Connect

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to Itron OpenWay format, invoking process callouts and invoking the remote endpoint to trigger the device events. BPEL modules are divided into “Composites,” or collections of business logic.

OnDemandRead Composite Process — The OnDemandRead composite invokes a sequence of web methods that call the head-end system to retrieve meter reading data and send it back to the OUAF layer. In the case of Itron OpenWay, this is accomplished using the ContingencyReadByEndpoints/GetContingencyReadByEndpointsResult services defined in the Data service WSDL.

ConnectDisconnect Composite Process — This composite is responsible for triggering the Connect and Disconnect events on the head-end system. A second, asynchronous reply sends the results back into the OUAF layer when the head-end system signals a change in the status of the device. The Itron OpenWay Adapter uses the ReconnectMeter/GetReconnectMeterResult web service pair to connect, and the DisconnectMeter/GetDisconnectMeterResult web service pair to disconnect. Each of these services is defined in the Control service WSDL.

CommissionDecommission Composite Process — Commissioning and Decommissioning of devices, sometimes referred to as “Provisioning,” is handled by the CommissionDecommission composite. Commissioning invokes the AddMeterDefinitions web service and Decommissioning uses the DeregisterMeters web service. Both are defined in the Provisioning service.

DeviceStatusCheck Composite — This composite uses the PingByEndpoints and GetPingByEndpointsResult web services to check the health of a device. These Itron OpenWay services are defined in the Control.Diagnostic service.

Common Composite — The Common composite contains three main classes of operations: Proxies, ProcessCallouts, and utility functions. Proxies are usually simple mediators that forward a web service call to a preset endpoint. In this Adapter they have two additional roles. First, they interrogate a composite property and determine whether the supported version of head-end software is 3.70 or 3.90. Some of the proxies will also append the callback URL for

Copyright © 2026, Oracle and/or its affiliates.

the StatusChanged service. Proxies are convenient because they allow head-end URLs and security to be set in a single composite. In this case, they also offer a common location to make these checks. ProcessCallouts are points of customization which allow users to modify data and/or initiate some external business process. Utility functions serve as a central location for business logic needed by multiple composites.

Web Services

The following web services are all defined in the Itron OpenWay head-end system:

- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** AddMeterDefinitions
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/CommissionDecommission/CommissionDecommissionService
- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** DeregisterMeters
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/CommissionDecommission/CommissionDecommissionService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** ReconnectMeter
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/ConnectDisconnect/ConnectDisconnectService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** DisconnectMeter
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/ConnectDisconnect/ConnectDisconnectService
- **DeviceStatusCheckService**
  - **BPEL Process:** DeviceStatusCheck
  - **Operation:** DeviceStatusCheck
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/DeviceStatusCheck/DeviceStatusCheckService
- **OnDemandReadService**
  - **BPEL Process:** OnDemandRead
  - **Operation:** ContingencyReadByEndpoints
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/OnDemandRead/OnDemandReadService

Itron OpenWay Web Services

The following table describes Itron OpenWay web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Copyright © 2026, Oracle and/or its affiliates.

Smart Grid Gateway Command	AMI Adapter Business Objects	Itron OpenWay Web Services	Itron OpenWay Operations
Device Commissioning	D8-AddMeterDefinitions	ProvisioningService	AddMeterDefinitions
Device Decommissioning	D8-DeregisterMeter	ProvisioningService	DeregisterMeters
Remote Connect	D8-ReconnectMeter	ControlService	ReconnectMeter
	D8-DetectLoadSideVoltageByMtr	DiagnosticService	DetectLoadSideVoltageByMeter
	D8-ReadDisconStateByMtr		ReadDisconnectStateByMeters
Remote Disconnect	D8-DisconnectMeter	ControlService	DisconnectMeter
Device Status Check	D8-PingByEndpoints	ControlService	PingByEndpoints
			GetPingByEndpointsResult
On-Demand Read	D8-ReadInterval	DataService	ContingencyReadByEndpoints
	D8-ReadScalar		GetContingencyReadByEndpointsResult
	D1-InitialLoadIMD		InterrogateByGroup
	D1-DeviceEventSeeder		GetInterrogateByGroupResult
		DataSubscriberService (Async Response)	DataArrived
Event Loading	D1-DeviceEventSeeder	ExceptionSubscriberService	ExceptionsArrived

Configuring an Itron OpenWay Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay to communicate with the Itron OpenWay application.

- Master Configurations
- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Master Configurations

Master Configurations are sources of global parameter records used by a system implementation. This section describes the master configuration that is specific to Oracle

Copyright © 2026, Oracle and/or its affiliates.

Utilities Smart Grid Gateway Adapter for Itron OpenWay. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about other master configurations used by Oracle Utilities Smart Grid Gateway.

Itron Version Master Configuration

This master configuration specifies the version of the Itron head-end system with which the system is communicating. The configuration options are Itron OpenWay 3.70, Itron OpenWay 3.90, and Itron OpenWay 6.10. Only one version can be specified at a time.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Itron OpenWay application in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	Used by OSB to instantiate an IMD
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D8-DetLoadSideVoltageMtrRslt	Itron - Detect Load Side Voltage by Meter Result
D8-DisconnectMeterResult	Disconnect Meter Result Service
D8-InterrogateByGroupResult	Itron - Interrogate By Group Result
D8-PingByEndpointsMDResponse	Itron - Ping By Endpoints Multi-Device Response XAI Inbound
D8-PingByEndpointsResponse	Itron - Ping By Endpoints Response inbound web service
D8-ReadDisconStateMtrRslt	Itron - Read Disconnect State by Meter Result Service
D8-ReadResult	Itron - Contingency Read Result
D8-ReconnectMeterResult	Itron - Reconnect Meter Result Service

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name

Copyright © 2026, Oracle and/or its affiliates.

- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests. A message sender should be configured for each command.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
D8-ITRON_IBG	Itron Interrogate By Group
D8-COMM	Itron Commission
D8-DCOMM	Itron Decommission
D8-DLSV	Itron Detect Load Side Voltage
D8-IPBE	Itron Ping By Endpoints
D8-RCONN	Itron Remote Connect
D8-RDCONN	Itron Remote Disconnect
D8-RDSS	Itron Read Disconnect State

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/Itron/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Itron/**<SERVICE>**
- where:
  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed

Copyright © 2026, Oracle and/or its affiliates.

- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
- **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

How to Use Enterprise Manager to Locate the URL for the Message Sender

Follow this procedure to find the correct URL for the command associated with a message sender:

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the service used by the message sender:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the service:
3. Click the service’s WSDL URL link to see the WSDL in the browser, or right click and save it to your machine.

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D8-CONREAD	Itron Contingency Read By Endpoints
D8-INTRGRP	Itron Interrogate By Group
D8-COMM	Itron Commission
D8-DCOMM	Itron Decommission
D8-DSC	Itron Device Status Check
D8-DSCMD	Itron Device Status Check Multi Device
D8-DLSV	Itron Detect Load Side Voltage
D8-RCONN	Itron Remote Connect
D8-RDCONN	Itron Remote Disconnect
D8-RDSS	Itron Read Disconnect State

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Itron OpenWay head end must be present in your system. If this is not present in your configuration, add it, along with the following Outbound

Copyright © 2026, Oracle and/or its affiliates.

Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - Itron OpenWay:

- **External System**: Itron OpenWay
- **Description**: Itron OpenWay
- **Outbound Message Types**:

Outbound Message Type	Message Sender
D8-INTRGRP	D8-ITRON_IBG
D8-COMM	D8-COMM
D8-DCOMM	D8-DCOMM
D8-DSC	Message sender associated with the Device Status Check Outbound Message Type
D8-DSCMD	Message sender associated with the Device Status Check Multi Device Outbound Message Type
D8-DLSV	D8-DLSV
D8-RCONN	D8-RCONN
D8-RDCONN	D8-RDCONN
D8-RDSS	D8-RDSS

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D8-Request xsl
- **Response XSL**: D8-Response.xsl

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the Itron OpenWay head end must be present in your system. If this are not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - Itron OpenWay:

- **Service Provider**: Itron OpenWay
- **Description**: Itron OpenWay
- **External Reference ID**: Itron OpenWay
- **External System**: Itron OpenWay
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number

Copyright © 2026, Oracle and/or its affiliates.

- **Pre-Commissioning Device ID Type**:
- **AMI Measuring Component ID Type**: Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Itron OpenWay service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Itron OpenWay service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D8-AddMeterDefinitions	Commission
Device Decommission	Device Removal	D8-DeregisterMeter	Decommission
Device Status Check	Device Status Check	D8-PingByEndpoints	Device Status Check

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Default Business Object	Default Outbound Message Type
Load Check*	Load Check	D8-DetectLoadSideVoltageByMtr	Detect Load Side Voltage by Meter
Multi-Devie Status Check	Multi-Devie Status Check	D8-PingByEndpointsMultiDevice	Read Disconnect State By Meters
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D8-ReadScalar	On Demand Read — Scalar
Remote Connect	Remote Connect	D8-ReconnectMeter	Connect
Remote Disconnect	Remote Disconnect	D8-DisconnectMeter	Disconnect

* The Load Check processing method can be used to ensure that it is safe to connect a meter. The load check processing method is executed during the Connection Ready state of the Remote Connect activity and can be configured to detect the possibility of a load side voltage at the meter, or to check the disconnect meter switch on the meter.

If the Read Disconnect State business object is used as the processing method, you should configure the disconnect switch state values that indicate the state of the switch. These values are defined on the D8-DisconnectSwitchStateLookup extendable lookup, and are then added in the Disconnect Switch State Error Values section of the Itron - Read Disconnect State Result Inbound communication type. See Itron OpenWay Disconnect Switch State Lookup for more information.

Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Itron OpenWay endpoint override DVM (D8–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Itron OpenWay adapter and the command used with each:

DVM Key	Commands
DataService390	Scheduled Read (Scalar) Scheduled Read (Interval) On-Demand Read (Scalar) On-Demand Read (Interval)
ProvisioningService370	Device Commissioning
ProvisioningService390	Device Decommissioning

Copyright © 2026, Oracle and/or its affiliates.

DVM Key	Commands
ControlService370	Device Status Check (if using v3.70) Remote Connect (if using v3.70) Remote Disconnect (if using v3.70)
ControlService390	Remote Connect (if using v3.90) Remote Disconnect (if using v3.90)
DiagnosticService390	Device Status Check (if using v3.90)
ProcessCallout	User Exit Functions

> **Note**
>
> The numbers in the keys above (370 and 390) designate the version of the Itron OpenWay head-end system.
>
> Only a single version of each key should be defined, based on the version of the Itron OpenWay head-end system

To define an override Endpoint URI for the Itron OpenWay adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.
2. Select the D8–EndpointOverrides.dvm in the left panel.
   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.
3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.
5. Enter the appropriate values in the **Add Domain Values** dialog as follows:
   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.
6. Click **OK**.

   The new DVM value will appear in the list of keys.
7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

Copyright © 2026, Oracle and/or its affiliates.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Itron OpenWay Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Itron OpenWay.

This section outlines some of the extendable lookups that must be configured for use with the Itron OpenWay adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Application Group Lookup

The Itron OpenWay head-end system allows users to define addressable groups of meters based on criteria such as a geographic identifier or zip code. These application groups can be represented in Oracle Utilities Smart Grid Gateway by using the Itron OpenWay Application Group Lookup. The groups defined for this lookup are used with Scheduled Read commands. Each value defined for the Itron OpenWay Application Group extendable lookup should include the following:

- **Application Group:** The application group name as defined in the Itron OpenWay head-end system.
- **Description:** A description of the application group.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

Disconnect Switch State Lookup

Some utilities may require verification of the load on a meter to ensure that it is safe to connect the meter. One method of doing this is to check the disconnect switch on the meter to see if it is connected or armed. To use this method, the Itron Openway Disconnect Switch State extendable lookup should be configured to list the possible states of the meter disconnect switch. Each value defined for the Itron OpenWay Interval Disconnect Switch State extendable lookup should include the following:

- **Switch State**: The Itron OpenWay disconnect switch state code
- **Description**: A description of the disconnect switch state code.

Copyright © 2026, Oracle and/or its affiliates.

UOM Code to Standard UOM Mapping

Usage received from a utility's head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the Itron OpenWay application
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Unit of Measure**: The unit of measure defined in the system.
- **Time of Use**: An optional time of use period, defined in the system, used to further distinguish the unit of measure.
- **Service Quantity Identifier**: An optional service quantity identifier, defined in the system, used to further distinguish the unit of measure.
- **Unit of Measure Magnitude Conversion**:
  - **Magnitude Multiplier**: An optional; multiplier used to convert raw values received from the head-end system to values appropriate for use with the system. For example, if a reading or interval data is received in Wh, a multipler of ".001" would convert Wh to kWh. If not provided or left blank, no conversion is performed.

Interval Status Code to Condition Mapping

Interval usage received from the Itron OpenWay application can include Itron OpenWay interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Itron OpenWay Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Itron OpenWay interval status codes to standard status codes when receiving usage from the Itron OpenWay application. Each value defined for the Itron OpenWay Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Itron OpenWay interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

Other Extendable Lookups

Business Object Name	Description
D8-DvcStatusMappingLookup	Itron - Device Status Lookup
D8-FailureReasonLookup	Itron - Failure Reason Lookup
D8-ItronVersions	Itron - Versions Lookup
D8-JobStatusLookup	Itron - Job Status
D8-ResultLookup	Itron - Result Lookup
D8-RptSelfReadOptionLookup	Itron - Report Self Read Option Lookup

Copyright © 2026, Oracle and/or its affiliates.

Using the Itron OpenWay Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

Landis+Gyr

The Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr supports communication with the Landis+Gyr Gridstream Command Center, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, and on-demand read. The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Version	Gridstream Command Center 7.5, 8.1, 8.4, 8.5, 9.0.4
Smart Meter Command Format	MultiSpeak 3.0 CIM 1.ov2 CIM 2.0
Bulk Usage/Event Data Format	California Metering Exchange Protocol (CMEP)
Market(s)	North America, portions of Asia Pacific, Sweden, Australia, New Zealand, and Latin America.
Commodities Supported	Electricity, Gas, Water
Architecture	PLC and RF

- Landis+Gyr Adapter Processing
- Configuring a Landis+Gyr Head-End System
- Configuring Landis+Gyr Extendable Lookups
- Using the Landis+Gyr Test Harness
- Landis+Gyr Interval Data Mapping

Landis+Gyr Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service

Copyright © 2026, Oracle and/or its affiliates.

format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

Initial Measurements

The usage data exported from the AMI head-end system as a file in Landis+Gyr format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D3-USAGE-BASE** - contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This can be upgraded without affecting the customization and environment settings added to SGG-D3-USAGE-CM.
2. **SGG-D3-USAGE-CM** allows for customization and simplifies future upgrades.

When importing non-interval usage data, separate initial measurements can be created for difference measurement types. For instance, if the data includes Power Factor or Volt data, separate initial measurements are created for each of these. See Non-Interval 'Plain' XML to IMD Mapping for more information about specific units of measure that trigger the creation of separate initial measurements.

The runtime configuration settings for the SGG-D3-USAGE-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRawIMD	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
modifyResultXMLInput	Specifies the name of an XQuery document (without the "xq" extension) used to map additional fields from the "plain" XML format to the result XML format sent as initial measurement data. See Mapping Additional Fields for more information.	
dateTimeInUTC	Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterUsage	Determines if usage should be filtered.	true false

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D3-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes

Copyright © 2026, Oracle and/or its affiliates.

normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```


Copyright © 2026, Oracle and/or its affiliates.

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Landis+Gyr UOM Code to Standard UOM Mapping** extendable lookup (D3-HeadendUOMLookup) are passed into the system for processing.

> **Note**
>
> Filtering of scalar initial measurement data is not supported in the Landis+Gyr adapter.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

The device event data exported from the head-end system as a file in Landis+Gyr format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing. The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D3-EVENT-BASE** containing components responsible for "actual" processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D3-EVENT-CM project.
2. **SGG-D3-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D3-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element. The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the device event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
modifyResultXMLInput	Specifies the name of an XQuery document (without the "xq" extension) used to map additional fields from the "plain" XML format to the result XML format sent as device event data. See Mapping Additional Fields for more information.	
dateTimeInUTC	Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Publishing Events

SGG can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D3-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Landis+Gyr Device Event Mapping** extendable lookup (D3-DeviceEventMappingLookup) are passed into the system for processing.

Copyright © 2026, Oracle and/or its affiliates.

Subscribing to Real-Time Device Events

The Landis+Gyr Command Center provides the ability to "subscribe" to device events from within their online interface. This is done by creating a subscriber in Command Center with an associated callback URL as well as a list of events types that subscriber is interested to receive. When an event that is subscribed to occurs it is sent to the callback URL in real time.

Within L+G Events are configured with one of three "alarm" settings. These settings determine how often the events will be sent to subscribers:

1. Alarm: immediately delivered from the meter
2. Advisory: sent based upon a delivery schedule
3. Log Only: sent only upon request (not applicable for our implementation real time event processing)

Command Center will communicate the events using a CIM format that describes the message as a noun/verb combination. The details of the event itself will be contained within a "payload" element of the standard structure. The payload will be formatted using the EndDeviceEvent message structure. This message identifies device events using a CIM 4-part category number. These numbers are four period separated numbers that will describe the type of device and the event. For example: 3.33.1.257 is for "Tamper attempt suspected".

- Segment 1: End Device event domain code (e.g. 3. meter/10. collector/11. router/12. HAN device)
- Segment 2: End Device Event Domain Part Codes (e.g. 1. Access/2. Battery)
- Segment 3: End Device Event Type Codes (e.g. 1. Alarm/2. Alarm Mgt)
- Segment 4: End Device Event Index (e.g. 1. Abort/2. Access Attempt)

Refer to the Landis+Gyr documentation for details about the CIM Category Numbers. CIM Category Numbers must be mapped to standard device event names using the Landis+Gyr Device Event Mapping extendable lookup.

SGG receives these messages through a BPEL composite that saves the incoming request as a file to be picked up by OSB.

The **AMIEventSubscriber** composite is responsible for receiving the event messages based on subscriptions defined in the L+G Command Center. The callback URL configured for the subscription in the Command Center should point to this BPEL composite.

The following OSB projects parse individual device events from the message and perform the validation and mapping of the information to the Device Event Seeder Format.

1. **SGG-D3-CIM-EVENT-BASE** contains components responsible for "actual" processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D3-CIM-EVENT-CM project.
2. **SGG-D3-CIM-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D3-CIM-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
populateRaw	Determines if the device event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Processing statistics are gathered for any real time events that are received (even if there is just one event in the message) in the same manner as device events received via the flat-file interface.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D3-CIM-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.

Copyright © 2026, Oracle and/or its affiliates.

3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** colulmn.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The Landis+Gyr adapter base package includes the following initial measurement business objects:

Business Object Name	Description
D3-InitialLoadIMDInterval	Landis+Gyr Initial Load IMD - Interval Used when loading Landis+Gyr interval measurements into the system for the first time.
D3-InitialLoadIMDScalar	Landis+Gyr Initial Load IMD - Scalar

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Landis+Gyr command:

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect (This command has sub-commands)	Initiate MR by Mtr Num Initiate Connect Disconnect	Reading Changed Notification Connect Disconnect State Changed Notification	Connect Device Create IMD Completion Event
Remote Disconnect (This command has sub-commands)	Initiate Connect Disconnect Initiate MR by Mtr Num	Connect Disconnect State Changed Notification Reading Changed Notification	Disconnect Device Create IMD Completion Event
Device Commissioning (Registration)	L+G Add Meter to Inventory		Device Commissioning
Device Commissioning (Installation)	L+G Meter Exchange Notification		Device Commissioning
Device Decommissioning	Meter Remove Notification		Device Decommissioning
Device Deregistration	L+G Meter Retire Notification		Device Deregistration
On-Demand Read (Scalar)	Initiate MR by Mtr Num	Reading Changed Notification	Create IMD Completion Event
On-Demand Read (Scalar) - CIM	CIM Meter On Demand Read (Scalar)	CIM Meter On Demand Read Response	Create IMD Completion Event
On-Demand Read (Interval)	Initiate MR by Mtr Num	Reading Changed Notification	Create IMD Completion Event
On-Demand Read (Interval) - CIM	CIM Meter On Demand Read (Interval)	CIM Meter On Demand Read Response	Create IMD Completion Event
Device Status Check	CIM Ping	CIM Ping Response	
Demand Reset	Schedule Demand Reset (Multispeak)	Schedule Demand Reset Response (Multispeak)	Create IMD Completion Event

Device Registration Commission Commands

Landis+Gyr device commission commands can be used to “register” the device and notify the L+G head-end system that meters have been added to inventory. Device commission commands of this type have the “Registration-Only Mode” flag set to “Yes”. An Enter algorithm on the “Commission Ready” state evaluates the Registration-Only Mode of the command and if set to “Yes”, the command skips the default “Waiting for Measurement” state and is transitioned to the “Execute Completion Event” state, and an activity log entry is created.

Only the device registration request is sent to the head-end system for device commission commands of this type.

Device registration commands are typically created when new devices are added to inventory in an asset management system such as Oracle Utilities Operational Device Management.

Device Installation Commission Commands

Landis+Gyr device commission commands can be used to notify the L+G head-end system that meters have been installed or exchanged. An Enter algorithm on the “Commission Ready”

Copyright © 2026, Oracle and/or its affiliates.

state evaluates the “Is Installation Check Unnecessary” flag of the command and if set to “False”, the algorithm creates an “L+G Meter Exchange Notification” outbound communication and sends an installation notification to the head-end system.

Device Deregistration Commands

The Landis+Gyr adapter supports the Device Deregistration command (based on the D1-DeviceDeregistration business object). This command sends a communication that deregisters the device in the head-end system, and is most often used when retiring a device. The specific message sent is defined for the Device Deregistration processing role for the L+G head-end system service provider.

Device deegistration commands are typically created when devices are retired in an asset management system such as Oracle Utilities Operational Device Management.

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CCOMPEV (Create IMD Completion Event) algorithm on the Create Completion Events status of the D3-ReadingChgNotification (Reading Changed Notification - SOA based) business object.

The D3-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the D3-ReadingChgNotification (Reading Changed Notification - SOA based) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Demand Reset Commands

For on-demand read commands associated with Demand Reset commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-SDRCCE (Create Schedule Demand Reset Completion Event) algorithm on the Create Completion Events status of the D3-ScheduleDemandResetResponse (Schedule Demand Response(Multispeak)) business object.

The D3-VALSDRCCE (Check if Device has any Filtering errors - Schedule Demand Reset) algorithm on the Validate status of the D3-ScheduleDemandResetResponse (Schedule Demand Response(Multispeak)) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-SDRCCE algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

CIM Commands

For CIM-based on-demand read commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CCMPEVCIM (Create CIM Completion Event) algorithm on the Create Completion Events status of the D3-CIMMeterReadingsResponse (CIM Meter On Demand Read Response SOA Based) business object.

The D3-VALCEVCIM (Check if Device has any Filtering errors CIM Completion Event) algorithm on the Validate status of the D3-CIMMeterReadingsResponse (CIM Meter On Demand Read Response SOA Based) business object is used to check for errors related to

Copyright © 2026, Oracle and/or its affiliates.

filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CCMPEVCIM algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Device Communication Base Package Business Objects

The Landis+Gyr Adapter base package includes the following communication business objects:

Business Object Name	Description	Protocol
D3-AddMeterToInventoryMultiSp	L+G Add Meter to Inventory (MultiSpeak)	MultiSpeak
D3-CIMGetLPData	CIM Meter On Demand Read (Interval)	CIM
D3-CIMMeterOnDemandRead	CIM Meter On Demand Read (Scalar)	CIM
D3-CIMMeterReadingResponse	CIM Meter On Demand Read Response	CIM
D3-CIMPing	CIM Ping	CIM
D3-CIMPingResponse	CIM Ping Response	CIM
D3-ConnectDisconStateChgNtf	Connect Disconnect State Changed Notification	MultiSpeak
D3-InitiateConnectDisconnect	Initiate Connect Disconnect	MultiSpeak
D3-InitiateMRByMtrNbr	Initiate Meter Read By Meter (MultiSpeak)	MultiSpeak
D3-MeterAddNotificationMultiSp	Meter Add Notification (MultiSpeak)	MultiSpeak
D3-MeterExNotificationMultiSp	L+G Meter Exchange Notification	MultiSpeak
D3-MeterRetireNotification	L+G Meter Retire Notification	MultiSpeak
D3-MtrRmvNotifMultiSpeak	Meter Remove Notification (MultiSpeak)	MultiSpeak
D3-ReadingChgNotification	Reading Changed Notification	MultiSpeak
D3-ScheduleDemandReset	Schedule Demand Reset (Multispeak)	MultiSpeak
D3-ScheduleDemandResetResponse	Schedule Demand Reset Response (Multispeak)	MultiSpeak

Landis+Gyr Event Data Mapping

The Landis+Gyr event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

Landis+Gyr Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	

Copyright © 2026, Oracle and/or its affiliates.

Landis+Gyr Flat File Field	Device Event Seeder BO Element	Comments
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted. These notifications are based on the following outbound message types.

Outbound Message Type	Description
D3-ADDMTRINV	Add Meter to Inventory
D3-COMMS	Commission Device
D3-CONNECT	Connect Device
D3-DECOMMS	Decommission
D3-DEMRESET	Demand Reset
D3-DERDEV	Deregister Device

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Type	Description
D3-DISCONNEC	Disconnect Device
D3-DVCSTSCHK	Device Status Check
D3-INITMRN	Initiate Meter Read by Meter Number
D3-INITMTR	Initiate Meter Read by Meter Number
D3-MTRADDNOT	Meter Add Notification Outbound Message Type
D3-MTREX	Meter Exchange Notification OB MSG
D3-MTRRMV	Meter Remove Notification

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Landis+Gyr includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.
D1-InitialLoadIMD	Used for initial measurement upload. The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.
D3-CIMMeterReadingsResponse	CIM Meter On Demand Read Response Retrieve response from CIM On Demand Read command

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service	Description
D3-CIMPingResponse	CIM Ping Response Retrieve response from CIM Device Status Check command
D3-ConDisconStChgNotification	Initiate Connect Disconnect response. Retrieve response from the Initiate Connect Disconnect command.
D3-ReadingChangedNotification	Reading Changed Notification Notification that a Landis+Gyr device reading has changed.
D3-ScheduleDemandResetResponse	Scheduled Demand Reset Response Retrieve response from Demand Reset command

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Landis+Gyr includes the following message senders:

Message Sender	Description
D3-Comms	Commission Device
D3-Connect	Connect Device
D3-Decomm	Decommission Device
D3-Decomms	Decommissioning Sender
D3-DemReset	Demand Reset
D3-DerDevice	Deregister Device
D3-Disconnec	Disconnect Device
D3-InitMTR	Initiate Meter Read by Meter Number Outbound Message
D3-MTREXMS	Meter Exchange Notification Message Sender
D3-RTSender	Real Time Sender
D3-RTSnd	Real-time Sender (Landis+Gyr)
D3-SDemReset	SG Demand Reset

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 3.0 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

OnDemandRead Composite Process: Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

ConnectDisconnect Composite Process: Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

Copyright © 2026, Oracle and/or its affiliates.

CommissionDecommission Composite Process: Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, one of the following second business callout services is invoked to determine if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceivedCallout
- isExecutingCommissionCompletedCallout
- isExecutingDecommissionReceivedCallout
- isExecutingDecommissionCompletedCallout
- isExecutingAddMeterToInventoryReceivedCallout
- isExecutingAddMeterToInventoryCompletedCallout
- isExecutingMeterExchangeNotificationReceivedCallout
- isExecutingMeterExchangeNotificationCompletedCallout

CIMOnDemandRead Composite Process: Invokes the remote endpoint to trigger the CIM on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

CIMDeviceStatusCheck Composite Process: This process is similar to CIM OndemandRead BPEL process. SGG uses the result of On Demand Read command to identify the status of the meter. If reads are successfully returned, then meter is running healthy otherwise it is considered as meter inactive/dead.

DemandReset Composite Process: Invokes the remote endpoint to trigger the demand reset event. An asynchronous reply responds to the OUAF layer when the reading arrives.

LGProcessCallout Composite: This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

Web Services

These web services are all defined in the Landis+Gyr head end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location.

Web Service	Related BPEL Process	Description
MR_CB	OnDemandRead CommissionDecommission DemandReset	This web service is defined by the Landis+Gyr head end system’s implementation of MR_Server. The WSDL defines the interface for requesting a meter reading from the head end system. The actual definition can be obtained from L&G or downloaded from multispeak.org. Build 3.0aa is appropriate if obtained from MultiSpeak. Default endpoint must be changed in configuration: http://demo.turtletech.com/Multispeak/webapi/MR_CB.asmx

Copyright © 2026, Oracle and/or its affiliates.

Web Service	Related BPEL Process	Description
CD_CB	ConnectDisconnect	This web service is defined by the Landis+Gyr implementation of CB_CD. The WSDL defines the interface for requesting a meter's connection or disconnection on the head end system. This web service defines the interface for reporting a connection or disconnection by the head end system. This web service is only invoked by the head end system; not OUAF. Only the CDStateChangedNotification web method is implemented in the composite. Default endpoint must be changed in configuration: http://demo.turtletech.com/Multispeak/webapi/CD_CB.asmx
CIMService	CIMOnDemandRead CIMDeviceStatusCheck	This web service is defined by the L+G head end's implementation of AMIRequest Server. The WSDL defines the interface for requesting a meter reading from the head end system. The actual definition should be obtained from L&G or downloaded from L&G SDK for CIM 2.0.
LGProcessCallout	OnDemandRead ConnectDisconnect CommissionDecommission	Imported from **LGProcessCallout Composite** Default endpoint must be changed in configuration: http://127.0.0.1:8000/soa-infra/services/default/LGProcessCallout/LGProcessCallout

Landis+Gyr Command Center Web Services

The following table describes the Land+Gyr Command Center web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Command	AMI Adapter Business Objects	Landis+Gyr Web Services	Landis+Gyr Operations
Device Commissioning	D3-MeterAddNotificationMultiSp	MR	MeterAddNotification
Device Decommissioning	D3-MtrRmvNotifMultiSpeak	MR	MeterRemoveNotification
Remote Connect/ Remote Disconnect	D3-InitiateConnectDisconnect	CD	InitiateConnectDisconnect
On-Demand Read	D3-InitiateMRByMtrNbr	MR	InitiateMeterReadByMeterNumber

Copyright © 2026, Oracle and/or its affiliates.

Smart Grid Gateway Command	AMI Adapter Business Objects	Landis+Gyr Web Services	Landis+Gyr Operations
On-Demand Read (CIM)	D3-CIMGetLPData D3-CIMMeterOnDemandRead	CIMService	ScheduleDemandRead
Demand Reset	D3-ScheduleDemandReset	MR	CIM

Configuring a Landis+Gyr Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr to communicate with the Landis+Gyr Command Center software.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the L+G Command Center in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D3-ConDisconStChgNotification	Initiate Connect Disconnect Response
D3-CIMMeterReadingsResponse*	CIM Meter On Demand Read Response
D3-CIMPingResponse	CIM Ping Respose
D3-ReadingChangedNotification*	Reading Changed Notification
D3-ScheduleDemandResetResponse	Scheduled Demand Reset Response

Copyright © 2026, Oracle and/or its affiliates.

*The Landis+Gyr adapter supports both MultiSpeak and CIM On Demand Read commands. You only need to configure the inbound service for the protocol you wish to use.

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
D3-Comms	Commission Device
D3-Connect	Connect Device
D3-Decomm	Decommission Device
D3-InitMTR	Initiate Meter Read by Meter Number Outbound Message
D3-RTSender	Real Time Sender
D3-RTSnd	Real-time Sender (Landis+Gyr)
D3_CIMODR	L+G CIM On Demand Read
D3_CIMPING	L+G CIM Ping
D3_SCHDEMRES	L+G Schedule Demand Reset

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction: http://xmlns.oracle.com/ouaf/multispeak_3.0/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**

Copyright © 2026, Oracle and/or its affiliates.

- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/D3/**<SERVICE>**/**<SERVICE>**

  where:

  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D3-ADDMTRINV	Add Meter to Inventory
D3-COMMS	Commission Device
D3-CONNECT	Connect Device
D3-DECOMMS	Decommission
D3-DEMRESET	Demand Reset
D3-DERDEV	Deregister Device
D3-DISCONNEC	Disconnect Device
D3-DVCSTSCHK	Device Status Check
D3-INITMRN	Initiate Meter Read by Meter Number
D3-INITMTR	Initiate Meter Read by Meter Number
D3-MTRADDNOT	Meter Add Notification Outbound Message Type
D3-MTREX	Meter Exchange Notification OB MSG
D3-MTRRMV	Meter Remove Notification

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

Copyright © 2026, Oracle and/or its affiliates.

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the L+G Command Center must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - Landis+Gyr:

- **External System**: LG
- **Description**: Landis+Gyr
- **Outbound Message Types**:

Outbound Message Type	Description	Message Sender
D3-ADDMTRINV	Add Meter to Inventory	Message sender associated with the Add Meter to Inventory Outbound Message Type
D3-COMMS	Commission Device	Message sender associated with the Commission Device Outbound Message Type
D3-CONNECT	Connect Device	Message sender associated with the Connect Device Outbound Message Type
D3-DECOMMS	Decommission	Message sender associated with the Decommission Device Outbound Message
D3-DEMRESET	Demand Reset	Message sender associated with the Demand Reset Outbound Message Type
D3-DERDEV	Deregister Device	Message sender associated with the Deregister Device Outbound Message Type
D3-DISCONNEC	Disconnect Device	Message sender associated with the Disconnect Device Outbound Message Type
D3-DVCSTSCHK	Device Status Check	Message sender associated with the Device Status Check Outbound Message Type
D3-INITMRN	Initiate Meter Read by Meter Number	Message sender associated with the Initiate Meter Read By Meter Number Outbound Message Type
D3-INITMTR	Initiate Meter Read by Meter Number	Message sender associated with the Initiate Meter Read By Meter Number Outbound Message Type
D3-MTRADDNOT	Meter Add Notification Outbound Message Type	Message sender associated with the Meter Add Notification Outbound Message Type
D3-MTREX	Meter Exchange Notification OB MSG	Message sender associated with the Meter Exchange Notification Outbound Message Type

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Type	Description	Message Sender
D3-MTRRMV	Meter Remove Notification	Message sender associated with the Meter Remove Notification Outbound Message Type

Note: The following apply to the above outbound message types*:

- **Processing Method**: Real-time
- **Message XSL**: D3-Request xsl (*use D3-DRRequest.xsl for the Demand Reset command)
- **Response XSL**: D3-Response.xsl (*use D3-DRResponse.xsl for the Demand Reset command)

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the L+G Command Center must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - Landis+Gyr:

- **Service Provider**: LG
- **Description**: Landis+Gyr
- **External Reference ID**: L+G
- **External System**: Landis+Gyr
- **Out Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the L+G service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

Copyright © 2026, Oracle and/or its affiliates.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the L+G service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D3-AddMeterToInventoryMultiSp	Add Meter to Inventory
Device Commission	Device Installation	D3-MeterExNotificationMultiSp	Meter Exchange Notification
Device Decommission	Device Removal	D3-MtrRmvNotifMultiSpeak	Decommission Meter Remove Notification
Device Deregistration	Device Deregistration	D3-MeterRetireNotification	Deregister Device
On-Demand Read (Scalar), Multispeak and CIM	On-Demand Read (Scalar)	MS: D3-InitiateMRByMtrNbr CIM: D3-CIMMeterOnDemandRead	MS: Initiate Meter Read by Meter Number CIM: CIM On Demand Read
On-Demand Read (Interval)	On-Demand Read (Interval)	D3-CIMGetLPData	CIM On Demand Read
Demand Reset	Demand Reset	D3-ScheduleDemandReset	Demand Reset
Device Status Check	Device Status Check	D3-CIMPing	Device Status Check

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Default Business Object	Default Outbound Message Type
Remote Connect	Remote Connect	D3-InitiateConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	D3-InitiateConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Landis+Gyr endpoint override DVM (D3–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Landis+Gyr adapter and the command used with each:

DVM Key	Commands
CIMService	On-Demand Read (Scalar) - CIM On-Demand Read (Interval) - CIM Device Status Check
CD_CB	Remote Connect Remote Disconnect
Metering	Device Commissioning (Registration)
MR_CB	Device Commissioning Device Decommissioning Device Commissioning (Installation) Device Deregistration On-Demand Read (Scalar) On-Demand Read (Interval)
LGProcessCallout	User Exit Functions

To define an override Endpoint URI for the Landis+Gyr adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D3–EndpointOverrides.dvm in the left panel.

Copyright © 2026, Oracle and/or its affiliates.

- In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
- In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list). The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon ("+") to add a new key.

   The **Add Domain Values** dialog opens.
5. Enter the appropriate values in the **Add Domain Values** dialog as follows:
   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.
6. Click **OK**.

   The new DVM value will appear in the list of keys.
7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Landis+Gyr Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Landis+Gyr.

This section outlines some of the extendable lookups that must be configured for use with the Landis+Gyr adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

CIM Response Status Extendable Lookup

The CIM Response Status extendable lookup is used to map descriptions to response status codes received from the L+G Command Center. Each value defined for the CIM Response Status extendable lookup should include the following:

- **Response Status**: The CIM status code for the response status
- **Description**: A description of the response status
- **Status**: The status of the lookup value (can be Active or Inactive)

Copyright © 2026, Oracle and/or its affiliates.

CIM Data Source Extendable Lookup

The CIM Data Source extendable lookup is used to map descriptions to data sources defined in the L+G Command Center. Each value defined for the CIM Data Source extendable lookup should include the following:

- **Data Source**: The CIM code for the data source
- **Description**: A description of the data source
- **Status** : The status of the lookup value (can be Active or Inactive)

L+G CIM 2.0 Header Verb Lookup

The L+G CIM 2.0 Header Verb Lookup (D3-CIM2HeaderVerbLookup) extendable lookup defines values for message type verbs used in CIM 2.0 messages. Refer to the **International Standard document** for detailed information about verbs. This extendable lookup is pre-populated with values from the International Standard document.

L+G CIM 2.0 - Measuring Period Lookup

The L+G CIM 2.0 - Measuring Period Lookup (D3-CIMMeasuringPeriodLookup) extendable lookup defines mappings between measuringPeriod codes and Interval Size values. The values are defined in the **International Standard document**.

L+G CIM 2.0 Reply Code Lookup

The L+G CIM 2.0 Reply Code Lookup (D3-CIMReplyCodeLookup) extendable lookup contains the enumeration of the values in the Reply / Error / Code element in replies from the head end system. The values are defined in the **International Standard document** . Values comprise two numbers separated by a dot. Example: `ReplyCode = <ReplyCode Category> “.” <ReplyCode Index>`

This extendable lookup is pre-populated from the International Standard document.

L+G CIM 2.0 Reply Result Lookup

The L+G CIM 2.0 Reply Result Lookup (D3-CIMResponseReplyRsltLookup) extendable lookup contains the enumeration of the values for the `Reply.Result` value returned from the head end system. This extendable lookup is pre-populated with values from the **International Standard document**.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

UOM Code to Standard UOM Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to

Copyright © 2026, Oracle and/or its affiliates.

Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

Interval Status Code to Condition Mapping

Interval usage received from the Landis+Gyr Command Center can include interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Landis+Gyr Interval Status Code to Condition Mapping extendable lookup is used for this purpose. Each value defined for the Landis+Gyr Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The interval status code
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.
- **Description**: A description of the interval status code.
- **Daylight Savings Time Effect for Interval**: Flag that indicates (yes or no) if the status code indicates data during Daylight Saving Time (DST). To ensure that all DST scenarios are handled appropriately, set this parameter to "Yes" for all status conditions that indicate the interval was during DST. This should include mapping an Interval Status of "10" to the "Regular" condition.

Using the Landis+Gyr Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr (LG) includes a test harness that can be configured to simulate the Landis+Gyr Gridstream Command Center head-end system for testing the two-way commands. The test harness is Multispeak 3.0 standard compliant and includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Middleware Test Harnesses for more information.

Landis+Gyr Interval Data Mapping

This section describes how data in Landis+Gyr import files is mapped to the XML document format created by OSB and sent to Oracle Utilities Smart Grid Gateway.

When Landis+Gyr data is processed, it is initially received in a tilda-separated file format, which is converted into a "plan" XML format before being converted into the "result" XML format which is sent to the IMD Seeder and/or Device Event Seeder inbound services.

- Non-Interval Usage with Additional Fields
- XML 'Plain' XML Format
- Non-Interval Usage to 'Plain' XML Mapping
- 'Plain' XML to IMD Mapping
- Non-Interval 'Plain' XML to IMD Mapping
- Mapping Additional Fields

Copyright © 2026, Oracle and/or its affiliates.

Non-Interval Usage with Additional Fields

The following is a sample file of non-interval usage that contains additional fields.

ID~PremiseID~ESIID~Provisioned~Meter~kWh~DateTime~Peak~PeakDateTime~Dmd~TouA~TouB~TouC~TouD~TouE~Volts~PF~PrevKWh~KWhRecv

EMED01~SL002~~~96968280~34315.000~08042010120000AM~2.66~03122009063000AM~~34315.000~0.000~0.000~0.000~0.000~~~34285.000~30

EMED01~SL001~~~96968285~33693.000~08042010120000AM~2.62~03122009061500AM~~33693.000~0.000~0.000~0.000~0.000~~~33653.000~40

This data is mapped to the "plain" XML format.

XML 'Plain' XML Format

The "Plain" XML contain elements to hold the extra fields (highlighted in bold).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
           targetNamespace=" http://xmlns.oracle.com/LandisGyrUsage"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MeterReads">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MeterRead">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Origin"/>
              <xs:element name="ServProvExtRefId"/>
              <xs:element name="RecordType">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="MEPMD01" />
                      <xs:enumeration value="EMED01" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              <xs:element name="RecordVersion" minOccurs="0">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="20080519" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              <xs:element name="TimeStamp" />
              <xs:element name="Premise" minOccurs="0" />
              <xs:element name="ESIID" minOccurs="0" />
              <xs:element name="Provisioned" minOccurs="0" />
              <xs:element name="MeterID" />
              <xs:element name="Purpose" minOccurs="0" />
```


Copyright © 2026, Oracle and/or its affiliates.

```
                <xs:element name="Comodity" minOccurs="0" />
                <xs:element name="Units" minOccurs="0" />
                <xs:element name="CalcConst" minOccurs="0"/>
                <xs:element name="Interval" minOccurs="0"/>
                <xs:element name="Count" minOccurs="0"/>
                <xs:element name="FirstIntervalDateTime" />
                <xs:element name="Data">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Row" maxOccurs="unbounded"
minOccurs="0">
                        <xs:complexType>
                          <xs:attribute name="v" />
                          <xs:attribute name="s" />
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="kWh" minOccurs="0" />
                <xs:element name="Peak" minOccurs="0" />
                <xs:element name="PeakDateTime" minOccurs="0" />
                <xs:element name="Dmd" minOccurs="0" />
                <xs:element name="TouA" minOccurs="0" />
                <xs:element name="TouB" minOccurs="0" />
                <xs:element name="TouD" minOccurs="0" />
                <xs:element name="TouC" minOccurs="0" />
                <xs:element name="TouE" minOccurs="0" />
                <xs:element name="Volts" minOccurs="0" />
                <xs:element name="PF" minOccurs="0" />
                <xs:element name="ExtraFields" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="ExtraField" maxOccurs="255"
minOccurs="0">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="FieldName" minOccurs="0"/>
                            <xs:element name="FieldValue" minOccurs="0"/>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="RawData"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
</xs:schema>
```

Non-Interval Usage to 'Plain' XML Mapping

Copyright © 2026, Oracle and/or its affiliates.

The following table shows the mapping between fields in incoming non-interval data and child elements of MeterReads/MeterRead element in the "Plain" XML format:

L+G Interval Structure Field	"Plain" XML element
ID	RecordType
Premise ID	Premise
ESIID	ESIID
Provisioned	Provisioned
Meter	MeterID
kWh	kWh
Date/Time of initial Read	FirstIntervalDateTime
Peak	Peak
Peak Date/Time	PeakDateTime
Dmd	Dmd
TouA	TouA
TouB	TouB
TouC	TouC
TouD	TouD
TouE	TouE
Volts	Volts
PF	PF
`<extraFieldName1>`	ExtraFields/ExtraField/FieldName with value of `<extraFieldName1>` ExtraFields/ExtraField/FieldValue with value in `<extraFieldName1>`
`<extraFieldName2>`	ExtraFields/ExtraField/FieldName with value of `<extraFieldName2>` ExtraFields/ExtraField/FieldValue with value in `<extraFieldName2>`
`<extraFieldName3>`	ExtraFields/ExtraField/FieldName with value of `<extraFieldName3>` ExtraFields/ExtraField/FieldValue with value in `<extraFieldName3>`
`<extraFieldNameN>`	ExtraFields/ExtraField/FieldName with value of `<extraFieldNameN>` ExtraFields/ExtraField/FieldValue with value in `<extraFieldNameN>`
RawData	A record content from incoming file.

'Plain' XML to IMD Mapping

The following table outlines how data from the "plain" XML format is mapped to the InitialLoadIMD format when the Landis+Gyr record type is set to "MEPMD01" (interval usage).

"Plain" XML element	InitialLoadIMD element	Note
Record Type	N/A	

Copyright © 2026, Oracle and/or its affiliates.

"Plain" XML element	InitialLoadIMD element	Note
RecordVersion	N/A	
Premise ID	N/A	
ESIID	N/A	
Provisioned	N/A	
Meter ID	dvcIdN	as is
Purpose	N/A	
Commodity	N/A	
Units	externalUOM	as is
CalcConst	mcm	as is
Interval	spi	Transform from DDHHMM to SPI
Count	N/A	
FirstIntervalDateTime	stDt	Convert to OUAF date/ time format
Data	msrs	Value -> msrs/mL/q
Status -> msrs/mL/sts/stsL/st		
N/A	imdType	'D1IL
Origin	externalId	the origin attribute in "Plain" XML (incoming file name)
N/A	serviceProviderExternalId	L+G
RawData	rawData	Vendor-specific "raw" data

Non-Interval 'Plain' XML to IMD Mapping

The following table outlines how non-interval data in the "plain" XML format is mapped to the InitialLoadIMD format when the Landis+Gyr record type is set to "EMED01" (non-interval usage).

"Plain" XML Element	InitialLoadIMD element	Notes
RecordType	N/A	
Premise	N/A	
ESIID	N/A	
Provisioned	N/A	
MeterID	dvcIdN	as is
kWh	enQty	as is
	uom	KWH
TimeStamp	enDt	Convert to OUAF date/time format This should be mapped to enDt. Relevant only to non-interval data.

Copyright © 2026, Oracle and/or its affiliates.

"Plain" XML Element	InitialLoadIMD element	Notes
Peak	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KW
PeakDateTime	enDt	Convert to OUAF date/time format
Dmd	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KW and 'Dmd' in "<mcIdN>
TouA	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate A' in "<mcIdN>
TouB	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH' and 'kWh Rate B' in "<mcIdN>
TouC	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate C' in "<mcIdN>
TouD	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate D' in "<mcIdN>
TouE	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate E' in "<mcIdN>
Volts	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with Volts
PF	enQty	as is A separate IMD Seeder will be created with <uom> in IMD Seeder populated with PF
N/A	imdType	D1IL
Origin	externalId	the origin element in "Plain" XML (incoming file name)
N/A	serviceProviderExternalId	L+G This is a constant. There is no such field in incoming structure.
RawData	rawData	Vendor-specific "raw" data When the "populateRawIMD" parameters in the EnvironmentSettings.xq file is set to true.

Mapping Additional Fields

Copyright © 2026, Oracle and/or its affiliates.

Measurement and device event files received from the Landis+Gyr head-end system can include additional fields containing data to be imported into Oracle Utilities Smart Grid Gateway. These additional fields must be mapped to elements within the XML document processed by OSB and sent to Smart Grid Gateway.

This mapping can be performed through use of a custom XQuery document, specified in the EnvironmentSettings.xq file via the "modifyResultXMLInput" parameter.

The following sample XQuery documents illustrate how additional fields can be mapped into the XML format sent to Smart Grid Gateway.

Sample XQuery — Initial Measurements

The following XQuery is an example that shows a transformation that passes in a root element with 3 children (the “result” XML, the “plain” XML, the environment settings) that returns a modified “result” XML For testing purposes, it changes the original value in the <enQty> “result” element and replaces it with a value from the “plain” XML depending on an environment setting variable. The <serviceProviderExternalId> value was also replaced by a hard-coded value.

```
declare namespace lan = "http://xmlns.oracle.com/LandisGyrUsage";
declare namespace xf = "http://tempuri.org/D3/lgimd";
declare namespace soap = "http://schemas.xmlsoap.org/soap/envelope/";
declare function xf:modifyResultXML($modifyResultXMLInput as element(*)) as
element(*){
<InitialLoadIMDList>
{
  for $InitLoadIMD in $ modifyResultXMLInput/InitialLoadIMDList/InitialLoadIMD
  return
    <InitialLoadIMD>
      <preVEE>
        <dvcIdN>{ data($InitLoadIMD/preVEE/dvcIdN) }</dvcIdN>
        <externalId>{ data($InitLoadIMD/preVEE/externalId)}</externalId>
        <uom>{ data($InitLoadIMD/preVEE/uom)}</uom>
        <mcIdN>{ data($InitLoadIMD/preVEE/mcIdN)}</mcIdN>
        <enDt>{ data($InitLoadIMD/preVEE/enDt)}</enDt>
        {
        if ($modifyResultXMLInput/EnvironmentSettings/test1="true")
        then <enQty>{ data($modifyResultXMLInput/lan:MeterReads/lan:MeterRead/
lan:ExtraFields/lan:ExtraField[lan:FieldName
= 'EF4']/lan:FieldValue) }</enQty>
         else <enQty>{ data($modifyResultXMLInput/lan:MeterReads/
lan:MeterRead/lan:ExtraFields/lan:ExtraField[lan:FieldName= 'EF2']/
lan:FieldValue) }</enQty>
         }
       <imdType>{ data($InitLoadIMD/preVEE/imdType) }</imdType>
      </preVEE>
      <serviceProviderExternalId>NewSPId</serviceProviderExternalId>
    </InitialLoadIMD>}</InitialLoadIMDList>
};
declare variable $modifyResultXMLInput as element(*)external;
xf:modifyResultXML($modifyResultXMLInput)
```

Sample XQuery — Device Events

The following XQuery is an example that shows a transformation that passes in a root element with 3 children (the "result" XML, the "plain" XML, and the environment settings) and returns a

Copyright © 2026, Oracle and/or its affiliates.

modified "result" XML For testing purposes, it changes the original value in the <externalCommunicationModuleIdentifier> "result" element and replaces it with a value from the "plain" XML depending on an environment setting variable. The <externalServiceLocationId> value is also replaced by a hard-coded value.

```
declare namespace lan = "http://xmlns.oracle.com/LandisGyrEvent";
declare namespace xf = "http://tempuri.org/D3/event";
declare namespace soap = "http://schemas.xmlsoap.org/soap/envelope/";
declare function xf:modifyResultXML($modifyResultXMLInput as element(*)) as
element(*){
  <DeviceEventSeeder>
    <externalSenderId>{data($modifyResultXMLInput/DeviceEventSeeder/
externalSenderId) }</externalSenderId>
    <deviceIdentifierNumber>{ data($modifyResultXMLInput/DeviceEventSeeder/
deviceIdentifierNumber) }</deviceIdentifierNumber>
    <externalEventName>{ data($modifyResultXMLInput/DeviceEventSeeder/
externalEventName) }</externalEventName>
    <eventDateTime>{ data($modifyResultXMLInput/DeviceEventSeeder/
eventDateTime) }</eventDateTime>
    <externalSourceIdentifier>{ data($modifyResultXMLInput/DeviceEventSeeder/
externalSourceIdentifier)
}</externalSourceIdentifier>
    <eventInformation>
      <externalEventCategory>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalEventCategory)
}</externalEventCategory>
      <externalEventSeverity>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalEventSeverity)
}</externalEventSeverity>
      <externalDeviceType>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalDeviceType)
}</externalDeviceType>
      <externalServiceLocationId>{1234 }</externalServiceLocationId>
      {
      if ($modifyResultXMLInput/EnvironmentSettings/testA="true")
      then
<externalCommunicationModuleIdentifier>{ data($modifyResultXMLInput/
lan:DeviceEvents/lan:DeviceEvent/lan:DeviceType)}</
externalCommunicationModuleIdentifier>
      else <externalCommunicationModuleIdentifier>{data($modifyResultXMLInput/
lan:DeviceEvents/lan:DeviceEvent/lan:CategoryId)}</
externalCommunicationModuleIdentifier>
      }
     <externalStatusValue>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalStatusValue)
}</externalStatusValue>
      <externalStatusDateTime>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalStatusDateTime)}</externalStatusDateTime>
    </eventInformation>
  </DeviceEventSeeder>
};
declare variable $modifyResultXMLInput as element(*) external;
xf:modifyResultXML($modifyResultXMLInput)
```


Copyright © 2026, Oracle and/or its affiliates.

MV90 for Itron

The Oracle Utilities Smart Grid Gateway MV-90 Adapter for Itron uses Oracle Service Bus (OSB) to facilitate communication between Oracle Utilities Smart Grid Gateway and MV-90. The following functionality is included:

Measurement Data Loading - Data parsing and transformation from MV-90 binary format into the Oracle Utilities Service and Measurement Data Foundation unified format.

Measurement Data Processing - Configurable mapping for MV-90 status codes to Oracle Utilities Service and Measurement Data Foundation standard values, along with configurable device event creation based on MV-90 status codes.

The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Version	SGG supports the .mv9 binary, mainframe data format.
Protocol	Binary file format.
Market(s)	Worldwide, but largely USA.
Architecture	AMR

- MV90 Adapter Processing
- Configuring an MV90 Head-End System
- Configuring MV90 Extendable Lookups

MV90 Adapter Processing

This section provides details concerning the OSB processing and OUAF objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data Loading

Initial Measurement Data Loading

The initial measurement data load and subsequent device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway service format. Payloads contain measurements in a head-end specific format OSB then places each service call into a JMS queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel then a service creates initial measurements with data in a common format with head-end-specific processing as needed.

- Initial Measurements
- Base Package Business Objects

Initial Measurements

The usage data exported from the AMI head-end system as a file in MV-90 format is loaded into Oracle Utilities as Initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

Copyright © 2026, Oracle and/or its affiliates.

1. **SGG-D5-USAGE-BASE** contains components responsible for the actual processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to the SGG-D5-USAGE-CM project.
2. **SGG-D5-USAGE-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D5-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRawIMD	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterUsage	Determines if usage should be filtered.	true false
usePhysicalChannel	Optional value. Determines whether the physical channel ID is passed to the IMD seeder to create the measuring component identifier number. If this is set to false, the MV90 LOGCHAN field is used.	true false (default)
fieldForDvcIdN	Optional value. Specifies which field is used as the value for the device ID.	DC_RECID DC_CUSTID DC_METERID (default)
MV90ScalarChannelSuffix	Optional value. Holds a suffix value to be added to the measuring component identifier number when a scalar IMD is created for register reads. Default value is “_S”	
processMV90ScalarData	Optional value. Determines if register reads are processed.	true false (default)

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
usePulseMultiplierAndOffset	Optional value. If set to "true", specifies the pulse values are multiplied by the pulse mutlipler. **Note** This setting only applies to MV90 files that have a Pulse datatype.	true false (default)

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D5-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Usage

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```


Copyright © 2026, Oracle and/or its affiliates.

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **MV90 UOM Code to Standard UOM Mapping** extendable lookup (D5-UOMtoStdUOMCodeMapLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Initial Measurement Data Load with Register Reads

In the MV90 format, measurement data can include interval data as well as register reads that can be used to create scalar measurements. For information on configuring the adapter for loading scalar data based on register reads, see Override Processing Methods for Scalar Initial Measurement Data.

Base Package Business Objects

The MV-90 Adapter for Itron base package includes the following initial measurement business objects:

Business Object Name	Description
D5-InitialLoadIMDInterval	MV-90 Initial Load IMD
D5-InitialLoadIMDScalar	MV90 - Initial Load IMD - Scalar

Configuring an MV90 Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway MV-90 Adapter for Itron to receive usage from an MV-90 system.

- Inbound Web Services
- External System
- Service Provider
- Processing Methods

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including incoming usage.

Copyright © 2026, Oracle and/or its affiliates.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the MV-90 system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - MV-90:

External System: MV90

Description: MV90 - Mainframe File Format

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the MV-90 system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about creating service providers.

Copyright © 2026, Oracle and/or its affiliates.

Service Provider - MV-90:

- **Service Provider**: MV90
- **Description**: Head-End System for MV90
- **External Reference ID**: MV90
- **External System**: MV90 - Mainframe File Format
- **Out Name/ID in Their System**: MV90
- **AMI Device ID Type**: Serial Number
- **AMI Measuring Component ID Type**:

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the MV90 service provider. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Override Processing Methods for Scalar Initial Measurement Data

In the MV-90 format, meter read start and stop values can represent either a calculated read based on the interval data that is included in the MV90 data, or a register read that has been logged using a separate channel. When the values represent a register read the application can create a second file for the scalar initial measurements. The creation of a scalar initial measurement file is triggered by the Register Type flag, DC_REGTYPE, in the MV-90 data and the processMV90ScalarData variable in the EnvironmentSettings.xq file. The following register read values are valid for the DC_REGTYPE field:

Value	Description
V	Visual energy readings
D	Visual demand and energy readings
E	Encoder register readings (energy only).
B	Both encoder and visual energy readings.
C	Calculate stop meter readings from the encoder base reading

In order to configure your adapter to create scalar initial measurements based on register reads, first define new measuring component types and register measuring components. Then use the new measuring component types to specify override processing methods on the Initial Measurement Creation processing method for your service provider.

Copyright © 2026, Oracle and/or its affiliates.

Measuring Component Types for Scalar Reads: To create initial measurement data for register reads, first create new measuring component types to represent the registers. These will be used to create measuring components for scalar reads.

Example Measuring Component Type

- **Measuring Component Type:** MV90_SCALAR_REG
- **Description:** MV90 Scalar Register
- **Measuring Component Business Object:** Register
- **Measurement Business Object:** Measurement
- **Service Type:** Electric Service
- **Allow Negative Consumption:** Allowed
- **Consumptive/Subtractive:** Subtractive
- **Read Method:** Automatic Read

Other attributes such as Value Identifiers, VEE Groups, and so on, should be defined based on requirements

Register Measuring Components for Scalar Reads: Use the new measuring component types to create measuring components for the scalar measurements created by the register read process. You must create a corresponding register measuring component for each interval channel for which you will receive scalar reads. The channel number for the registers must be based on the channel numbers of the corresponding interval channels, plus the suffix defined in the “MV90ScalarChannelSuffix” environment setting. The default value for this setting is “_S”. These registers must also be added to the device configuration used by the interval channels (in other words, both measuring components - interval and scalar - must be on the same device). The following table shows examples of these values for the measuring components:

Device Type/Serial Number	MV-90 Interval Channel Number	MV-90 Scalar Register Channel Number
MV90 Electric/00001	1	1_S
MV90 Electric/00002	1	1_S
MV90 Electric/00003	1	1_S

Override Processing Methods for MV-90 Service Provider: Add an override processing method to the Initial Measurement Creation processing method for the MV90 service provider for each register measuring component type. The business object for each override processing method should be “D5-InitialLoadIMDScalar”. The following tables shows examples of the override processing methods:

Measuring Component Type	Business Object
MV90 Scalar Register	D5-InitialLoadIMDScalar
Measuring Component Type 2	D5-InitialLoadIMDScalar
Measuring Component Type n	D5-InitialLoadIMDScalar

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses

Copyright © 2026, Oracle and/or its affiliates.

this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

UOM Translation

UOM translation processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map MV90 UOM codes to standard UOM codes when receiving usage from the MV-90 system.

Configuring MV90 Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway MV90 Adapter for Itron.

This section outlines some of the extendable lookups that must be configured for use with the MV-90 Adapter for Itron. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Interval Status Code to Event Mapping

The MV-90 Adapter for Itron does not accept device events from an MV-90 system, but can create device events based on specific interval status codes in an incoming usage reading. The MV-90 Interval Status Code to Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when creating device events based on interval status codes received with usage from the MV-90 system.

Each value defined for the Interval Status Code to Event Mapping extendable lookup should include the following:

- **MV90 Interval Status**: The interval status code used by the MV-90 system
- **Description**: A description of the MV-90 interval status code
- **Event Duration Mode**: The duration type for the event. Can be either “Individual” or “Continuous” (used for events with a duration, such as an outage).
- **Device Event Type**: The Device Event Type for the device event created for this interval status code. For status codes with an Event Duration Mode of “Continuous” this is the start event, or the first of the paired events created for this status code.
- **End Event Type**: The Device Event Type for the “end” device event created for this interval status code. For status codes with an Event Duration Mode of “Continuous” this is the end event, or the last of the paired events created for this status code.

Example: The MV-90 “Low Voltage” status code could be configured to create a “Low Voltage” device event as follows:

- **MV90 Interval Status**: <MV-90 interval status code for low voltage>
- **Description**: Low Voltage detected for meter
- **Event Duration Mode**: Individual
- **Device Event Type**: Low Voltage
- **End Event Type**: N/A

Copyright © 2026, Oracle and/or its affiliates.

Example: The MV-90 “Power Outage” status code could be configured to create a “Last Gasp” device event, and a “Power Restoration” event as follows:

- **MV90 Interval Status**: <MV-90 interval status code for outage>
- **Description**: Outage
- **Event Duration Mode**: Continuous
- **Device Event Type**: Last Gasp
- **End Event Type**: Power Restoration

UOM Code to Standard UOM Code Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

Other Extendable Lookups

Business Object Name	Description
D5 - ChanelStatusCodeLookup	MV-90 Channel Status Code Lookup
D5-IntStsCdToCndMapPrLookup	Condition Mapping with Priority

Networked Energy Services

The Oracle Utilities Smart Grid Gateway Adapter for Networked Energy Services (NES) supports two-way communication with the NES server. Communications include measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, and on-demand read. The adapter uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language Process Manager (BPEL PM) to facilitate communication between Oracle Utilities Smart Grid Gateway and NES. The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Version	Networked Energy Services 5.3
Protocol	Proprietary (NES specific)
Market(s)	Worldwide
Architecture	ANSI and IEC, PLC

- Networked Energy Services Adapter Processing
- Configuring a Networked Energy Services Head-End System
- Configuring Networked Energy Services Extendable Lookups
- Configuring NES Usage and Event Extract Processing

Copyright © 2026, Oracle and/or its affiliates.

Networked Energy Services Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

Initial Measurements

Usage data is exported from the head-end system as a file in NES format and is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D4-USAGE-BASE** contains components responsible for the actual processing of incoming data. This should not be modified during configuration.
2. **SGG-D4-USAGE-CM** allows you to segment your customizations so that future upgrades of base functionality implemented in SGG-D4-USAGE-BASE do not affecting the customization and environment settings.

The runtime configuration settings for the SGG-D4-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterUsage	Determines if usage should be filtered.	true false

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D4-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

Copyright © 2026, Oracle and/or its affiliates.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **NES UOMCode Mapping Extendable Lookup** extendable lookup (D4-HeadendUOMLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property

Copyright © 2026, Oracle and/or its affiliates.

specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

The device event data exported from the head-end system as a file in NES format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D4-EVENT-BASE** contains components responsible for the actual processing of incoming data. This should not be modified during configuration.
2. **SGG-D4-EVENT-CM** allows you to segment your customizations so that future upgrades of base functionality implemented in SGG-D4-EVENT-BASE do not affecting the customization and environment settings.

The runtime configuration settings for the SGG-D4-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the device event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
serviceProviderExternalId	The External ID of the NES service provider.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Copyright © 2026, Oracle and/or its affiliates.

Publishing Events

The NES adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D4-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description	Default Value
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **NES Device Event Mapping** extendable lookup (D4-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

Copyright © 2026, Oracle and/or its affiliates.

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D4-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The NES adapter base package includes the following device, initial measurement, and device event business objects:

Business Object Name	Description
D4-InitialLoadIMDInterval	NES Initial Load IMD Interval
D4-InitialLoadIMDScalar	NES Initial Load MD Scalar
D4-SmartMeter	Smart Meter - NES
D4-EventExtractScheduler	NES Event Extract Scheduler
D4-PayloadExtractScheduler	Payload Extract Scheduler
D4-UsageExtractScheduler	NES Usage Extract Scheduler

Device Communication

Copyright © 2026, Oracle and/or its affiliates.

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes and Web Services

Communication Flows

The table below lists the communications created for each NES command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	Retrieve Meter Information Read Load Status Read Billing Data On-Demand Connect Load	Read Load Status Notification Read Billing Data On-Demand Notification Connect Load Notification	Create IMD Connect Device
Remote Disconnect	Disconnect Load Read Billing Data On-Demand	Disconnect Load Notification Read Billing Data On-Demand Notification	Create IMD Disconnect Device
Device Commissioning	Add Meter Request Set ATM Configuration	Set ATM Configuration Notification	Device Commissioning
Device Decommissioning	Decommission Meter Request	Decommission Meter Notification	Device Decommissioning
On-Demand Read (Scalar)	Read Billing Data On-Demand	Read Billing Data On-Demand Notification	Create IMD
On-Demand Read (Interval)	Read Load Profile On Demand	Read Load Profile On-Demand Notification	Create IMD

Commissioning and Decommissioning Communications

Commissioning a previously decommissioned NES device requires a different process than commissioning a new device.

When you commission a device for the first time, a commissioning command is sent to BPEL to create the meter in NES using the DeviceManager.CreateMeter command. If Automatic Topology Management is selected then the DeviceManager.SetATMConfiguration command is also sent.

Copyright © 2026, Oracle and/or its affiliates.

When you decommission a device, BPEL sends the DeviceManager.Move command to NES for the device. In NES you need to confirm that the device has been moved to a non-DC tree. You then need to manually delete the device from the non-DC tree. To permanently decommission the device you must delete it by using the DeviceManager.Delete command from the NES interface.

To recommission a device that has been previously decommissioned, you need to log into NES and manually move the meter from the non-DC tree to the DC tree that it belonged to previously. This will not be possible if the device has been deleted from within NES.

Device Communication Base Package Business Objects

The NES base package includes the following communication business objects:

Meter Commissioning

Business Object	Description
D4-AddMeterRequest	Add Meter Request
D4-GenericAMIDeviceIdentifier	Generic AMI Device Identifier
D4-RetrieveMeterIdentifier	Retrieve Meter Identifier
D4-RetrieveMeterInfo	Retrieve Meter Information
D4-SetATMConfigNotification	Set ATM Configuration Notification
D4-SetATMConfiguration	Set ATM Configuration

Meter Decommissioning

Business Object	Description
D4-DecommMeterNotification	Decommission Meter Notification
D4-DecommMeterRequest	Decommission Meter Request

On-Demand Read

Business Object	Description
D4-ReadBillingData	Read Billing Data On-Demand
D4-ReadBillingDataNtf	Read Billing Data On-Demand Notification
D4-ReadLoadProfile	Read Load Profile On Demand
D4-ReadLoadProfileNtf	Read Load Profile On-Demand Notification
D4-ReadLoadStatus	Read Load Status
D4-ReadLoadStatusNtf	Read Load Status Notification

Remote Connect

Business Object	Description
D4-ConnectLoad	Connect Load
D4-ConnectLoadNtf	Connect Load Notification

Copyright © 2026, Oracle and/or its affiliates.

Remove Disconnect

Business Object	Description
D4-DisconnectLoad	Disconnect Load
D4-DisconnectLoadNtf	Disconnect Load Notification

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

Outbound Message Type	Description
D4-ADDMETREQ	Add Meter Request
D4-CONLOAD	Connect Load
D4-DECMETREQ	Decommission Meter Request
D4-DISCONLD	Disconnect Load
D4-HNLBATEVT	NES - Handle Batch Event
D4-HNLBATUSG	NES - Handle Batch Usage
D4-RDLOADST	Read Load Status
D4-READLP	Read Load Profile
D4-READOUT	Read Billing Data
D4-RETMETID	Retrieve Meter Identifier
D4-RETRIEVE	Retrieve Meter Information
D4-SETATMCFG	Set ATM Configuration

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound

Copyright © 2026, Oracle and/or its affiliates.

communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway Adapter for NES uses the following inbound web services to import usage and device events:

Inbound Web Service	Description
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.
D1-InitialLoadIMD	Used for initial measurement upload. The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.

The Oracle Utilities Smart Grid Gateway Adapter for NES includes the following inbound web services for receiving communications from the NES:

Inbound Web Service Name	Description
D4-ConnectLoadNotification	Connect Load Notification
D4-DecommMeterNotificationXAI	Meter Decommission Notification
D4-DisconnectLoadNotification	Disconnect Load Notification
D4-ReadBDNotificationXAI	Read Billing Data On-Demand Notification
D4-ReadingATMConfigNotification	Get Response Initiate Read By Meter Number
D4-ReadLoadStatusNotification	Read Load Status Notification
D4-ReadLPNotificationXAI	Read Load Profile On-Demand Notification

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway Adapter for NES includes the following message senders:

Message Sender	Description
D4-ADDMETREQ	Add Meter Request
D4-CONLOAD	Connect Load
D4-DECOMMIS	Device Decommissioning
D4-DISCONLD	Disconnect Load

Copyright © 2026, Oracle and/or its affiliates.

Message Sender	Description
D4-HNLBATEVT	NES Handle Batch Event
D4-HNLBATUSG	NES Handle Batch Usage
D4-RDLOADST	Read Load status
D4-READBILLD	NES On-Demand Read (Scalar)(2)
D4-READLP	Read Load Profile
D4-RETMETID	Retrieve Meter Identifier
D4-RETRIEVE	Retrieve Meter Information
D4-ReadBill	Read Billing Data
D4-SETATMCFG	Set ATM Configuration

BPEL Processes and Web Services

These processes are responsible for performing the conversion from Oracle Utilities format to NES format, invoking process callouts and invoking the remote endpoint to trigger the device events.

All web services receive XML from an incoming service then transform the call to a format which is recognized by the NES system.

CreateMeter: Receives the incoming XML from the message sender service, transforms it to the NES API parameters node, passes it to the DeviceManager.Create NES API web service method then collects XML from CreateMeter to be returned to Oracle Utilities Smart Grid Gateway.

Messages

CreateMeter RequestMessage: Contains the parameters which are formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.Create method.

CreateMeter ResponseMessage: Contains the Device ID that is used when the DeviceManager.Create method validates the security key and the XML parameters successfully.

Related Processes

Meter Commissioning

ConnectLoadRemoteConnect
Receives the incoming XML from the D4-CONLOAD (Connect Load) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

GrabBatchEvents
Retrieves the un-processed events from the event repository; transforms each batch to replace GUID identifiers with a readable form, and finally prepares a new file for OSB before deleting the batch of events. The scheduled call to this service can contain the preferred dates to query for or can accept the default which is the previous day usage.

Messages

GrabBatchEventsSoapIn

GrabBatchEventsSoapOut

GrabBatchUsage

Copyright © 2026, Oracle and/or its affiliates.

Retrieves the un-processed usage from the usage repository; transforms each batch to replace GUID identifiers with a readable form, and finally prepares a new file for OSB before deleting the batch of usage. The scheduled call to this service can contain the preferred dates to query for or can accept the default which is the previous day usage.

Messages

GrabBatchUsageSoapIn

GrabBatchUsageSoapOut

Related Processes

Connection

Event Management

Usage Processing

MeterDecommissioning
Receives the incoming XML from the message senderOutbound service and transforms to NES API parameters node to be passed to the DeviceManager.Move NES API web service method. The process then collects the return XML to be returned to Oracle Utilities Smart Grid Gateway.

Messages

MeterDecommissioningRequestMessage: This message is the input for the Move method. The method contains the parameters conveniently formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.Move method.

MeterDecommissioningResponseMessage: This message contains status stating that the DeviceManager.Create method validated the security key and the XML parameters are accepted successfully. If the status is SUCCEEDED, it also contains Tracking ID for the call, Device ID and StatusType ID.

Related Processes

Meter Decommissioning

PerformCommand
This web service facilitates most of the communication between Oracle Utilities Smart Grid Gateway and the NES system. If your implementation requires functionality that was not delivered as part of the base package this web service will play an integral part in managing those customizations.

PerformCommand executes a command which generates the events that advance the Oracle Utilities Smart Grid Gateway lifecycle then returns the output string to the caller. The output might be a tracking ID or information about a device.

Messages

PerformCommandSoapIn: The sXmlParameters input is prepared from within Oracle Utilities Smart Grid Gateway to execute primarily a group of five commands, these are Read Load Status, On-Demand Read, Connect Load, Disconnect Load and Set ATM Configuration.

PerformCommandSoapOut:

The return from PerformCommand is the xml from the DeviceManager.PerformCommand for Read Load Status, On-Demand Read, Connect Load, Disconnect Load and Set ATM Configuration.

ReceivePanoramixEvents

Copyright © 2026, Oracle and/or its affiliates.

This web service is actually never called by the implementation, but rather it is called by NES and is needed when configuring the NES server.

ReceivePanoramixEvents obtains the APIKey from the NES authentication service, then, with the event definition ID from the parameter list for this service, switches to the branch for the event definition to begin any specific processing. For example, an on-demand read completion event means that the BPEL PM process needs to retrieve the usage using the RESULTID found in the event xml. All events and usage to be passed on to Oracle Utilities Smart Grid Gateway must first be transformed to replace the GUID with a readable form items. This involves using the BPEL PM DVM lookup functionality.

Messages

ReceivePanoramixEventsSoapIn

ReceivePanoramixEventsSoapOut

Related Processes

Event Management

PrePerformCommandCallout / PostPerformCommandCallout
These web services are hooks designed to allow customization of messages just prior and just after sending them to the NES server. Your implementation may wish to modify the data for special purposes. This is the one of the only places where modification of BPEL PM code is allowed.

ReceivedPanoramixEventsCallout
This web service is a hook designed to allow customization of messages just prior and just after sending them to the NES server. Your implementation may wish to modify the data for special purposes. This is one of the only places where modification of BPEL PM code is allowed.

ReadBillingDataOnDemand
This web service method, initiate a call to NES system with formatted inbound xml parameter. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system's Device Manager Service. When the call is invoked the system receives the return XML from DeviceManager stating the status of the command execution and, if succeeded, the tracking ID of this call.

Messages

ReadBillingDataOnDemandRequestMessage: Contains the input parameter for DeviceManager Perform Command to read billing data OnDemand. It formats parameters from the calling system to match incoming XML for the NES DeviceManager.PerfomCommand method.

ReadBillingDataOnDemandResponseMessage: Transforms the response xml from NES system format to Oracle Utilities Smart Grid Gateway format.

Related Processes

On Demand Reads

ReadLoadProfileOnDemand
This web service method, initiate a call to NES system's Device Manager Service with formatted inbound xml parameter. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system thru Device Manager Service. After invoking the call, the system receives the response message from DeviceManager stating the status of the command execution and, if succeeded, the tracking ID of this call.

Copyright © 2026, Oracle and/or its affiliates.

Messages

ReadLoadProfileOnDemandRequestMessage: This message is for interval on demand read request to the NES system. Transforms the incoming XML from Oracle Utilities Smart Grid Gateway to NES recognizable format.

ReadLoadProfileOnDemandResponseMessage: Transforms the response xml for ReadLoadProfileOnDemandRequest from NES system format to Oracle Utilities Smart Grid Gateway format.

Related Processes

On Demand Reads

ReadLoadStatusRemoteConnect
Receives the incoming XML from the D4-RDLOADST (Read Load Status) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

RemoteDisconnect
Receives the incoming XML from the D4-DISCONLD (Disconnect Load) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

RetrieveMeterInfo
Initiates a call to the Request Management Service with formatted inbound xml parameters. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system thru Request Management Service and retrieves a response with Meter Information.

Messages

RetrieveMeterInformationRequestMessage: Contains the input parameter for DeviceManager Perform Command to retrieve meter information. It also formats parameters from the calling system to match incoming XML for the NES DeviceManager.PerfomCommand method.

RetrieveMeterInformationResponseMessage: Transforms the response xml from NES system format to Oracle Utilities Smart Grid Gateway format.

Related Processes

Remote Disconnect

Remote Connect

RetrieveMeterIdentifier
Receives a call from Oracle Utilities Smart Grid Gateway for RetrieveMeterIdentifier process and invokes the RetrieveByParameter method in the NES system's DeviceManager service.

Converts incoming Oracle Utilities Smart Grid Gateway formatted input to sXmlParameters in NES format then invokes RetrieveMeterIdentifier.

Messages

RetrieveMeterIdentifierRequestMessage: Transforms the xml received from the calling system to match incoming XML for the NES API DeviceManager.RetrieveByParameter method. This method requires the Type of ID and the associated ID as input.

RetrieveMeterIdentifierResponseMessage: When the XML is synchronously returned from the head-end system for the RetrieveByParameter call, the DEVICEID node is only included in the APIPAYLOAD if the status of the call is Succeeded.

Related Processes

Copyright © 2026, Oracle and/or its affiliates.

Meter Commissioning

SetATMConfiguration
This web service method is specific to set Automated Topology Management (ATM) configuration command which assigns the Meter to the Data Concentrator.

Converts incoming Oracle Utilities Smart Grid Gateway formatted input to sXmlParameters in NES format then invokes the DeviceManager PerformCommand. A connection should establish to begin the NES command processing.

Messages

SetATMConfigurationRequestMessage: This message is the input for the SetATMConfiguration method. The method contains the parameters conveniently formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.PerformCommand method where the COMMANID is Constants.DeviceCommands.SET_ATM_CONFIGURATION

SetATMConfigurationResponseMessage: when the XML is synchronously returned from the head-end system for the PerformCommand call, the COMMAND node is only included in the APIPAYLOAD if the status of the call is Succeeded.

Related Processes

Meter Commissioning

NES Web Services

The following table describes NES web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Commands	AMI Adapter Business Objects	NES Web Services	NES Operations
Device Commissioning	D4-AddMeterRequest D4-RetrieveMeterIdentifier D4-SetATMConfiguration	DeviceManager	Create RetrieveByParameter PerformCommand: SET_ATM_CONFIGURATION
Device Decommissioning	D4-DecommMeterRequest	DeviceManager	PerformCommand: MOVE_DEVICE_ADD
Remote Connect	D4-ConnectLoad D4-ReadLoadStatus D4-RetrieveMeterInfo	DeviceManager	Retrieve PerformCommand: CONNECT_LOAD READ_LOAD_STATUS
Remote Disconnect	D4-DisconnectLoad	DeviceManager	PerformCommand: DISCONNECT_LOAD
On-Demand Read	D4-ReadBillingData	DeviceManager	PerformCommand Command ID: READ_BILLING_DATA_ON_DEMAND (Scalar) READ_LOAD_PROFILE_ON_DEMAND (Interval)

Configuring a Networked Energy Services Head-End System

Copyright © 2026, Oracle and/or its affiliates.

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway NES Adapter to communicate with the NES software.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the NES in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D4-ConnectLoadNotification	Connect Load Notification
D4-DecommMeterNotificationXAI	Meter Decommission Notification
D4-DisconnectLoadNotification	Disconnect Load Notification
D4-ReadBDNotificationXAI	Read Billing Data On-Demand Notification
D4-ReadingATMConfigNotification	Get Response Initiate Meter Read By Meter Number
D4-ReadLoadStatusNotification	Read Load Status Notification
D4-ReadLPNotificationXAI	Read Load Profile On-Demand Notification

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

Copyright © 2026, Oracle and/or its affiliates.

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
D4-ADDMETREQ	Add Meter Request
D4-CONLOAD	Connect Load
D4-DECOMMIS	Device Decommissioning
D4-DISCONLD	Disconnect Load
D4-HNLBATEVT	NES Handle Batch Event
D4-HNLBATUSG	NES Handle Batch Usage
D4-RDLOADST	Read Load status
D4-READBILLD	NES On-Demand Read (Scalar)(2)
D4-READLP	Read Load Profile
D4-RETMETID	Retrieve Meter Identifier
D4-RETRIEVE	Retrieve Meter Information
D4-ReadBill	Read Billing Data
D4-SETATMCFG	Set ATM Configuration

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction: http://xmlns.oracle.com/ouaf/echelon/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/D4/**<SERVICE>**/**<SERVICE>**

  where:

Copyright © 2026, Oracle and/or its affiliates.

- **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
- **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
- **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
- **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
- **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D4-ADDMETREQ	Add Meter Request
D4-CONLOAD	Connect Load
D4-DECMETREQ	Decommission Meter Request
D4-DISCONLD	Disconnect Load
D4-HNLBATEVT	NES - Handle Batch Event
D4-HNLBATUSG	NES - Handle Batch Usage
D4-RDLOADST	Read Load Status
D4-READLP	Read Load Profile
D4-READOUT	Read Billing Data
D4-RETMETID	Retrieve Meter Identifier
D4-RETRIEVE	Retrieve Meter Information
D4-SETATMCFG	Set ATM Configuration

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the NES must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

Copyright © 2026, Oracle and/or its affiliates.

External System - NES:

- **External System**: NES
- **Description**: NES
- **Outbound Message Types**::

Outbound Message Type	Message Sender
D4-ADDMETREQ	D4-ADDMETREQ
D4-CONLOAD	D4-CONLOAD
D4-DECMETREQ	D4-DECOMMIS
D4-DISCONLD	D4-DISCONLD
D4-HNLBATEVT	D4-HNLBATEVT
D4-HNLBATUSG	D4-HNLBATUSG
D4-RDLOADST	D4-RDLOADST
D4-READLP	D4-READBILLD
D4-READOUT	D4-READLP
D4-RETMETID	D4-RETMETID
D4-RETRIEVE	D4-RETRIEVE
D4-SETATMCFG	D4-SETATMCFG

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D4-Request xsl
- **Response XSL**: D4-Response.xsl

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the NES must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - NES

- **Service Provider**: NES
- **Description**: NES
- **External Reference ID**: NES
- **External System**: D4-NES
- **Out Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

Copyright © 2026, Oracle and/or its affiliates.

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the NES service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the NES service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D4-SetATMConfiguration	Set ATM Configuration
Device Decommission	Device Removal	D4-DecommMeterRequest	Decommission Meter Request
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D4-ReadBillingData	Read Billing Data
On-Demand Read (Interval)	On-Demand Read (Interval)	D4-ReadLoadProfile	Read Load Profile

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Status Check	Device Status Check	D4-ReadLoadStatus	Read Load Status
Remote Connect	Remote Connect	D4-ConnectLoad	Connect Load
Remote Disconnect	Remote Disconnect	D4-DisconnectLoad	Disconnect Load

Other Processing Methods

This section outlines details for other processing methods used by the NES adapter.

Processing Role	Default Business Object	Default Outbound Message Type
NES Add Meter Request	D4-AddMeterRequest	Commission
NES Retrieve Meter Identifier	D4-RetrieveMeterIdentifier	Commission Retrieve Meter Identifier
NES Retrieve Meter Information	D4-RetrieveMeterInfo	Retrieve Meter Info

Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The NES endpoint override DVM (D4–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the NES adapter and the command used with each:

DVM Key	Commands
DeviceManager	Remote Connect Remote Disconnect Device Commissioning Device Decommissioning On-Demand Read (Scalar) On-Demand Read (Interval)
EventManager	On-Demand Read (Scalar) On-Demand Read (Interval)

Copyright © 2026, Oracle and/or its affiliates.

DVM Key	Commands
GatewayManager	Remote Connect Remote Disconnect Device Commissioning Device Decommissioning On-Demand Read (Scalar) On-Demand Read (Interval)
SettingManager	Remote Connect Remote Disconnect Device Commissioning Device Decommissioning On-Demand Read (Scalar) On-Demand Read (Interval)
UserManager	Remote Connect Remote Disconnect Device Commissioning Device Decommissioning On-Demand Read (Scalar) On-Demand Read (Interval)
ProcessCallout	User Exit Functions

To define an override Endpoint URI for the NES adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D4–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.

4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

   The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.

8. Click the **Publish** button to activate all the changes in the editing session.

Copyright © 2026, Oracle and/or its affiliates.

Enter an optional note for the session in the **Publish Session** dialog.

Changes take effect immediately upon publishing the session.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Networked Energy Services Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Networked Energy Services.

This section outlines some of the extendable lookups that must be configured for use with the NES adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

UOM Code to Standard UOM Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

Interval Status Code to Condition Mapping

Interval usage received from the NES can include NES interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The NES Interval Status Code to Condition Mapping extendable lookup is used to determine how to map NES interval status codes to standard status codes when receiving usage from the NES.

Each value defined for the NES Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The NES interval status code

Copyright © 2026, Oracle and/or its affiliates.

- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.
- **Description**: A description of the interval status code.
- **Priority**: The priority of the interval status code, represented as a numeric value where larger numbers correlate to higher priorities.

Other Extendable Lookups

Business Object	Description
D4-EntityTypeLookup	Entity Type
D4-ExtServiceReturnCodeLookup	External Service Return Code
D4-GatewayTypeLookup	Gateway Type
D4-IDTypeLookup	ID Type
D4-InformationReturnTypeLookup	Information Return Type
D4-LoadVoltageStatusTypeLookup	Load Voltage Status Type
D4-TaskPriorityLookup	Task Priority
D4-TimeZoneManagerLookup	Time Zone Manager

Configuring NES Usage and Event Extract Processing

Usage and events received from the NES server must be requested from Oracle Utilities Smart Grid Gateway. Extract requests for usage and events are triggered by a batch process, and sent to the NES via middleware. The NES compiles a batch of usage or events and sends it to BPEL services to create a file to be processed by Oracle Service Bus (OSB) services, it is then loaded as SGG events and/or usage.

This section describes the processing performed by the extract request process, and the configuration steps involved.

- Extract Request Processing
- Extract Request Configuration Steps
- Executing Extract Requests

Extract Request Processing

This section provides a detailed description of the extract request process.

- The “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) checks for “NES Usage Extract Scheduler” or “NES Event Extract Scheduler” activities that are in the “Active” state, and transitions these to the “Send Request” state. It also updates the “Latest Request Start Date/Time” and “Latest Request End Date/Time” fields on the activity based on the date and time of the request.
- When the activity enters the “Send Request” state, an Enter algorithm (either “Usage Extract Scheduler Send Request” (D4-USGSNDREQ) or “Event Extract Scheduler Send Request” (D4-EVTSNDREQ)) sends a request via an message sender to the middleware (via the GrabBatchUsage or GrabBatchEvent BPEL services), which in turn send the request to the NES server.
- After the request is sent, the “NES Usage Extract Scheduler” or “NES Event Extract Scheduler” activity to set back to the “Active” state.

Copyright © 2026, Oracle and/or its affiliates.

- The NES server calls the BPEL services “GrabBatchEvents” and “GrabBatchUsage.” BPEL then routes the incoming events or usage to a file system, which is used by OSB components to process the usage or events.

Extract Request Configuration Steps

This section outlines the objects that must be configured to support batch extracts of usage and events from an NES head-end system. To configure the NES Adapter to support sending extract requests for usage and events, you must do the following:

- Create message senders (one for usage, one for events) to send extract requests.
- Create Outbound Message Types (one for usage, one for events) for extract requests.
- Associate the message senders and Outbound Message Types to the External System used with the head-end system service provider
- Create “NES Usage Extract Scheduler Type” and “NES Event Extract Scheduler Type” activity types.
- Create “NES Usage Extract Scheduler” and “NES Event Extract Scheduler” activities.

The following sections provide details for configuring these object. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders, Outbound Message Types, and External Systems.

Note: Configuration of the OSB and BPEL services used by this process is done when installing and configuring the OSB and BPEL components of the NES Adapter.

Message Sender - NES Handle Batch Event

Create an message sender for event extracts as follows:

Main Tab:

- **Message Sender**: Enter a code for the batch event message sender (D4-HNLBATEVT)
- **Description**: NES Handle Batch Event
- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes messages via HTTP real-time)
- **Active**: True (checked)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/NES/GrabBatchEvents
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/NES/HandleBatchEvent/BatchEventHandler

  where:

  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager

Copyright © 2026, Oracle and/or its affiliates.

- **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed

Message Sender - NES Handle Batch Usage

Create an message sender for usage extracts as follows:

Main Tab:

- **Message Sender**: Enter a code for the batch usage message sender (D4-HNLBATUSG)
- **Description**: NES Handle Batch Usage
- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes messages via HTTP real-time)
- **Active**: True (checked)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/NES/GrabBatchUsage
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/NES/HandleBatchUsage/BatchUsageHandler

  where:

  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed

Outbound Message Type - NES Handle Batch Event

Create an Outbound Message Type for event extract requests as follows:

- **Outbound Message Type**: Enter a code for the batch event Outbound Message Type (D4-HNLBATEVT)
- **Description**: NES Handle Batch Event
- **Business Object**: D1-OutboundMessage
- **Priority**: - Priority 20 (or lower)

Outbound Message Type - NES Handle Batch Usage

Create an Outbound Message Type for usage extract requests as follows:

- **Outbound Message Type**: Select a code for batch usage Outbound Message Type (D4-HNLBATUSG)
- **Description**: NES Handle Batch Usage

Copyright © 2026, Oracle and/or its affiliates.

- **Business Object**: D1-OutboundMessage
- **Priority**: Priority 20 (or lower)

External Systems

Add the event and usage extract Outbound Message Types and message senders to the external system used for the NES head-end system as follows:

- **External System**: NES
- **Description**: NES
- **Outbound Message Types**:

Outbound Message Type	Message Sender
D4-HNLBATEVT	D4-HNLBATEVT
D4-HNLBATUSG	D4-HNLBATUSG

Note: The following apply to the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D4-Request xsl
- **Response XSL**: D4-Response.xsl

Activity Type - NES Event Extract Scheduler

Create an Activity Type for event extract requests as follows:

- **Activity Type**: Enter a code for the activity type (NES-EVENT-EXTRACT)
- **Description**: Enter a description for the activity type (NES Event Extract Scheduler Type)
- **Related Transaction BO**: NES Event Extract Scheduler
- **Activity Type Status**: Active
- **Outbound Message Type**: NES Handle Batch Event
- **External System**: The external system used for the NES head-end system
- **Exception Handling**:
  - **To Do Type**: Activity To Do Type
  - **To Do Role**: System Default Role
  - **Retry Frequency**: Select retry frequency in case if Activity reaches an error state (00:03:00)
  - **Maximum Retries**: Select max number of times the activity will be retried - since it reached the error state (1)
  - **Discard Reason**: Select from the dropdown if available.

Activity Type - NES Usage Extract Scheduler

Create an Activity Type for usage extract requests as follows:

- **Activity Type**: Enter a code for the activity type (NES-USAGE-EXTRACT)
- **Description**: Enter a description for the activity type (NES Usage Extract Scheduler Type)
- **Related Transaction BO**: NES Usage Extract Scheduler

Copyright © 2026, Oracle and/or its affiliates.

- **Activity Type Status**: Active
- **Outbound Message Type**: NES Handle Batch Usage
- **External System**: The external system used for the NES head-end system
- **Exception Handling**:
  - **To Do Type**: Activity To Do Type
  - **To Do Role**: System Default Role
  - **Retry Frequency**: Select retry frequency in case if Activity reaches an error state (00:03:00)
  - **Maximum Retries**: Select max number of times the activity will be retried - since it reached the error state (1)
  - **Discard Reason**: Select from the dropdown if available.

Activity - NES Event Extract Scheduler

Initiate an Activity for event extract requests as follows:

1. Select **Main**->**Communication**->**Activity+**
2. Select "NES Event Extract Scheduler Type" from the **Activity Type** drop-down list.
3. Enter the number of hours for each extract request in the **Number of Hours of Data Request** field. The default is 24.
4. *Optional*: Enter start and end dates and times for the first extract request in the **Current Request Start Date/Time** and **Current Request End Date/Time** fields.

   Note that these are optional fields. If not populated, the values are populated as follows:

   - **Current Request Start Date/Time**: taken from the last request stop date/time (which is populated by the “Event Extract Scheduler Send Request” algorithm (D4-EVTSNDREQ)
   - **Current Request End Date/Time**: the Current Request Start Date/Time plus the number of hours in the **Number of Hours of Data Request** field.

Activity - NES Usage Extract Scheduler

Initiate an Activity for usage extract requests as follows:

1. Select **Main**->**Communication**->**Activity+**
2. Select "NES Usage Extract Scheduler Type" from the **Activity Type** drop-down list.
3. Enter the number of hours for each extract request in the **Number of Hours of Data Request** field. The default is 24.
4. *Optional*: Enter start and end dates and times for the first extract request in the **Current Request Start Date/Time** and **Current Request End Date/Time** fields.

   Note that these are optional fields. If not populated, the values are populated as follows:

   - **Current Request Start Date/Time**: taken from the last request stop date/time (which is populated by the “Usage Extract Scheduler Send Request” algorithm (D4-USGSNDREQ)
   - **Current Request End Date/Time**: the Current Request Start Date/Time plus the number of hours in the **Number of Hours of Data Request** field.

Copyright © 2026, Oracle and/or its affiliates.

Executing Extract Requests

To schedule and execute usage and device event extract requests, set up the “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) to run at a frequency appropriate to when you wish to retrieve usage and device events from the NES. For example, to retrieve usage and device events on a daily basis, schedule the “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) to run once every 24 hours.

Sensus

The Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI supports communication with the Sensus Regional Network Interface (RNI), including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read. The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Version	Sensus RNI 4.4, 4.5.1, 4.10, 4.12, 4.14, 4.15, 4.16
Protocol	MultiSpeak 4.1
Market(s)	Worldwide
Architecture	Long range radio WAN (mesh)

- Sensus Adapter Processing
- Configuring a Sensus Head-End System
- Configuring Sensus Extendable Lookups
- Using the Sensus Test Harness

Sensus Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements

Copyright © 2026, Oracle and/or its affiliates.

- Device Events
- Base Package Business Objects

Initial Measurements

The usage data exported from the AMI head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as Initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D6-USAGE-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This can be upgraded without affecting the customization and environment settings added to SGG-D6-USAGE-CM.
2. **SGG-D6-USAGE-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D6-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRawIMD	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
dateTimeInUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterUsage	Determines if usage should be filtered.	true false
useExternalTOU	Indicates whether or not an externally reference TOU period should be used when processing measurement data.	true false

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
scalarExpanding	Defines the number of register reads to process in scalar payloads. When defined, Sensus payload processing uses this setting to ensure that all the scalar reads included in a payload will be processed. Default value is 2.	

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D6-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Sensus UOM Code to Standard UOM Mapping** extendable lookup (D6-HeadendUOMLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Copyright © 2026, Oracle and/or its affiliates.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

The device event data exported from the head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D6-EVENT-BASE** contains components responsible for “actual” processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D6-EVENT-CM project.
2. **SGG-D6-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D6-EVENT-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
dateTimeInUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Publishing Events

The Sensus adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Copyright © 2026, Oracle and/or its affiliates.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D6-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description	Default Value
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Sensus Device Event Mapping** extendable lookup (D6-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

Copyright © 2026, Oracle and/or its affiliates.

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D6-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The Sensus RNI adapter base package includes the following device and initial measurement business objects:

Business Object Name	Description
D6-InitialLoadIMDInterval	Sensus Initial Load IMD - Interval Used when loading Sensus interval measurements into the system for the first time.
D6-InitialLoadIMDScalar	Sensus Initial Load IMD - Scalar
D6-SmartMeter	Sensus Smart Meter

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows

Copyright © 2026, Oracle and/or its affiliates.

- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Sensus RNI command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	Sensus – Initiate Connect Disconnect	Sensus – Connect / Disconnect State Change	Connect Device Completion Event
Remote Disconnect	Sensus – Initiate Connect Disconnect	Sensus – Connect / Disconnect State Change	Disconnect Device Completion Event
Device Commissioning	Sensus – Meter Add Notification		Device Commissioning Completion Event
Device Decommissioning	Sensus – Meter Remove Notification		Device Decommissioning Completion Event
On-Demand Read (Scalar)	Sensus – Initiate Meter Read By Meter ID	Sensus – Reading Changed Notification	Create IMD Completion Event
On-Demand Read (Interval)	Interval data not supported		
Device Status Check	Sensus – Initiate Outage Detection	Sensus – Outage Detection Event Notification	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D6-CCOMPEV (Create IMD Completion Event) algorithm on the Create Completion Events status of the D6-ReadingChgNotification (Sensus - Reading Changed Notif SOA based) business object.

The D6-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the D6-ReadingChgNotification (Sensus - Reading Changed Notif SOA based) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D6-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Device Communication Base Package Business Objects

The Sensus RNI Adapter base package includes the following communication business objects:

Copyright © 2026, Oracle and/or its affiliates.

Business Object Name	Description
D6-ConnectDisconStateChgNtf	Sensus - Connect/Disconnect State Change
D6-InitiateConnectDisconnect	Sensus - Initiate Connect Disconnect
D6-InitiateMeterByMeterId	Sensus - Initiate Meter Read By Meter ID
D6-InitiateOutageDetection	Sensus - Initiate Outage Detection
D6-MeterAddNotification	Sensus - Meter Add Notification
D6-MeterRemoveNotifcation	Sensus - Meter Remove Notification
D6-OutageDetectEvtNotification	Sensus - Outage Detection Event Notification
D6-ReadingChgNotification	Sensus - Reading Changed Notification
D6-UnsolicitedEvtNotification	Sensus - Unsolicited Event Notification

Sensus Event Data Mapping

The Sensus event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

Sensus Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

Copyright © 2026, Oracle and/or its affiliates.

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgement and response messages are sent and received validating that commands have been transmitted.

Outbound Message Type	Description
D6-CONDISCON	Sensus Initiate Connect Disconnect
D6-INITMTR	Initiate Meter Read By Meter ID
D6-INTOUTDET	Initiate Outage Detection Request

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service	Description
D1-InitialLoadIMD	Used for initial measurement upload. The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.
D1-DeviceStatusCheck	Device Status Check This service is invoked by the integration layer to instantiate a Device Status Check command.
D1-InitialLoadIMD	Used by OSB to instantiate an IMD This inbound web service is used by OSB to instantiate an Initial Measurement Data for incoming interval usage in the Sensus format.
D1-RemoteConnect	Remote Connect This service is invoked by the integration layer to instantiate a Remote Connect command.
D1-RemoteDisconnect	Remote Disconnect This service is invoked by the integration layer to instantiate a Remote Disconnect command.
D6-ConDisconStChgNotification	Initiate Connect Disconnect response. Retrieve response from the Initiate Connect Disconnect command.
D6-OutageDetectionEventNotification	Initiate Outage Detection Response Retrieve response from the Initiate Outage Detection Event Notification command.
D6-ReadingChangedNotification	Reading Changed Notification Notification that a Sensus device reading has changed.
D6-UnsolicitedEventNotification	Unsolicited Event Response Retrieve unsolicited notifications when an event triggers an alarm on the meter.

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following message senders:

Message Sender	Description
D6-CONDISCON	Sensus Initiate Connect/Disconnect
D6-INTOUTDET	Initiate Outage Detection Request
D6-InitMID	Initiate Meter Read By Meter ID Outbound Message

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 4.1 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

Copyright © 2026, Oracle and/or its affiliates.

OnDemandRead Composite Process — Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

ConnectDisconnect Composite Process — Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

CommissionDecommission Composite Process — Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, a one of the following second business callout services is invoked to determined if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceivedCallout
- isExecutingCommissionCompletedCallout
- isExecutingDecommissionReceivedCallout
- isExecutingDecommissionCompletedCallout

DeviceStatusCheck Composite — Invokes the remote endpoint to trigger the initiate outage detection event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

ProcessCallout Composite — This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

Web Services

These web services are all defined in the Sensus RNI head end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location. These web services have HTTP security by default. You may need to modify the security as a part of your implementation.

Web Service	Related BPEL Process	Description
CB_ServerService	ConnectDisconnect	This web service defines the return interface, the means by which the status is returned to the calling system. This web service is only be invoked by the head end system, not OUAF. Only the CDStateChangeNotification web method is implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/ConnectDisconnect/CB_ServerService

Copyright © 2026, Oracle and/or its affiliates.

Web Service	Related BPEL Process	Description
CB_Server	OnDemandRead	This web service defines the return interface, the means by which the reading is returned to the calling system. This web service is only be invoked by the head end system, not OUAF. Only the ReadingChangedNotification web method is implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/OnDemandRead/CB_Server
OA_ServerService	DeviceStatusCheck	This web service defines the asynchronous return for InitateOutageDetectionEventRequest for solicited responses. It is also used for unsolicited alarms. This web service is only be invoked by the head end system, not OUAF. Only the ODEventNotification, PingURL, and GetMethods web methods are implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/DeviceStatusCheck/OA_ServerService

Sensus RNI Web Services

The following table describes the Sensus RNI web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Command	AMI Adapter Business Objects	Sensus Web Services	Sensus Operations
Device Commissioning	D6-MeterAddNotification	MR	MeterAddNotification
Device Decommissioning	D6-MeterRemoveNotification	MR	MeterRemoveNotification
Remote Connect/ Remote Disconnect	D6-InitiateConnectDisconnect	CD	InitiateConnectDisconnect
	D6-ConnectDisconStateChgNtf	CB	CDStatesChangedNotification (async reply)
Device Status Check	D6-InitiateOutageDetection	OD	InitiateOutageDetectionEventRequest
	D6-OutageDetectEvtNotification	OA	ODEventNotification (async reply)
On-Demand Read	D6-InitiateMeterByMeterId	MR	InitiateMeterReadingsByMeterID
	D6-ReadingChangedNotification	CB	ReadingChangedNotification (async reply)

Configuring a Sensus Head-End System

Copyright © 2026, Oracle and/or its affiliates.

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI to communicate with the Sensus RNI.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Sensus Regional Network Interface (RNI) in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-DeviceEventSeeder	Device Event Seeder
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D6-ConDisconStChgNotification	Initiate Connect Disconnect Response
D6-OutageDetectionEventNotification	Initiate Outage Detection Response
D6-ReadingChangedNotification	Reading Changed Notification
D6-UnsolicitedEventNotification	Unsolicited Event Response

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Copyright © 2026, Oracle and/or its affiliates.

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
D6-CONDISCON	Sensus Initiate Connect/Disconnect
D6-INTOUTDET	Initiate Outage Detection Request
D6-InitMID	Initiate Meter Read By Meter ID Outbound Message

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Sensus/**<SERVICE>**
- where:
  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

Copyright © 2026, Oracle and/or its affiliates.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D6-CONDISCON	Sensus Initiate Connect Disconnect
D6-INITMTR	Initiate Meter Read By Meter ID
D6-INTOUTDET	Initiate Outage Detection Request

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - Sensus:

- **External System**: Sensus
- **Description**: Sensus
- **Outbound Message Types**:

Outbound Message Type	Description	Message Sender
D6-CONDISCON	Sensus Initiate Connect Disconnect	D6-CONDISCON
D6-INITMTR	Initiate Meter Read By Meter ID	D6-InitMID
D6-INTOUTDET	Initiate Outage Detection Request	D6- INTOUTDET

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D6-Request xsl
- **Response XSL**: D6-Response.xsl

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

Copyright © 2026, Oracle and/or its affiliates.

A service provider that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about creating service providers.

Service Provider - Sensus:

- **Service Provider**: Sensus
- **Description**: Sensus
- **External Reference ID**: Sensus
- **External System**: Sensus
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Sensus service provider. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Sensus service provider, based on the requirements of each implementation using the “How to Create OB

Copyright © 2026, Oracle and/or its affiliates.

COMM/Send OB Message" processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D6-MeterAddNotification	Commission
Device Decommission	Device Removal	D6-MeterRemoveNotification	Decommission
Device Status Check	Device Status Check	D6-InitiateOutageDetection	Initiate Outage Detection Request
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D6-InitiateMeterByMeterId	Initiate Meter Read by Meter ID
Remote Connect	Remote Connect	D6-InitiateConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	D6-InitiateConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an "Endpoint Override" Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Sensus endpoint override DVM (D6–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Sensus adapter and the command used with each:

DVM Key	Commands
MR_Server	Device Commissioning Device Decommissioning On-Demand Read (Scalar)
CD_Server	Remote Connect Remote Disconnect
OD_Server	Device Status Check
ProcessCallout	User Exit Functions

To define an override Endpoint URI for the Sensus adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

Copyright © 2026, Oracle and/or its affiliates.

The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D6–EndpointOverrides.dvm in the left panel.
   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.
3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon ("+") to add a new key.

   The **Add Domain Values** dialog opens.
5. Enter the appropriate values in the **Add Domain Values** dialog as follows:
   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.
6. Click **OK**.

   The new DVM value will appear in the list of keys.
7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Sensus Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Sensus.

This section outlines some of the extendable lookups that must be configured for use with the Sensus adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event

Copyright © 2026, Oracle and/or its affiliates.

- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the "Standard Event Name" extendable lookup.

UOM Code to Standard UOM Mapping

Usage received from a utility's head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)

Interval Status Code to Condition Mapping

Interval usage received from the Sensus RNI can include Sensus interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Sensus Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Sensus interval status codes to standard status codes when receiving usage from the Sensus RNI.

Each value defined for the Sensus Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Sensus interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

Other Extendable Lookups

Business Object Name	Description
D6-CDReasonCodeLookup	Sensus Connect/Disconnect Reason Code
D6-LoadActionCodeLookup	Sensus Load Action Code
D6-OutageEventTypeLookup	Sensus Outage Event Type
D6-SensusTimeUnits	Sensus Time Units
D6-ServiceTypeMappingLookup	Sensus Service Type Mapping

Using the Sensus Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

Copyright © 2026, Oracle and/or its affiliates.

Silver Spring Networks

The Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks supports communication with the Silver Spring Networks UtilityIQ application, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read. The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Versions	UtilityIQ Version 4.4, 4.7, 4.10, 4.14, 5.0, 5.1, 5.2, 6.1
Protocol	Proprietary MultiSpeak 4.1
Market(s)	Worldwide
Architecture	RF WAN (mesh) based on Access Points

- Silver Spring Networks Adapter Processing
- Configuring a Silver Spring Networks Head-End System
- Configuring Silver Spring Networks Extendable Lookups
- Using the Silver Spring Networks Test Harness

Silver Spring Networks Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

Initial Measurements

Copyright © 2026, Oracle and/or its affiliates.

The usage data exported from the AMI head-end system as a file in Silver Spring Networks XML format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. SGG-D7-SSNXML-BASE contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to SGG-D7-SSNXML-CM.
2. SGG-D7-SSNXML-CM allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D7-SSNXML-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the initial measurement data is populated as raw data. Valid values are:	true false
callPreProcessing	Determines if the preprocessing proxy service is called. Valid values are:	true false
callPostProcessing	Determines if the postprocessing proxy service is called. Valid values are:	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
destinationRootElementEvent	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false
filterUsage	Determines if usage should be filtered.	true false

Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Copyright © 2026, Oracle and/or its affiliates.

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D7-SSNXML-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description	Default Value
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **SSN - UOM Code to Standard UOM Mapping** extendable lookup (D7-HeadendUOMLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

The device event data exported from the head-end system as a file in Silver Spring Networks CSV format is loaded into Oracle Utilities as a device event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

Copyright © 2026, Oracle and/or its affiliates.

1. SGG-D7-CSV-BASE contains components responsible for “actual” processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D7-CSV-CM project.
2. SGG-D7-CSV-CM allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D7-CSV-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valie Values
populateRaw	Determines if the device event data is populated as raw data. Valid values are:	true false
callPreProcessing	Determines if the preprocessing proxy service is called. Valid values are:	true false
callPostProcessing	Determines if the postprocessing proxy service is called. Valid values are:	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered. Valid values are:	true false

Publishing Events

The Silver Spring Networks adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D7-CSV-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Copyright © 2026, Oracle and/or its affiliates.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Copyright © 2026, Oracle and/or its affiliates.

Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **SSN - Device Event Mapping** extendable lookup (D7-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D7-CSV-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.

Copyright © 2026, Oracle and/or its affiliates.

10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The Silver Spring Networks adapter base package includes the following device and initial measurement business objects:

Business Object Name	Description
D7-InitialLoadIMDInterval	SSN - Initial Load IMD - Interval Used when loading Silver Spring Network (SSN) interval measurements into the system for the first time.
D7-InitialLoadIMDScalar	SSN - Initial Load IMD - Scalar Used when loading Silver Spring Network (SSN) scalar measurements into the system for the first time.
D7-SmartMeter	SSN — Smart Meter

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound/Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Silver Spring Networks command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	SSN–Connect or Disconnect	SSN – Connect or Disconnect Response Remote Provisioning Job (Get Status)	Connect Device Completion Event

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Disconnect	SSN–Connect or Disconnect	SSN – Connect or Disconnect Response	Disconnect Device Completion Event
Device Commissioning (see Note below)	SSN- Replace Location		Device Commissioning Completion Event
Device Decommissioning	SSN- Replace Device At Location (Decomm)		Device Decommissioning Completion Event
On-Demand Read (Scalar)	SSN – Add Meter Read Job (Scalar)	SSN – Meter Read Response (Scalar)	Create IMD Completion Event
On-Demand Read (Interval)	SSN – Add Meter Read Job (Interval)	SSN – Meter Read Response (Interval)	Create IMD Completion Event
Device Status Check	SSN – Add Ping job	SSN – Ping Job Response	

> **Note**
>
> Service points must be provisioned in the Silver Spring Networks (Itron AMM) system before the Device Commissioning command can be used.

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D7-CCOMPEV (Create Completion Event Scalar) algorithm on the Create Completion Events status of the D7-MeterReadResponseScalar (SSN - Meter Read Response (Scalar)) business object (for scalar on-demand read commands), and/or on the D7-CCOMPEVIN (Create Completion Event Interval) algorithm on the Create Completion Events status of the D7-MeterReadResponseInterval (SSN - Meter Read Response (Interval)) business object (for interval on-demand read commands).

When filtering is enabled, the “Filter if invalid external MC identifier” parameter on the D7-CCOMPEV and D7-CCOMPEVIN algorithms controls what should happen if a measuring component cannot be found using the MC Identifier Number sent from the head end system. When set to 'Yes' and no measuring component is found, the initial measurement will be filtered. When set to 'No' and no measuring component is found, the system will attempt to find the measuring component using the UOM and TOU. The value of the “Filter if invalid external MC identifier” on these algorithms must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

The D7-VALDVCFLS (SSN - Check if Device has any Filtering errors (Scalar)) algorithm on the Validate status of the D7-MeterReadResponseScalar (SSN - Meter Read Response (Scalar)) business object is used to check for errors related to filtering for scalar on demand read commands. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D7-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled). The value of the “Filter if invalid

Copyright © 2026, Oracle and/or its affiliates.

external MC identifier” on this algorithm must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

The D7-VALDVCFLI (SSN - Check if Device has any Filtering errors (Interval)) algorithm on the Validate status of the D7-MeterReadResponseInterval (SSN - Meter Read Response (Interval)) business object is used to check for errors related to filtering for interval on demand read commands. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D7-CCOMPEVIN algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled). The value of the “Filter if invalid external MC identifier” on this algorithm must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

Receiving Results via JMS Queue

Some Silver Spring Networks commands can be configured to return results to a JMS queue, eliminating the need for the system to periodically poll for completed commands. This feature is enabled by setting the **Receive Job Results via JMS** parameter to "Yes" on the communication type used by the command.

The table below lists the commands that support this feature and the communication type business objects that correspond to each.

Command	Communication Type Business Object
Connect/Disconnect	D7-ConnectDisconnectType
Device Status Check	D7-AddPingJobType
On-Demand Read (Interval)	D7-AddMeterReadJobIntervalType
On-Demand Read (Scalar)	D7-AddMeterReadJobScalarType

The JMS queue to which command results are sent (all commands send results to the same JMS queue) is defined during installation. Refer to **Deploying the SOA Adapter for Silver Spring Networks** in the *Oracle Utilities Smart Grid Gateway Installation Guide* for details about creating JMS queues.

Device Communication Base Package Business Objects

The Silver Spring Networks Adapter base package includes the following communication business objects:

Business Object Name	Description
D7-AddMeterReadJobInterval	SSN - Add Meter Read Job (Interval)
D7-AddMeterReadJobScalar	SSN - Add Meter Read Job (Scalar)
D7-AddPingJob	SSN - Add Ping Job
D7-ConnectDisconnect	SSN - Connect or Disconnect
D7-ConnectDisconnectResp	SSN - Connect or Disconnect Response
D7-GetStatus	SSN - Get Status
D7-GetStatusResponse	SSN - Get Status Response
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping Job Response

Copyright © 2026, Oracle and/or its affiliates.

Business Object Name	Description
D7-ReplaceDeviceAtLocForDecomm	SSN - Replace Device At Location (Decomm)
D7-ReplaceLocation	SSN - Replace Location

Silver Spring Networks Event Data Mapping

The Silver Spring event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

Silver Spring Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders

Copyright © 2026, Oracle and/or its affiliates.

- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

Outbound Message Type	Description
D7-COMMS	Replace Device At Location
D7-OB MSG TY	Outbound Message Type SSN

Inbound/Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound web services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Silver Spring Networks includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.
D1-DeviceStatusCheck	Device Status Check This service is invoked by the integration layer to instantiate a Device Status Check command.
D1-InitialLoadIMD	Used by OSB to instantiate an IMD The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service	Description
D1-RemoteConnect	Remote Connect This service is invoked by the integration layer to instantiate a Remote Connect command.
D1-RemoteDisconnect	Remote Disconnect This service is invoked by the integration layer to instantiate a Remote Disconnect command.
D7-ConnectDisconnectResponse	Connect Disconnect Response Retrieves response for Remote provisioning Job Connect or Disconnect commands.
D7-GetStatusResponse	D7-GetStatusResponse Retrieve response from the Get Status command.
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping Response Retrieves response from the Ping Job Response command.

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to Silver Spring Networks format, invoking process callouts and invoking the remote endpoint to trigger the device events.

OnDemandRead Composite Process — Provides access points to edge application and handles data between edge application and head end system. It invokes sequence of web methods to head end system and retrieves meter read and send it back to Edge application.

ConnectDisconnect Composite Process — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and invokes the remote endpoint to trigger the connect event. A second, asynchronous reply will call back into the OUAF layer when the status change is completed at the head-end system. Another asynchronous reply will call back into the OUAF layer to send Meter Read Results.

CommissionDecommission Composite Process — Performs the conversion from Oracle Utilities format to SSN UIQ format, invokes process callouts, and invokes the remote endpoint to trigger the commission or decommission of meter.

DeviceStatusCheck Composite — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and makes a call via a proxy to the head-end system starting the Meter Ping operation. In an ideal scenario, the job status is returned as completed and the results are acquired and sent back to OUAF. If the job takes longer, OUAF will initiate a second request that will poll the head end system for the job status. When the job is completed, the results are returned to OUAF.

Common Composite — Contains two main classes of operations: Proxies and ProcessCallouts. Proxies are simple mediators that forward a web service call to a preset

Copyright © 2026, Oracle and/or its affiliates.

endpoint. No transformations are performed. They are convenient because they allow head end URLs and security to be set in a single composite. ProcessCallouts are points of customization which allow users to modify data and/or initiate some external business process.

BulkRequest Composite — Provides access points to requesting application. It decouples the bulk request into single commands for each meter/device in the request and sends it to edge application for processing.

Web Services

The following web services are all defined in the Silver Spring Networks head-end system:

- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceDeviceAtLocation
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/CommissionDecommissionService
- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceLocation
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/CommissionDecommissionService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** AddRemoteProvisioningJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/ConnectDisconnect/ConnectDisconnectService
- **DeviceStatusCheckService**
  - **BPEL Process:** DeviceStatusCheck
  - **Operation:** AddPingJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/DeviceStatusCheck/DeviceStatusCheckService
- **AddMeterReadJobService**
  - **BPEL Process:** AddMeterRead
  - **Operation:** AddMeterReadJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/AddMeterReadJobService
- **OnDemandReadService**
  - **BPEL Process:** OnDemandRead
  - **Operation:** GetJobStatus
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/OnDemandReadService

Copyright © 2026, Oracle and/or its affiliates.

Silver Spring Networks Utility IQ Web Services

The following table describes the Silver Spring Networks Utility IQ web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Command	AMI Adapter Business Objects	Silver Spring Networks Web Services	Silver Spring Networks Operations
Device Commissioning	D7-ReplaceLocation	Device Manager	findDevice ReplaceLocation
Device Decommissioning	D7-ReplaceDeviceAtLocForDecomm	Device Manager	findDevice ReplaceDeviceAtLocation
Remote Connect/ Remote Disconnect	D7-ConnectDisconnect D7-GetStatus	Device Manager Job Manager DeviceResults	findDevice addRemoteProvisioningJob getJobStatusForDevice getRemoteProvisioningResultsByJobID findJob getJobStatus getMeterReadResultsByJobID
Device Status Check	D7-AddPingJob	Device Manager Job Manager DeviceResults	findDevice addPingJob, getJobStatus getPingResultsByJobID
On-Demand Read	D7-AddMeterReadJobInterval D7-AddMeterReadJobScalar	Device Manager Job Manager DeviceResults	findDevice addMeterReadJob getJobStatus getMeterReadResultsByJobID

Configuring a Silver Spring Networks Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks to communicate with the Silver Spring Networks UtilityIQ.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Inbound Web Services

Copyright © 2026, Oracle and/or its affiliates.

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Silver Spring Networks UtilityIQ application in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	Used by OSB to instantiate an IMD
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D7-ConnectDisconnectResponse	Connect Disconnect Response
D7-GetStatusResponse	Get Status Response
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping Response

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests. An message sender should be configured for each command.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Copyright © 2026, Oracle and/or its affiliates.

Message Sender	Description
D7-DECOMM	SSN Replace Device at Location for Decommission
D7-COMM	SSN Replace Location - Commission
D7-ADDJOB	SSN Add Meter Read Job
D7-ADDPING	SSN Add Ping Job
D7-CONNECT	SSN Connect Device
D7-GTSTATUS	SSN Get Status

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/ssn/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/SSN/**<SERVICE>**
- where:
- **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
- **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
- **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
- **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
- **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D7-COMMS	Replace Device At Location

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Type	Description
D7-OB MSG TY	Outbound Message Type SSN

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Silver Spring Networks UtilityIQ must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - Silver Spring Networks:

- **External System**: Silver Spring Networks
- **Description**: Silver Spring Networks
- **Outbound Message Types**:

Outbound Message Type	Description
D7-OB MSG TY	Outbound Message Type SSN

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D7-Request xsl
- **Response XSL**: D7-Response.xsl

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the Silver Spring Networks UtilityIQ must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - Silver Spring Networks:

- **Service Provider**: Silver Spring Networks
- **Description**: Silver Spring Networks

Copyright © 2026, Oracle and/or its affiliates.

- **External Reference ID**: Silver Spring Networks
- **External System**: Silver Spring Networks
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **Pre-Commissioning Device ID Type**:
- **AMI Measuring Component ID Type**: Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Silver Spring Networks service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Silver Spring Networks service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D7-ReplaceLocation	Commission
Device Decommission	Device Removal	D7-ReplaceDeviceAtLocForDecomm	Decommission
Device Status Check	Device Status Check	D7-AddPingJob	Get Status

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Default Business Object	Default Outbound Message Type
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D7-AddMeterReadJobScalar	Read Meter Data
On-Demand Read (Interval)	On-Demand Read (Interval)	D7-AddMeterReadJobInterval	Read Meter Data
Remote Connect	Remote Connect	D7-ConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	D7-ConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Silver Springs Network endpoint override DVM (D7–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Silver Springs Network adapter and the command used with each:

DVM Key	Commands
DeviceResults4.4	Remote Connect
DeviceResults4.7	Remote Disconnect
DeviceResults4.10	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check
DataAggregation4.4	Not used in this version.
DataAggregation4.7	
DataAggregation4.10	
DeviceManager4.4	Device Commissioning
DeviceManager4.7	Device Decommissioning
DeviceManager4.10	Remote Connect
	Remote Disconnect
	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check

Copyright © 2026, Oracle and/or its affiliates.

DVM Key	Commands
JobManager4.4	Remote Connect
JobManager4.7	Remote Disconnect
JobManager4.10	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check
ProcessCallout	User Exit Functions

> **Note**
>
> The numbers in the keys above (4.4, 4.7, and 4.10) designate the version of the Silver Springs Network IQ head-end system.
>
> Only a single version of each key should be defined, based on the version of the Silver Springs Network IQ head-end system

To define an override Endpoint URI for the Silver Springs Network adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.
2. Select the D7–EndpointOverrides.dvm in the left panel.
   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.
3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.
5. Enter the appropriate values in the **Add Domain Values** dialog as follows:
   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.
6. Click **OK**.

   The new DVM value will appear in the list of keys.
7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

Copyright © 2026, Oracle and/or its affiliates.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Silver Spring Networks Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Silver Spring Networks.

This section outlines some of the extendable lookups that must be configured for use with the Silver Spring Networks adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system.
- **Description**: A description of the device event.
- **Status**: The status of the lookup value (can be Active or Inactive).
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

UOM Code to Standard UOM Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the head-end system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive).
- **Unit of Measure**: The unit of measure defined in the system.

Interval Status Code to Condition Mapping

Interval usage received from the Silver Spring Networks UtilityIQ can include Silver Spring Networks interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Silver Spring Networks Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Silver Spring Networks interval status codes to standard status codes when receiving usage from the Silver Spring Networks UtilityIQ.

Each value defined for the Silver Spring Networks Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Silver Spring Networks interval status code
- **Description**: A description of the interval status code.

Copyright © 2026, Oracle and/or its affiliates.

- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

Other Extendable Lookups

Business Object Name	Description
D7-ActivationStatusLookup	SSN - Activation Status
D7-ChannelStatusLookup	SSN - Channel Status
D7-DataTimeZoneLookup	SSN - Data Time Zone
D7-DeviceStatusLookup	SSN - Device Status
D7-ExecutionStatusLookup	SSN - Execution Status
D7-HeadendSQILookup	SSN - Head-End SQI to Standard SQI
D7-HeadendTOULookup	SSN - Head-End TOU to Standard TOU
D7-IntervalStatusLookup	SSN - Interval Status
D7-MeterConnectionStatusLookup	SSN - Meter Connection Status
D7-MeterReadTypeLookup	SSN - Meter Read Type
D7-OldAdminStateNameLookup	SSN - Old Admin State Name
D7-PriorityLookup	SSN - Priority Status
D7-ProvisioningActionLookup	SSN - Provisioning Action
D7-ProvisioningCommandStatus	SSN - Provisioning Command Status
D7-ProvisioningDataType	SSN - Provisioning Data Type
D7-ReadModeLookup	SSN - Read Mode
D7-RegisterReadSourceLookup	SSN - Register Read Source
D7-RegisterStatusLookup	SSN - Register Status
D7-SSNVersionLookup	SSN - Version
D7-StatusFlagNameLookup	SSN - Status Flag Name

Using the Silver Spring Networks Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks (SSN) includes a test harness that can be configured to simulate the Silver Spring Networks UtilityIQ head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

Using Smart Grid Gateway Middleware Test Harnesses

The Oracle Utilities Smart Grid Gateway Adapter includes middlware test harnesses for each supported head end system. The test harnesses can be configured to simulate a head-end system for initial testing of one-way or two-way commands. The test harnesses are Multispeak 3.0 standard compliant and include a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. This section describes the general components of the test harnesses.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This section applies to middleware implementations of Oracle Utilities Smart Grid Gateway **only**. This section does NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

- Test Harness Design
- Locating the WSDL for the Test Harness
- Web Services

Test Harness Design

Each test harness is divided into two main layers: a front end and back end layer. The front end set of services implements the head-end system interfaces, which receive requests corresponding to the specifications shown in the list below. Each of these services calls into the "back end" layer of the test harness, which defines meters and sets their attributes. These meters are stored in a file within the test harness called meterdb.xml.

> **Note**
>
> The meterdb.xml file can be modified pre-deployment, but not post-deployment. However, the test harness retains an in-memory "database" of the meters in the file. The in-memory representation can be modified using the utility web services. Note that any changes to the in- memory structure will be lost when the server is restarted or the test harness composite is redeployed.

Itron OpenWay

- www.itron.com.ami.2008.10.control
  - ReconnectMeter
  - GetReconnectMeterResult
  - DisconnectMeter
  - GetDisconnectMeterResult
- www.itron.com.ami.2008.10.data
  - ContingencyReadByEndpoint
  - GetContingencyReadByEndpointResult
  - InterrogateByGroup
  - GetInterrogateByGroupResult
- www.itron.com.ami.2008.10.control
  - PingByEndpoints
  - ReconnectMeter
  - GetReconnectMeterResult

Copyright © 2026, Oracle and/or its affiliates.

  - DisconnectMeter
  - GetDisconnectMeterResult
- www.itron.com.ami.2012.03.control.diagnostic
  - PingByEndpoints
  - GetPingByEndpointsResult
- www.itron.com.ami.2009.08.provisioning
  - AddMeterDefinitions
  - DeregisterMeters

Landis+Gyr

- http://www.multispeak.org/Version_3.0 MR_CB (Meter Reading_Customer Billing)
  - MeterAddNotification
  - MeterRemoveNotification
  - InitiateMeterReadByMeterNumber
- http://www.multispeak.org/Version_3.0 CD_CB (Connect/Disconnect_Customer Billing)
- InitiateConnectDisconnect

The LG Harness will send below responses to corresponding BPEL composites:

- http://www.multispeak.org/Version_3.0 CB_MR (Customer Billing_Meter Reading)
- ReadingChangedNotification
- http://www.multispeak.org/Version_3.0 CB_CD (Customer Billing_Connect/Disconnect)
- CDStateChangedNotification

Sensus RNI

- http://www.multispeak.org/Version_4.1_Release MR_Server
  - MeterAddNotification
  - MeterRemoveNotification
  - InitiateMeterReadingsByMeterID
- http://www.multispeak.org/Version_4.1_Release CD_Server
  - IitiateConnectDisconnect
- http://www.multispeak.org/Version_4.1_Release CB_Server
  - CDStatesChangedNotification (async reply)
  - ReadingChangedNotification (async reply)
- http://www.multispeak.org/Version_4.1_Release OD_Server
  - InitiateOutageDetectionEventRequest

Silver Spring Networks

- urn:com:ssn:schema:service:v1.4:DataAggregation and urn:com:ssn:schema:service:v1.6:DataAggregation
  - getMeterFieldStatus
- urn:com:ssn:schema:service:v1.4:DeviceManager and urn:com:ssn:schema:service:v1.6:DeviceManager

Copyright © 2026, Oracle and/or its affiliates.

  - FindDevice
  - ReplaceDeviceAtLocation
  - ReplaceLocation
- urn:com:ssn:schema:service:v1.4:DeviceResults and urn:com:ssn:schema:service:v1.6:DeviceResults
  - getRemoteProvisioningResultsByJobID
  - getMeterReadResultsByJobID
  - getPingResultsByJobID
- urn:com:ssn:schema:service:v1.4:JobManager and urn:com:ssn:schema:service:v1.6:JobManager
  - addRemoteProvisioningJob
  - getJobStatus
  - addMeterReadJob
  - addPingJobfindJob
  - getJobStatusForDevice

Locating the WSDL for the Test Harness

Follow these procedures to locate the test harness web service definition language (WSDL):

How to Use Enterprise Manager to Locate the WSDL

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the test harness composite:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the composite:
3. Click the UtilService WSDL URL link to see the WSDL in the browser, or right click and save it to your machine

Depending on your requirements, it may be necessary to download the associated schema found in the wsdl:types section. The URL can be pasted into a browser tab and downloaded in the same manner as the WSDL. The main schema has imported schemas that may also be required.

How to Use a Direct URL to Locate the WSDL

The WSDL can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the following form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}?WSDL
```

For example, the test harness WSDL can be found at the following locations for each SGG adapter.

Copyright © 2026, Oracle and/or its affiliates.

Itron OpenWay:

```
http://{server name}:{port number}/soa-infra/services/Itron_Test/
ItronTestHarness/UtilService? WSDL
```

Landis+Gyr:

```
http://{server name}:{port number}/soa-infra/services/LG_Test/LGTestHarness/
UtilService?WSDL
```

Sensus RNI:

```
http://{server name}:{port number}/soa-infra/services/Sensus_Test/Sensus/
UtilService?WSDL
```

Silver Spring Networks:

```
http://{server name}:{port number}/soa-infra/services/SSN_Test/SSNTestHarness/
UtilService?WSDL
```

Web Services

This section describes the web services included in the test harness BPEL composite for each supported head-end system.

- General Services
- Locate Meter Services
- Meter Administration Services
- Meter Attribute Administration Services

General Services

This section describes the general services of the test harness composite.

LoadMeterIndex

This web service loads the data store from the internal file. By default, if the store is already in memory, it will NOT reload. This behavior can be overridden with the forceReload parameter.

Input — LoadMeterIndexInput

Part: payload

Element: LoadMeterIndexRequest

Parameter	Description
forceReload	A switch telling the system whether to reload the meter index from the configuration file. Default is false.

Output — LoadMeterIndexOutput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: LoadMeterIndexResult

Parameter	Description
loaded	A boolean value for whether or not the index was reloaded from the configuration file

Fault — UtilityFault (see UtilityFault for more details).

ViewAuditTrail

This web service returns the audit log for the entire session.

Input — ViewAuditTrailInput

Part: payload

Element: ViewAuditTrailRequest

Output — ViewAuditTrailOutput

Part: payload

Element: ViewAuditTrailResult

An Entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing the latest update.

Fault — See UtilityFault, above.

UtilityFault

Fault with similar mapping to SGG/OUAF faults:

Typically, the faultCode, faultString, faultActor, and detail/text elements will be populated.

Locate Meter Services

This section describes the locate meter web services of the test harness composite.

FindMeters

This web service queries the data store for one or more meters. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input — FindMetersInput

Part: payload

Element: FindMetersRequest

Parameter	Description
id	The meter ID for which to search

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description
isRegex	The provided id can be a regex value when this parameter is true. Hint: to search for all meters in the system, use ".*" for the ID.

Output — FindMetersOutput

Part: payload

Element: FindMetersResult

Zero or more meter objects can be returned from the search

Fault — See UtilityFault. Unlike other methods, FindMeters does not throw an exception if the meter is not found. As such, it can be used to test for the existence of a Meter prior to querying for it.

IsMeterDefined

This web service queries whether a particular meter is defined in the data store.

Input — IsMeterDefinedInput

Part: payload

Element: IsMeterDefinedRequest

Parameter	Description
id	The meter ID for which to search

Output — IsMeterDefinedOutput

Part: payload

Element: IsMeterDefinedResult

Whether or not the provided ID is part of the index.

Fault — See UtilityFault. Thrown when meter id is not found.

GetMeter

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input — GetMeterInput

Part: payload

Element: GetMeterRequest

Parameter	Description
id	The meter ID for which to search

Output — GetMeterOutput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: GetMeterResult

The meter object requested by the ID.

Fault — See UtilityFault. Thrown when meter id is not found.

Meter Administration Services

This section describes the meter administration services of the test harness composite.

AddMeters

This web service adds a set of meters to the in-memory data store. This will not permanently add it to the control file.

Input — AddMetersInput

Part: payload

Element: AddMetersRequest

Parameter	Description
id	The identification code for the meter.
macID	A MAC address that must be unique within the system.
utility	An informational string.
serviceType	One of the valid ServiceType values (see schema). "Electric" is the only option at this time.
isCommissioned	Whether or not the meter is in a commissioned state.
loadActionCode	One of the possible LoadActionCode values used in Connect and Disconnect (see schema).
outageEventType	One of the possible OutageEventType values used in Device Status Check (see schema).
executionStatus	One of the possible ExecutionStates (see schema). These values control how the meter will respond to commands.
groupName	The name linking multiple meters together into a set.
jobExecutionStatus	One of the possible Job Execution Status values (see schema). This attribute determines how requested jobs perform.
updateIfExisting	Whether or not to update the meter with the provided values if it already exists in the index.
Comment	An informational string describing the purpose of the meter.
Channels	A listing of unit of measures supported by this meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output — AddMetersOutput

Part: payload

Copyright © 2026, Oracle and/or its affiliates.

Element: AddMetersResult

Whether or not each meter was added to the index.

Fault — See UtilityFault

RemoveMeter

This web service removes a meter from the in-memory data store. This will not permanently remove it from the control file.

Input — RemoveMeterInput

Part: payload

Element: RemoveMeterRequest

Parameter	Description
id	The ID for the meter to be removed

Output — RemoveMeterOutput

Part: payload

Element: RemoveMeterResult

Whether or not the meter was removed from the index.

Fault — See UtilityFault. Thrown when meter id is not found.

AddMeterChannel

This web service adds a new channel to a single meter.

Input — AddMeterChannelInput

Part: payload

Element: AddMeterChannelRequest

Parameter	Description
id	The identification code for the meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output — AddMeterChannelOutput

Part: payload

Element: AddMeterChannelResult

Whether or not the channel was added to the index.

Fault — See UtilityFault. Thrown when meter id is not found.

Copyright © 2026, Oracle and/or its affiliates.

RemoveMeterChannel

This web service removes a Channel from a meter.

Input — RemoveMeterChannelInput

Part: payload

Element: RemoveMeterChannelRequest

Parameter	Description
id	The ID for the meter to be removed.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.

These three parameters are combined to locate a unique channel

Output — RemoveMeterChannelOutput

Part: payload

Element: RemoveMeterChannelResult

Whether or not the channel was removed from the meter.

Fault — See UtilityFault. Thrown when meter id is not found.

ReadScalarMeter

This web service generates a scalar reading for each channel of a given meter.

Input — ReadScalarMeterInput

Part: payload

Element: ReadScalarMeterRequest

Parameter	Description
id	The ID for the meter to be read

Output — ReadScalarMeterOutput

Part: payload

Element: ReadScalarMeterResult

Zero or more scalar readings for the given meter.

Parameter	Description
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.
value	A random number representing the scalar reading.

Copyright © 2026, Oracle and/or its affiliates.

Meter Attribute Administration Services

This section describes the meter administration services of the test harness composite.

GetLoadActionCode

This web service queries whether the given meter is connected or disconnected. This method is used by the Connect/Disconnect service. The values for load action code are:

- Connect
- Disconnect

Input — GetLoadActionCodeInput

Part: payload

Element: GetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the load action code status should be retrieved

Output — GetLoadActionCodeOutput

Part: payload

Element: GetLoadActionCodeResult

The connection status of the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

SetLoadActionCode

This web service updates the load action code for a given meter. This method is used by the Connect/Disconnect service. The values for load action code are:

- Connect
- Disconnect

Input — SetLoadActionCodeInput

Part: payload

Element: SetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the load action code status should be set.
value	The new value of LoadActionCode to set on the meter.

Output — SetLoadActionCodeOutput

Part: payload

Element: SetLoadActionCodeResult

Copyright © 2026, Oracle and/or its affiliates.

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.

IsCommissioned

This web service queries the commissioning status for a given meter. This service is used by the Commission/Decommission process. The commissioning attribute can be true or false.

Input — IsCommissionedInput

Part: payload

Element: IsCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be retrieved

Output — IsCommissionedOutput

Part: payload

Element: IsCommissionedResult

The value of the Commissioned status attribute for the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

SetCommission

This web service updates the commissioning status for a given meter. This service is used by the Commission/Decommission process. The commissioning attribute can be true or false.

Input — SetCommissionedInput

Part: payload

Element: SetCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be set
value	The new value of Commissioned status to set on the meter

Output — SetCommissionedOutput

Part: payload

Element: SetCommissionedResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.

Copyright © 2026, Oracle and/or its affiliates.

GetExecutionStatus

This web service queries the status of the property controlling the overall execution of the command. The possible values of execution status are:

- Success - The command should complete successfully
- ResponseTimeout - The asynchronous response will never arrive
- SyncOperationFail - A simulated fault will occur in the during the initial request
- AsyncOperationFailure - A simulated fault will occur in the asynchronous response

Input — GetExecutionStatusInput

Part: payload

Element: GetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be retrieved

Output — GetExecutionStatusOutput

Part: payload

Element: GetExecutionStatusResult

The value of the ExecutionStatus attribute for the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

SetExecutionStatus

This web service updates the property controlling the overall completion of the command. The possible values of execution status are:

- Success - The command should complete successfully
- ResponseTimeout - The asynchronous response will never arrive
- SyncOperationFail - A simulated fault will occur in the during the initial request
- AsyncOperationFailure - A simulated fault will occur in the asynchronous response

Input — SetExecutionStatusInput

Part: payload

Element: SetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be set
value	The new value of ExecutionStatus to set on the meter

Output — SetExecutionStatusOutput

Part: payload

Element: SetExecutionStatusResult

Copyright © 2026, Oracle and/or its affiliates.

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.

Creating a Custom Adapter for Smart Grid Gateway

Customers can create their own customized adapter for using Smart Grid Gateway with a specific head-end system. This can be done using the Oracle Utilities Smart Grid Gateway Adapter Development Kit. This section describes the Adapter Development Kit and its components.

> **Note**
>
> Unless otherwise noted, the contents of this section apply to middelware implementations of Oracle Utilities Smart Grid Gateway **only**, and do NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

- Adapter Development Kit Overview
- Adapter Development Kit Processing
- Oracle Service Bus Processing
- Business Processing Execution Language Processing
- Configuring an Adapter Development Kit Head-End System
- Configuring Adapter Development Kit Extendable Lookups
- Using the Adapter Development Kit Test Harness
- The Adapter Development Kit Native Format
- Adapter Development Kit Native Format Example
- Adapter Development Kit Native Format Schema

Adapter Development Kit Overview

The Oracle Utilities Smart Grid Gateway (SGG) Adapter Development Kit provides a starting point for customers to create their own customized adapter for using SGG with a specific head-end system. SGG uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) to facilitate communication with other head-end systems. The following functionality can be configured:

Measurement Data and Device Event Loading: Data parsing and transformation via Oracle Service Bus from the smart meter format into the Oracle Utilities Service and Measurement Data Foundation unified format for measurement data and device events.

Measurement Data and Device Event Processing: Configurable mapping for meter status codes and device event names to Oracle Utilities Service and Measurement Data Foundation standard values.

Copyright © 2026, Oracle and/or its affiliates.

Commissioning Communication: Business objects and BPEL processes to support the Meter Add Notification message.

Connect Communication: Business objects and BPEL processes to support Initiate Connect/Disconnect and Connect/Disconnect State Change Notification messages.

Disconnect Communication: Business objects and BPEL processes to support Initiate Connect/Disconnect and Connect/Disconnect State Change Notification messages.

Decommissioning Communication: Business objects and BPEL processes to support Meter Removal Notification message.

On-Demand Read: Business objects and BPEL processes to support Initiate Meter Read by Meter Number and Reading Changed Notification messages.

Device Status Check: Business objects and Oracle BPEL processes to support the outbound Initiate Outage Detection communication, the inbound Outage Detection Event Notification communication, and the processing of Event Notifications.

What Does the Adapter Development Kit Provide?

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following:

Sample Oracle Service Bus Processes: A file parsing and transformation process framework, delivered as an Oracle Service Bus (OSB) configuration, that can be used as a reference for creating the functionality to import usage readings and device events from a specific head-end system.

Sample Oracle Business Process Execution Language Processes: A set of sample communication processes that support Remote Connect, Remote Disconnect, Commission, Decommission, Device Status Check, and On-Demand Read commands delivered as an Oracle Business Process Execution Language (BPEL) project.

Oracle Utilities Smart Grid Gateway Business Objects: Business objects to support measurement data and device event loading and two-way commands such Commission, Decommission, Remote Connect, Remote Disconnect, and Device Status Check.

Sample Two-Way Communication Test Harness: A head-end system emulator delivered as a BPEL composite.

Demonstration Data: A set of sample data provided in the demonstration database that can be used with the adapter processes as delivered (prior to any customization you might perform as part of your implementation).

Adapter Development Kit Processing

This section provides details concerning the OSB processing, BPEL Processes, and OUAF objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

Initial Measurement Data and Device Event Loading

The initial measurement data load and subsequent device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway

Copyright © 2026, Oracle and/or its affiliates.

service format. Payloads contain measurements and meter events in some head-end specific format OSB then places each service call into a JMS queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel then a service creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements and Device Events
- Base Package Business Objects

Initial Measurements and Device Events

The usage and event data exported from the AMI head-end system is loaded into Oracle Utilities as initial measurement and device event data. You can customize processing of this by configuring the following base product OSB projects:

1. **SGG-DG-CSV-BASE** contains components that are not to be changed on customer site. They implement functions specific to the CSV format processing such as validation and transformation.
2. **SGG-DG-CSV-CM** allows for customization and simplifies future upgrades.
3. **SGG-DG-XML-BASE** contains components that are not to be changed on customer site. They implement functions specific to the IMD and event upload format processing such as validation and transformation.
4. **SGG-DG-XML-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-DG-CSV-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-CSV-CM EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementEvent	Holds the name of inbound web service for the device event seeder.	
destinationRootElementIMD	Holds the name of inbound web service for the IMD seeder.	

The runtime configuration settings for the SGG-DG-XML-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-XML-CM EnvironmentSettings.xq file:

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
populateRawIMD	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the IMD seeder.	

For additional information about the sample OSB implementation included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Oracle Service Bus Processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the SGG-DG-CSV-CM and SGG-DG-XML-CM projects. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Processing Data in the Adapter Development Kit Native Format

Usage and event data exported from the AMI head-end system can be loaded into Oracle Utilities in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). You can customize processing of this by configuring the following base product OSB projects:

1. **SGG-DG-SEEDER-BASE** contains components that are not to be changed on customer site. They implement functions specific to the IMD and event upload format processing such as validation and transformation.
2. **SGG-DG-SEEDER-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-DG-SEEDER-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-SEEDER-CM EnvironmentSettings.xq file:

Element	Description	Valid Values
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement Interval	Holds the name of inbound web service for the interval IMD seeder.	

Copyright © 2026, Oracle and/or its affiliates.

Element	Description	Valid Values
destinationRootElement Scalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
destinationRootElement Event	Holds the name of inbound web service for the device event seeder.	
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false
filterUsage	Determines if usage should be filtered.	true false

See The Adapter Development Kit Native Format for more information about the ADK “native” format.

Publishing Initial Measurement Data and Device Events

The Adapter Development Kit can be configured to publish initial measurement data and device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-DG-SEEDER-CM OSB project are used in publishing measurement data and device events to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems.

Initial measurement and device event data is published in the “native” initial measurement data format (the format of the initial measurement and device event seeder business objects). This format includes normalized unit of measure, condition, and device event codes. See the *Oracle*

Copyright © 2026, Oracle and/or its affiliates.

Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide for more details about this format.

> **Note**
>
> Initial measurement data published via this feature is published prior to VEE processing. In addition, filtering can NOT be applied to initial measurement or device event data published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The Adapter Development Kit can be configured to filter initial measurement data passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```


Copyright © 2026, Oracle and/or its affiliates.

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **ADK - UOM Code to Standard UOM Mapping** extendable lookup (DG-HeadendUOMLookup) are passed into the system for processing.

Filtering Events

The Adapter Development Kit can be configured to filter device events passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **ADK - Device Event Mapping** extendable lookup (DG-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D3-CIM-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.

Copyright © 2026, Oracle and/or its affiliates.

8. Enter "false" in the **value** colulmn.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The adapter development kit base package includes the following device and initial measurement business objects:

Business Object Name	Description
DG-InitialLoadIMDInterval	Generic Initial Load IMD - Interval Used when loading customized adapter interval measurements into the system for the first time.
DG-InitialLoadIMDScalar	Generic Initial Load IMD - Scalar
DG-SmartMeter	ADK — Smart Meter

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, device status check, or an on-demand read. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related head-end system web service. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Inbound / Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Adapter Development Kit command:

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	ADK - Initiate Connect or Disconnect	ADK - Connect/ Disconnect State Change	Connect Device Create IMD
Remote Disconnect	ADK - Initiate Connect Disconnect	ADK - Connect/ Disconnect State Change	Disconnect Device Create IMD
Device Commissioning	ADK - Meter Add Notification		Device Commissioning
Device Decommissioning	ADK - Meter Remove Notification		Device Decommissioning
On-Demand Read (Scalar)	ADK - Initiate Meter Read by Meter ID	ADK - Reading Changed Notification	Create IMD
On-Demand Read (Interval)	ADK - Initiate Meter Read by Meter ID	ADK - Reading Changed Notification	Create IMD
Device Status Check	ADK - Initiate Outage Detection	ADK - Outage Detection Event Notif	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the DG-CCOMPEV (Create IMD Completion Event) algorithm on the Create Completion Events status of the DG-ReadingChgNotification (ADK - Reading Changed Notification) business object.

The DG-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the DG-ReadingChgNotification (ADK - Reading Changed Notification) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the DG-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Device Communication Base Package Business Objects

The adapter development kit base package includes the following communication business objects:

Business Object Name	Description
DG-ConnectDisconStateChgNtf	Generic - Connect/Disconnect State Change
DG-InitiateConnectDisconnect	Generic - Initiate Connect Disconnect
DG-InitiateMeterByMeterId	Generic - Initiate Meter Read By Meter ID
DG-InitiateOutageDetection	Generic - Initiate Outage Detection
DG-MeterAddNotification	Generic - Meter Add Notification
DG-MeterRemoveNotifcation	Generic - Meter Remove Notification

Copyright © 2026, Oracle and/or its affiliates.

Business Object Name	Description
DG-OutageDetectEvtNotification	Generic - Outage Detection Event Notification
DG-ReadingChgNotification	Generic - Reading Changed Notification

Event Data Mapping

The head-end system event file format can map as follows into the business object, D1-DeviceEventMappingLookup:

Head-End System Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)

Copyright © 2026, Oracle and/or its affiliates.

- Message XSL, W3C Schema, and Response XSL (as applicable)

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The adapter for Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events. The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event. If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.
D1-InitialLoadIMD	Used for initial measurement upload. The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.
D1-DeviceStatusCheck	Device Status Check This service is invoked by the integration layer to instantiate a Device Status Check command.
D1-InitialLoadIMD	Used by OSB to instantiate an IMD This inbound web service is used by OSB to instantiate an Initial Measurement Data for incoming interval usage in the Generic format.
D1-RemoteConnect	Remote Connect This service is invoked by the integration layer to instantiate a Remote Connect command.
D1-RemoteDisconnect	Remote Disconnect This service is invoked by the integration layer to instantiate a Remote Disconnect command.

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service	Description
DG-ConDisconStChgNotification	Initiate Connect Disconnect response. Retrieves asynchronous response from Initiate Connect Disconnect command.
DG-OutageDetectionEventNotification	Initiate Outage Detection Response Retrieve response from the Initiate Outage Detection Event Notification command.
DG-ReadingChangedNotification	Reading Changed Notification Notification that a device reading has changed.

Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following message senders:

Message Sender	Description
DG-COMM	Generic Adapter Commission
DG-DCOMM	Generic Adapter Decommission
DG-INTOUDET	Generic Adapter — Initiate Outage Detection Request
DG-XAISender	Commission / Decommission Sender

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 4.1 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

OnDemandRead Composite Process: Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

ConnectDisconnect Composite Process: Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

CommissionDecommission Composite Process: Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, a one of the following second business callout services is invoked to determined if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceived-Callout
- isExecutingCommissionCompleted-Callout
- isExecutingDecommissionReceived-Callout
- isExecutingDecommissionCompleted-Callout

Copyright © 2026, Oracle and/or its affiliates.

DeviceStatusCheck Composite: Invokes the remote endpoint to trigger the initiate outage detection event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

ProcessCallout Composite: This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

For additional information about the BPEL processes included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Business Processing Execution Language Processing.

Web Services

These web services are all defined in the head-end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location. These web services have HTTP security by default. You may need to modify the security as a part of your implementation.

Web Service	Related BPEL Process	Description
CB_ServerService	ConnectDisconnect	This web service defines the return interface, the means by which the status is returned to the calling system. This web service is only be invoked by the head end system, not OUAF. Only the CDStateChangeNotification web method is implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/ConnectDisconnect/CB_ServerService
CB_Server	OnDemandRead	This web service defines the return interface, the means by which the reading is returned to the calling system. This web service is only be invoked by the head end system, not OUAF. Only the ReadingChangedNotification web method is implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/OnDemandRead/CB_Server

Copyright © 2026, Oracle and/or its affiliates.

Web Service	Related BPEL Process	Description
OA_ServerService	DeviceStatusCheck	This web service defines the asynchronous return for InitateOutageDetectionEventRequest for solicited responses. This web service is only be invoked by the head end system, not OUAF. Only the ODEventNotification, PingURL, and GetMethods web methods are implemented in the composite. The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/DeviceStatusCheck/OA_ServerService

For additional information about the web services included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Business Processing Execution Language Processing.

Oracle Service Bus Processing

This chapter describes how to use Oracle Service Bus (OSB) and the components of the Oracle Utilities Smart Grid Gateway Adapter Development Kit to develop an adapter for importing usage reading and device events.

- OSB Overview
- Oracle Utilities Smart Grid Gateway Adapters

OSB Overview

The examples in this chapter provide an implementer with the information required to create a customized adapter using the Oracle Utilities Smart Grid Gateway Adapter Development Kit. Customized adapters are required in situations where a productized Smart Grid Gateway adapter is not available for a particular head-end system. The examples below explain how to receive and process inbound meter usage and events. The examples contain two parts: the incoming file parsing functionality implemented in Java, and an easy, configurable and customizable Oracle Service Bus (OSB) configuration, including an implementation of post-parsing functionality.

- OSB Prerequisites
- OSB Related Documentation
- OSB Processing Terms and Acronyms

OSB Prerequisites

Development of an Oracle Utilities Smart Grid Gateway adapter requires a number of technical skills as well as a development environment.

Required Technical Skills

The following technical skills are required for developing an adapter using the development kit:

Copyright © 2026, Oracle and/or its affiliates.

- Experience in Java programming
- Experience in development using XQuery and XPath
- Experience in XML programming
- Experience in OSB development and administration
- Understanding of Oracle Utilities Service and Measurement Data Foundation (SMDF) architecture

Environment Requirements

Development of the Oracle Utilities Smart Grid Gateway adapter requires an environment configured as follows:

- The Oracle WebLogic server must be installed.
- The OSB must be installed.
- An OSB domain must be created.
- Java Messaging Service (JMS) queues for Initial Measurement Data (IMD) and Notification messages must be created.
- The spl-d1-osb-2.0.1.jar containing the com.splwg.d1.sgg.osb.common Java package must be available. This is a single jar for all vendor-specific adapters.
- The archive sgg-osb-generic-adapter.zip must be available. This file is downloaded as a part of adapter development kit deployment. It contains the FileParser folder with sample source code and the com.splwg.dg.sgg.osb.configuration.jar file with sample OSB projects.

OSB Related Documentation

For more information about using the adapter development kit and Oracle Service Bus, see the following Oracle documents:

- *Oracle Fusion Middleware User's Guide for Technology Adapters*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*
- *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*
- *Oracle Fusion Middleware User's Guide for Oracle JDeveloper*
- *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*
- *Oracle Utilities Meter Data Management / Smart Grid Gateway Administrative User Guide*

OSB Processing Terms and Acronyms

This section lists several terms and acronyms used throughout this chapter.

Term	Definition
AMI	Advanced Metering Infrastructure
CSV	Comma-Separated Values
FA	File Adapter. Oracle JCA File Adapter.
FP	File Parser, Java code that parses input file and generates Plain XML
GA	Smart Grid Gateway generic adapter

Copyright © 2026, Oracle and/or its affiliates.

Term	Definition
GFP	Generic File Processor, Java code that is invoked by FA. It communicates with FP
IMD	Initial Measurement Data
IMD XML	The final XML containing the IMD.
JAR	Java Archive
JAXB	Java Architecture for XML Binding
JCA	Java Connector Architecture
JDeveloper	Freeware Oracle integrated development environment for development of Java-based SOA and Java EE applications
JMS	Java Message Service
MDB	Message Driven Bean
SMDF	Service and Measurement Data Foundation
OSB	Oracle Service Bus
OUAF	Oracle Utilities Application Framework
Plain XML	An intermediate XML containing all of the values from the inbound file. It is necessary because OSB Message Flow can handle only XML.
PPS	Processing Proxy Service, OSB proxy service where the validation and transformation of Plain XML is implemented. It is necessary to catch errors that occurred before Plain XML is transformed to IMD or Device Event seeder structures.
RFD	Rejected File Descriptor, File containing information about a portion of input file that was a cause of error.
RPPS	Result Processing Proxy Service, OSB proxy service where processing of transformation result is implemented. It is necessary to catch errors that occurred after Plain XML is transformed to IMD or Device Event seeder structures.
SGG	Smart Grid Gateway
Weblogic	Oracle J2EE Application Server
WSD	Web Session Directory
XML	eXtensible Markup Language
XPath	Programming language for selecting nodes from an XML document
XQuery	Programming language to query XML.

Oracle Utilities Smart Grid Gateway Adapters

This section contains detailed information on the different components that make up an Oracle Utilities Smart Grid Gateway adapter and the application logic that is needed to implement one.

- Adapter Components
- Processing Life Cycle
- Java Project
- OSB Configuration
- Setting Up the Adapter Environment

Copyright © 2026, Oracle and/or its affiliates.

- Additional Information

Adapter Components

The following table lists the adapter components:

Component	Description
JCA File Adapter (FA)	Technology adapter for reading and writing files on the local file system. It is responsible for polling files from incoming folder and passing it to the Generic File Processor.
Generic File Processor (GFP)	Framework component that is implemented in Java. It is responsible for instantiation of the File Parser, getting Plain XML from it, and passing it to OSB message Flow. It also performs payload statistics related functionality such as gathering data and generating notification messages.
File Parser (FP)	Component implemented in Java. It is specific to every different incoming format. It is responsible for parsing the incoming file, breaking the payload into logical parts (debatching), generating Plain XML for every logical part, and returning XML back to invoking GFP. **Plain XML** could be any logical subset of data that can be mapped to IMD. It is an intermediate format/schema between the raw data and IMD XML. The Plain XML schema must be defined by the file parser developer. Raw data is read in portions from the input file and converted to Plain XML before passing to OSB message flow. The reason for data being read in portions is as follows: Assume that an input file contains readings for 100 Measuring Components. You would want to read the input file in portions for two particular reasons. 1. Reading data of exactly one measuring component ensures that it is mapped appropriately to one IMD at a time. 2. Reading data in portions ensures that the entire file is not loaded into memory, which could cause resource issues.
Inbound Proxy Service	OSB proxy service that contains FA-related configuration settings. It statically routes all messages to Processing proxy Service.
Processing Proxy Service (PPS)	OSB proxy service that validates and transforms Plain XML. This service is necessary for catching errors that occur before Plain XML is transformed to IMD or Device Event seeder structures.
Result Processing Proxy Service (RPPS)	OSB proxy service that processes transformed data. This service is necessary for catching errors that occur after Plain XML is transformed to IMD or Device Event seeder structures.

Processing Life Cycle

This section outlines the life cycle from the initial input file to the initial measurement output. Refer to OSB Processing Terms and Acronyms for descriptions of the abbreviations and acronyms used in this chapter.

Copyright © 2026, Oracle and/or its affiliates.

See the **Logic Sequence Diagram** for general description of how the components interact.

OSB Processes

1. The JCA File Adapter starts reading the file.
2. FA instantiates and initializes the GFP.
3. FA invokes GFP passing an open stream to the incoming file.
4. GFP sends **D1-PayloadStatistics** notification message to FA.
5. GFP instantiates and initializes (if it is not done yet) the FP that is defined in the Inbound Proxy properties for parsing the file.
6. GFP invokes the FP (See File Parser Processes below).
7. In case of error in FP it invokes GFP. GFP sends **D1-PayloadErrorNotif** notification message to FA.
8. GFP gets the Plain XML as a return from FP.
9. GFP returns the Plain XML to the FA.
10. The Plain XML is passed via InboundProxyService to Processing Proxy service (PPS). In PPS's message flow, the Plain XML is validated and transformed to IMD XML.
11. PPS passes the IMD to Result Processing Proxy service (RPPS). RPPS publishes the IMD XML to a JMS queue, which is then picked up by Message Driven Bean (MDB).
12. OSB processes 3 through 9 are repeated until FP returns NULL on process 9.
13. GFP sends **D1-PayloadSummary** notification message to FA.

File Parser Processes

1. FP starts reading the input stream
2. FP instantiates a JAXB object of the type of Plain XML. The JAXB related interfaces and implementation classes are generated by using JDeveloper (See *Oracle Fusion Middleware User's Guide for Oracle JDeveloper* or Generating Java Classes from XML Schemas).
3. FP reads the input stream in segments until it hits a logical end of data with which it could create a Plain XML.
4. FP populates the data that is read into the Plain XML JAXB object.
5. FP marshals the JAXB object into Plain XML and passes back GFP. If the end of file (EOF) is reached FP passes back NULL.
6. File parser processes 2 through 5 are performed for every invocation by GFP.

Java Project

The source code in the example files contains an implementation of functionality to parse an incoming payload in Comma Separated Values (CSV) and XML formats. This implementation can be used as a reference for creating a project that is specific to your head-end system. The descriptions below are mostly related to CSV parsing functionality as it is a more complex and interesting example. The XML related parsing example is provided only to illustrate how XML-based input formats may also be processed.

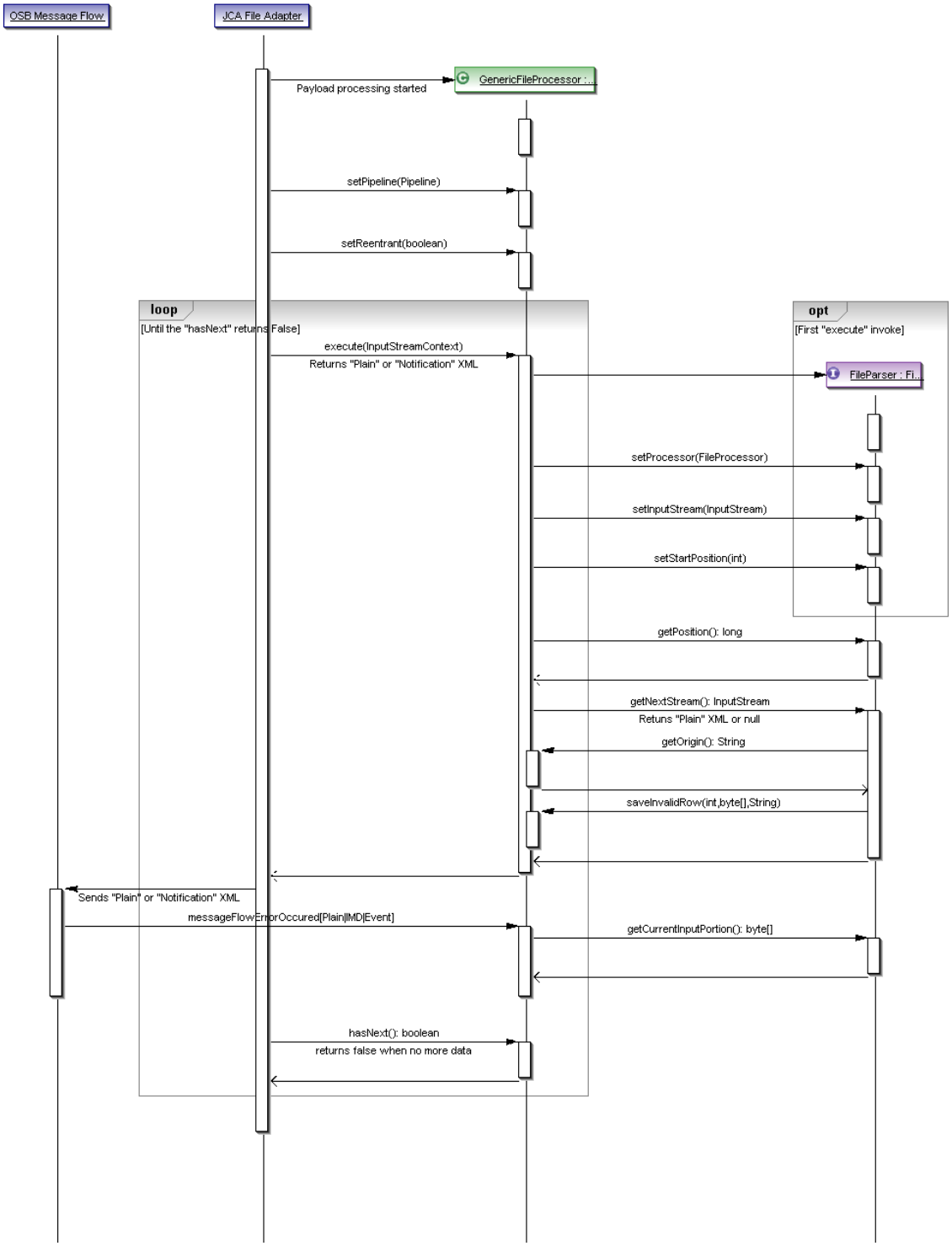
- Logic Sequence Diagram
- Java Implementation

Copyright © 2026, Oracle and/or its affiliates.

- Implementation Details

Logic Sequence Diagram

The sequence diagram below is helpful for understanding the File Parser functionality. The custom class must implement the FileParser and FileParser2 Java interfaces described in the SGG D1 jar to be qualify as a file parser. For more information, see FileParser Interface and FileParser2 Interface.



Copyright © 2026, Oracle and/or its affiliates.

When FA finds a file in the input directory, it instantiates and calls GFP which in turn starts an interaction with the FP. The interaction can be categorized into the following three phases:

1. Instantiation and Initialization
    - setProcessor()
    - setInputStream()
    - setStartPosition()
2. Transaction
    - getPosition()
    - getNextStream()
3. Exception Handling
    - getCurrentInputPortion()

Instantiation and Initialization Phase

The GFP is instantiated and receives an Input Stream when a file gets into the input directory. At that point an instance of the FP is instantiated. Certain methods, as specified above, are invoked on the FP to initialize.

Transaction Phase

Once the FP is initialized, GFP uses the FP instance to generate the Plain XML structure. This happens with the invocation of getNextStream() method on the FP.

On every invocation, the FP returns exactly one Plain XML. The FP invocations continues until NULL is returned from the FP. NULL indicates the end of file and an indication that parsing is complete for a specific file. GFP stops invoking getNextStream() for a particular input file when it receives a NULL from the FP.

FP uses the file input stream to read the file in portions. It could read byte-by-byte or line-by-line, but at the end of each getNextStream() method call it would have read only so much that is sufficient to create one Plain XML structure. For example, in an input file, if there is sufficient data for only one Plain XML then the first call to getNextStream() would return a Plain XML and the next one would return NULL. But if there is more than one, getNextStream() calls would continue to return Plain XML structures for every incoming portion until end of file is reached.

Exception Handling Phase

Exception handling inside the FP can be categorized into two phases: **Reactive** and **Recovery:**

Reactive Phase

This phase involves catching an exception and reporting it to the GFP. It is achieved by invoking saveInvalidRow() method on the GFP (this.fileProcessor).

Note that when an exception occurs, no further parsing is done and NULL is passed back from getNextStream().

If one fails to return NULL, errors such as “Simultaneous good and bad result returned by the File Parser” can be found in the Weblogic server log.

When an exception is reported to the GFP, it does three things:

Copyright © 2026, Oracle and/or its affiliates.

1. Creates an XML payload D1-PayloadErrorNotif and passes it to the OSB message flow. The message is published to the Notification Queue from inside PPS of the BASE OSB project.
2. It increments the “transaction error occurred” variable by invoking the utility method inside the D1 jar. All the errors occurring during the lifetime of a file parsing are captured and are reported through the message D1-PayloadSummary - which is again published to Notification Queue.
3. Finally, it creates a data file in the error directory with the portion of raw data that is read when the error occurred. It also creates a Rejected File Descriptor (RFD) file. The raw input portion is available by utilizing getCurrentInputPortion() method, which is discussed below.

As noted in the above image, the saveInvalidRow() method takes three input parameters:

1. The position at which the parsing started. It will be written to an RFD file. This is helpful in identifying the location of an error in the input file.
2. The raw data that will be written into error data file. It should be in the same to the incoming file format and it should contain data that is read since current invocation of getNextStream() method is started. It will allow correcting troubled data and place “fixed” file into an incoming folder for further re-processing.
3. The error message that is reported in D1-PayloadErrorNotif message. Note, the content of the error message has to be defined by the file parser developer.

Recovery Phase

This phase must not be confused with recovering from a handled exception discussed above. This case is when file parsing gets completely interrupted by power failure or network interruption.

Preparation for recovery: GFP maintains an internal index to the current read position of the file. This is achieved by GFP by invoking getPosition() method on the FP before every getNextStream() method call.

```
public long getPosition() {
    long retVal = this.fileProcessor.getCurrentInputPosition();
    return retVal;
}
```

Recovery: In case of an interruption and subsequent restore, GFP sets the start position on the File Parser. For instance, if the failure had occurred at byte 415, the start position would be set to 415 so that FP starts to read from that point and not from 0. The value 415 can be referred to as the recovery point.

```
public void setStartPosition(long position) {
    this.startPosition = position;
}
```

When the file parser starts parsing again, two things must occur to ensure that it starts parsing from the recovery point:

1. Store the recovery point to the startPosition field.
2. Incorporate skip logic to start the pointer from the recovery point.

Java Implementation

The Java component of the project includes the following classes and packages:

Copyright © 2026, Oracle and/or its affiliates.

1. **com.splwg.dg.osb.common.FileParserGenGeneric** class is a super class with members that are common for CSV and XML related functionality. It implements the com.splwg.d1.sgg.osb.common.FileParser interface (see FileParser Interface). The implementation of com.splwg.d1.sgg.osb.common.FileParser2 (see FileParser2 Interface) has been added to define the IMD & Event Upload Statistics related functionality.
2. **com.splwg.dg.osb.csv.FileParserCSV** class contains the implementation of the parsing functionality for CSV formats. It performs the following logic:
    - Parses the incoming payload.
    - Transforms incoming data to Usage or Event related structures based on incoming data type.
    - Extends those structures with file information and external service provider identifier.
    - Returns one by one that structures as a stream to caller.
3. **com.splwg.dg.osb.plain** package contains JAXB related interfaces and implementation classes corresponding to the Plain XML Schema
4. **com.splwg.dg.osb.xml.FileParserXML** class contains the implementation of parsing functionality for the “IMD and Event Online Upload” XML format (See the *Oracle Utilities Service and Measurement Data Foundation User’s Guide*). It performs the following logic:
    - Parses the incoming payload.
    - Extracts information about current device.
    - Breaks the incoming data to separate initial measurement data structures.
    - Extends those structures with device information, file information and external service provider identifier.
    - Returns one by one the structures as a stream to caller.

Implementation Details

This section provides an overview of the implementation details. See the comments in the source code for more detailed information.

OSB Configuration

This section outlines the OSB projects provided with the Adapter Development Kit.

- OSB Project Summary
- OSB Project Implementation Details

OSB Project Summary

The OSB Configuration consists of four projects. These projects can be categorized by either functionality or content:

Functionality by Incoming File Format

CSV format processing:

- SGG-DG-CSV-BASE
- SGG-DG-CSV-CM

XML format processing:

- SGG-DG-SEEDER-BASE

Copyright © 2026, Oracle and/or its affiliates.

- SGG-DG-SEEDER-CM
- SGG-DG-XML-BASE
- SGG-DG-XML-CM

Content (by purpose of content)

Content containing business logic implementation:

- SGG-DG-CSV-BASE
- SGG-DG-SEEDER-BASE
- SGG-DG-XML-BASE

Content containing configuration settings related to the objects and variables required during the processing of payloads:

- SGG-DG-CSV-CM
- SGG-DG-SEEDER-CM
- SGG-DG-XML-CM

Project Contents

The following table describes the contents of each of these projects:

Project	Description
SGG-DG-CSV-BASE	Contains the components responsible for “actual” processing of data coming in CSV format. It can be upgraded in future releases without affecting the customization and environment settings that are stored in the SGG-DG-CSV-CM project. This project performs the following functions: • Processes notification messages that are sent from the GFP. • Calls the customizable local service proxies for pre- and post- processing of passed data • Validates passed data against the XSD schema for Plain XML format • Transforms passed Plain XML into D1-InitialLoadIMD or D1-DeviceEventSeeder formats according to type of incoming data. • Sends result structures to destination JMS queue. • Updates statistic information via calling the messageFlowIncResultIMD or messageFlowIncResultEvent method accordingly. • Handles errors that occur to provide information to the File Processing component. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make Java calls to appropriate methods (messageFlowErrorOccurredPlain, messageFlowErrorOccurredIMD or messageFlowErrorOccurredEvent) according to the current processing stage and data type.

Copyright © 2026, Oracle and/or its affiliates.

Project	Description
SGG-DG-CSV-CM	Contains environment related configuration settings. It allows the customization and simplifies the future upgrades as well (See Configurng the OSB Project for example values).
SGG-DG-XML-BASE	Contains the components responsible for “actual” processing of data coming in XML format. It can be upgraded in future releases without affecting the customization and environment settings that are in SGG-DG-XML-CM project. This project performs the following functions: • Processes notification messages that are sent from GFP. • Calls the customizable local service proxies for pre- and post- processing of passed data. • Validates passed data. • Transform passed data. In current tutorial a simple XQuery code is used just to show the functionality available for transformation. • Sends result structures to destination JMS queue in form of acceptable by the D1-DeviceEventSeeder and/or D1-InitialLoadIMD inbound web services. • Updates statistic information via calling messageFlowIncResultIMD method. • Handles errors to provides information to GFP. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make a Java calls to appropriate methods (messageFlowErrorOccurred or messageFlowErrorOccurredIMD) according to the current processing stage.
SGG-DG-XML-CM	Contains environment related configuration settings. It allows the customization and simplifies future upgrades as well.

Copyright © 2026, Oracle and/or its affiliates.

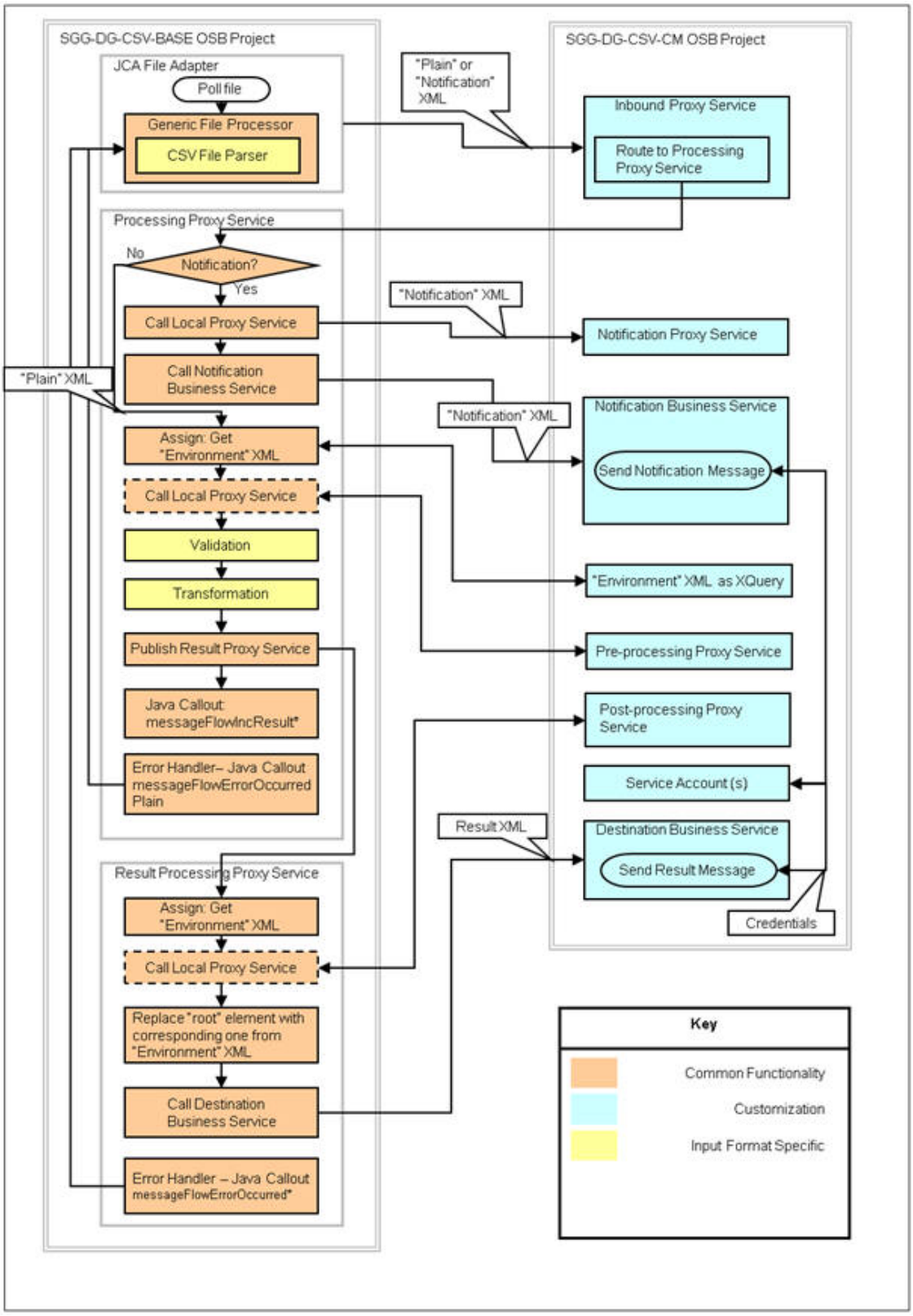
Project	Description
SGG-DG-SEEDER-BASE	Contains the components responsible for processing of data coming in “native” XML format. It can be upgraded in future releases without affecting the customization and environment settings that are in SGG-DG-SEEDER-CM project. This project performs the following functions: • Processes notification messages that are sent from GFP. • Calls the customizable local service proxies for pre- and post- processing of passed data. • Validates passed data. • Splits payload into individual initial measurements and/or device events. • Sends result structures to destination JMS queue in form of acceptable by the D1-InitialLoadIMD and/or D1-DeviceEventSeeder inbound web services. • Updates statistic information via calling messageFlowIncResultIMD method. • Handles errors to provides information to GFP. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make a Java calls to appropriate methods (messageFlowErrorOccurred or messageFlowErrorOccurredIMD) according to the current processing stage.
SGG-DG-SEEDER-CM	Contains environment related configuration settings. It allows the customization and simplifies future upgrades as well.

OSB Project Implementation Details

This section provides an overview of the implementation details. For more detailed information see the comments contained in the source code.

The following diagram shows the major processes and the data flow implemented in the OSB Configuration projects. The implementation description that follows is related to CSV parsing.

Copyright © 2026, Oracle and/or its affiliates.



Setting Up the Adapter Environment

This section contains detailed information about the setting up of an environment with an example adapter.

Copyright © 2026, Oracle and/or its affiliates.

- Setting Up the Java Project
- Configurng the OSB Project

Setting Up the Java Project

1. Open the sgg-osb-generic-adapter.zip file and extract the FileParser folder into a temporary location on your hard disk.
2. Create a Java project in JDeveloper and import source code into it from the folder extracted previously.
3. Add the spl-d1-osb-2.0.1.jar file to the project's Libraries and Classpath properties.
4. Deploy the project as a jar file to the destination OSB domain's lib folder.

   Note: The WebLogic server should be restarted after jar file is copied to lib folder. For more details, see *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

Configurng the OSB Project

1. Open the sgg-osb-generic-adapter.zip file and extract the com.splwg.dg.sgg.osb.configuration.jar file into a temporary location on your hard disk.
2. Ensure that OSB is running in the domain that is created for the adapter.
3. Import resources by using the OSB Administration Console via the **System Administration > Import/Export > Import Resources** menu item from the extracted com.splwg.dg.sgg.osb.configuration.jar file. Note that conflicts may occur during import. They will be fixed in the following steps.
4. Check and change the JCA Transport Configuration properties of the SGG-DG-CSV-CM/Proxy Services/InboundProxyService proxy service according to the deployment environment:

   Where:

   - **Endpoint Properties/SGG_INPUT_PARSER** is a name of class that implements file parsing functionality. See a Java project mentioned early.
   - **Endpoint Properties/SGG_ERROR_FOLDER** is a folder where a rejected transaction from incoming file will be placed in case when the parsing or validation is failed.
   - **Endpoint Properties/SGG_SP_EXT_REF_ID** is a value corresponding to the target Service Provider's external reference id on OUAF side. It is used in the XML structures related to the IMD & Event Upload Stats functionality. Also, this value will be placed into D1-InitialLoadIMD/serviceProviderExternalId and D1-DeviceEventSeeder/externalSenderId elements.
   - **Activation Spec Properties/IncludeFiles** is the naming convention that the Oracle File Adapter uses to poll for inbound files.
   - **Activation Spec Properties/PhisicalArchiveDirectory** is a folder where to archive successfully processed files.
   - **Activation Spec Properties/PhisicalDirectory** is an input folder to be polled.
5. Change the Endpoint URI in the Transport Configuration section of the SGG-DG-CSV-CM/Business Services/DestinationBusinessService business service according to the deployment environment with value containing URI for JMS server and queue where IMD or Device Event structures will be sent (see *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus* for more detail.):

Copyright © 2026, Oracle and/or its affiliates.

6. Change the Endpoint URI in the Transport Configuration section of the SGG-DG-CSV-CM/Business Services/NotificationBusinessService business service according to the deployment environment:
7. Update the credentials stored in the SGG-DG-CSV-CM/Service Accounts/DestinationServiceAccount service account according to the deployment environment with user ID and password required to access JMS server:
8. Repeat steps 4-7 for the SGG-DG-XML-CM project

Additional Information

This section contains additional information about the Adapter Development Kit.

- Processing Large Input Files
- FileParser Interface
- FileParser2 Interface
- FileProcessor Interface
- Generating Java Classes from XML Schemas

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

FileParser Interface

This section provides the source code of the FileParser interface used by the Java class delivered with the Java package described earlier in this section.

```
package com.splwg.d1.sgg.osb.common;
import java.io.IOException;
import java.io.InputStream;
/** <p>
 *  FileParser component is responsible for processing the input
 *  stream and returning a Plain XML containing all data from
 *  incoming transaction in a shape expected by the following
 *  Oracle Service Bus message flow.
 * </p>
 */
public interface FileParser {
    /**
     * Returns the next message in Plain XML format
     * or null if there are no more messages
     * @return next message as InputStream
     */
    InputStream getNextStream() throws IOException;
    /**
     * Returns the current position in the currently
     * parsed source input stream
     * @return the position in source input stream after
     * last incoming row was read and parsed
```


Copyright © 2026, Oracle and/or its affiliates.

```
     */
    long getPosition();
    /**
     * Sets source stream
     * @param input the Input Stream that should be parsed
     */
    void setInputStream(InputStream input);
    /**
     * Sets "owner" FileProcessor
     * @param owner the Instance of FileProcessor that will
     * be notified about current processing via call to
     * utility methods (saveInvalidRow and setParseFinishNote)
     */
    void setProcessor(FileProcessor owner);
    /**
     * Sets the position from which the parsing of source
     * stream should be started.
     * It is necessary if last time the processing of
     * current file has been interrupted
     * @param position the starting position
     */
    void setStartPosition(long position);
    /**
     * Returns the byte array that contains a portion of an incoming
     * file that most recently has been read and converted to Plain XML.
     * It should be in the same format as incoming file. The header and
     * tail records, if there are, should be included and updated according
     * to the content
     */
    byte[] getCurrentInputPortion();
    /**
     * Returns type of currently generated transaction - USAGE or EVENT
     */
    FileProcessor.TransactionType getCurrentTransactionType();
    /**
     * Returns the number of successfully generated transactions of given
type
     */
    int getTransactionCount(FileProcessor.TransactionType type);
    /**
     * Returns the number of expected transactions in input stream
     * Returns -1 if information not available
     */
    int getTotalExpected();
}
```

FileParser2 Interface

This section provides the source code of the FileParser2 interface used by the Java class delivered with the Java package described earlier in this section. This interface has been added to include a way for the File Processor to retrieve the file creation date which can be stored inside the incoming file.

```
package com.splwg.d1.sgg.osb.common;
import java.util.Date;
```


Copyright © 2026, Oracle and/or its affiliates.

```
public interface FileParser2 extends FileParser {
    /**
     * FileProcessor usage method
     * Returns the File creation date and time stamp stored
     * in the file
     * Returns null if information not available
     */
    Date getCreationDateTime();
}
```

FileProcessor Interface

This section contains a portion of the FileProcessor interface related to the file parsing functionality.

```
package com.splwg.d1.sgg.osb.common;
public interface FileProcessor {
    public enum TransactionType {
        USAGE, EVENT
    }
    /**
     * FileParser usage method
     *
     * @return a string containing the name of currently processed inbound
file
     */
    String getOrigin();
    /**
     * FileParser usage method Creates a file in the Error Folder and saves
     * passed row parameter to it
     *
     * @param position
     *            the number representing the position in file where invalid
     *            raw data is started
     * @param row
     *            the byte array containing an invalid raw data in vendor
     *            specific format the same with format of incoming file
     * @param failCause
     *            the string showing the cause of fail It will be used while
     *            logging the fail message.
     */
    void saveInvalidRow(long position, byte[] row, String failCause);
    /**
     * FileParser usage method
     * Returns the current offset in input file
     */
    long getCurrentInputPosition();
}
```

Generating Java Classes from XML Schemas

In JDeveloper you can use JAXB (Java Architecture for XML Binding) to generate Java classes from XML schemas. JAXB is an easy way to incorporate XML data and processing functions in Java applications without having to know XML. You can generate a JAXB 1.0 or 2.0 content model, including the necessary annotations, from an XML schema.

Copyright © 2026, Oracle and/or its affiliates.

When the JAXB binding compiler is run against an XML schema, JAXB packages, classes, and interfaces are generated. You can then use the generated JAXB packages and the JAXB utility packages in a binding framework to unmarshal, marshal, and validate XML content.

To generate Java classes from XML schemas with JAXB:

1. From the main menu choose **File > New > Business Tier > TopLink/JPA** and select either JAXB 1.0 or 2.0 Content Model from XML Schema to open the compilation dialog.
2. Select the schema file and optionally the JAXB customization file to use and the package to which the generated classes will be added.

The JAXB package and generated classes are added to the Application Resources folder.

Business Processing Execution Language Processing

This section describes the Business Process Execution Language (BPEL) processes included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit that support two-way communication between the adapter and the head-end system. The ideas and general practices described in this section can be applied to any other adapter.

- WSDLs, Endpoints, and Messages
- Composite Components
- Configuring and Customizing Adapter BPEL Processes
- Commands
- Working with Enterprise Manager
- MultiSpeak Implementation

WSDLs, Endpoints, and Messages

WSDLs are Web Service Definition Language files that describe a web service. New services must utilize the WSDLs to determine the structure of requests and responses. For remote services, they contain the locations to access the services. For both remote and local (hosted) services, the files will contain the definitions for each web service including names, arguments, exceptions, and the structure of the input and output messages.

In Oracle Serivce Oriented Architecture (SOA), WSDLs can be classified as a “service” or as a “reference.” Services are hosted on the SOA server; that is, the implementation of the web service is on the local application server. Referenced services are implemented on a different server. This distinction is also relevant to the composite level. Services are implemented in the current composite. References are located elsewhere, possibly on the same app server, but in a different composite.

An “endpoint” is simply the URL for a web service. Since web services communicate via HTTP, each will have a unique URL. Once a service’s endpoint is known, a message service can be targeted to it.

How to locate WSDLs and Endpoints

There are two ways to locate WSDLs in an installed Adapter: through Enterprise Manager and by using a direct URL. Only “hosted” WSDLs can be located in this way. Referenced (remote) WSDLs must be either located in source code or obtained from the hosted location.

How to Use Enterprise Manager to Locate WSDLs and Endpoints:

Copyright © 2026, Oracle and/or its affiliates.

1. Open Enterprise Manager and use the navigation pane to open the Dashboard of the desired composite.
2. The top bar of the Dashboard contains several buttons and icons. One of these is an “earth” icon with a puzzle piece over it. Click this icon to display the WSDL and endpoint URIs for the composite.
3. Click the URL link to see the WSDL in the browser, or right click and save it to your machine for use in developing new services.

   Depending on the requirements, It may be necessary to download the associated schema or WSDL file(s). Schemas are available within the WSDL’s “types” element. Find associated WSDL URLs within the import element. That URL can be pasted into a browser tab.

   The endpoint for the service is also visible in this window. It is this URL that should be added to message sender configuration. For example:

   The value HTTP Header/SOA Action is taken from the service’s WSDL. Locate the wsdl:definitions/wsdl:binding/wsdl:operation/soap:operation element for the web method being invoked. An attribute called soapAction will contain the value for this field. A shortcut to this field is using Enterprise Manager's Testing framework (locate the Test tab from the dashboard view for the composite). Once the WSDL is parsed, the SOAP Action will appear in a field on the request tab:

   The values **HTTP Login User** and **HTTP Login Password** should be set to a valid WebLogic user that has access to the module. The **HTTP Method** should always be set to POST and the HTTP URL 1 should be set to the value of the endpoint above.

How to Use a Direct URL to Locate WSDLs and Endpoints

The WSDLs can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}
```

Composite Components

This section outlines several important logical features of the BPEL composites.

- Composite Properties
- Proxies
- Process Callouts

Composite Properties

Most composites contain properties within the main file, composite.xml. They are essentially global constants that can be preconfigured with default values and changed at deployment time. They can be accessed post-deployment in Enterprise Manager.

Typically, these properties represent timeouts and boolean properties for activating or deactivating functionality. Other uses might include setting default values of constants such as URLs. They are accessed with the ora:getPreference() BPEL function.

They can be changed during development, at deployment time, or after deployment using different techniques for each.

Copyright © 2026, Oracle and/or its affiliates.

Development Changes

At development time, properties can be created at will within the “component” sections of the composite.xml file. The image below shows an example of the properties found in DeviceStatusCheck:

```
<component name="DeviceStatusCheck">
  <implementation.bpel src="DeviceStatusCheck.bpel"/>
  <property name="bpel.preference.isExecutingInitODRequestReceivedCallout">true</property>
  <property name="bpel.preference.isExecutingInitODEventRequestCompleted">true</property>
  <property name="bpel.preference.isExecutingODEventNotificationArrived">true</property>
  <property name="bpel.preference.timeout.callback.years">0</property>
  <property name="bpel.preference.timeout.callback.months">0</property>
  <property name="bpel.preference.timeout.callback.days">0</property>
  <property name="bpel.preference.timeout.callback.hours">0</property>
  <property name="bpel.preference.timeout.callback.minutes">0</property>
  <property name="bpel.preference.timeout.callback.seconds">45</property>
  <property name="bpel.preference.OD_ServerCallbackEndpoint">http://127.0.0.1:8001/soa-infra/serv
  <property name="bpel.preference.externalSenderID">DSC Sensus</property>
  <property name="partnerLink.OD_ServerProxy.idempotent" type="xs:string"
            many="false">false</property>
</component>
```

Properties must be prefixed with “bpel.preference.” The BPEL process associated with them (see the implementation.bpel element) can use the ora:getPreference() method to extract the value. When accessing the values, the “bpel.preference” prefix is dropped.

Pre-Deployment Changes

While the values of properties cannot be changed at runtime, they can be altered using the configuration plan each composite uses for deployment. Most configuration plans will contain multiple property elements within the “component” section. Modifying the value in the replace element and redeploying the composite will change the behavior of the property:

```
<component name="DeviceStatusCheck">
   <property name="bpel.preference.isExecutingInitODRequestReceivedCallout">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.isExecutingInitODEventRequestCompleted">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.isExecutingODEventNotificationArrived">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.timeout.callback.years">
      <replace>0</replace>
   </property>
   <property name="bpel.preference.timeout.callback.months">
      <replace>0</replace>
   </property>
   <property name="bpel.preference.timeout.callback.days">
      <replace>0</replace>
   </property>
```

When the composite is deployed, the properties will contain the values in the “replace” elements.

Copyright © 2026, Oracle and/or its affiliates.

Post-Deployment Changes

After deployment, changes can still be made to the values of the properties through Enterprise Manager.

1. In the Enterprise Manager navigation window, open WebLogic Domain/{domain name}. Right click on the domain and select **System MBean Browser**.
2. A new navigation pane opens in the window on the right. Select **Application Defined MBeans/oracle.soa.config/Server: {domain name}/SCAComposite/{Composite} [1.0]/ SCAComposite.SCAComponent/{Component}** to access the controls for a particular component. A “component” is a single BPEL or Mediator within a larger SOA composite

   A list of options will open in the right pane.
3. Select **Properties.**

   The Property sheet shows a list of all the properties. Opening one allows editing of the property value.
4. Click **Apply** to save any changes.

Proxies

Proxy web services make no changes to the data going through the system and are just used to centralize endpoint configuration. The message structures are identical to those of the head-end system. Data is passed through without modification. Mediator objects within the Common composite are used to move data from Oracle Utilities Smart Grid Gateway to the head-end system.

Process Callouts

Some services are designated as “Process Callouts.” These services allow integrators direct access to the XML coming to and from the head-end system. They are useful in the event the integrator needs to provide custom enrichment or transformation of the inbound or outbound data. Because process callouts are generally located before and after calls to the head-end system or when data arrives, they can also serve as an “event” trigger if the integrator requires additional processing logic. The services define identical inputs and outputs and are based on head-end schema definitions. To use a callout, the integrator should create a web service implementing one or more of the WSDLs defined in the Common composite. Other composites should then be configured to target the new Callout-type service.

A typical example is for re-rendering the meter identifier being sent to the head-end system. If the standard meter ID should be modified, a process callout service is a candidate for making this change. Note that because the schema is based off the head-end system's definition, deviations from the schema are not allowed.

Configuring and Customizing Adapter BPEL Processes

This section outlines how to configure and customize the Oracle Business Process Execution Language (BPEL) processes provided with the Adapter Development Kit to work with your implementation.

- Editing Configuration Files
- Customizing BPEL Processes

Copyright © 2026, Oracle and/or its affiliates.

This section outlines the changes that need to made in various configuration and build files to enable the application servers used in your implementation to communicate with each other, and to support the specifics of your head-end system. These files are also used in packaging and deploying the processes.

Server Definitions

In order for your adapter to work properly, you must set up the application servers that run the components used by the adapter communicate with each other as follows:

- The Oracle Utilities Smart Grid Gateway application server must be able to send and receive messages to and from the SOA Suite application server.
- The SOA Suite application server must be able to send and receive messages to and from the head-end system application server, and the Oracle Utilities Smart Grid Gateway application server
- The head-end system application server must be able to receive and send messages from and to the SOA Suite application server.

The types of servers and the tokens that need to be replaced for each are listed below. Further below are the specific configuration files that must be modified, as well as the locations in each files that must be modified. Note that a port is listed, but may not be necessary depending on the type of installation. Also note that the SOA Server will have to have the partition name defined that is used.

- **SOA Server**: This is the application server running the SOA Suite components, including Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL). The SOA server is referenced using the following tokens:
  - {SOA_HOST}: the server where the SOA server has been installed
  - {SOA_PORT_NUMBER}: the port used by the SOA server
  - {SOA_PARTITION_DG}: the partition used by the SOA server. There are different partitions for the different adapters used by Oracle Utilities Smart Grid Gateway.
- **XAI (OUAF) Server**: This is the application server running the Oracle Utilities Smart Grid Gateway (including the Oracle Utilities Application Framework, or OUAF) software, including the inbound/outbound message components. This server is referenced using the following tokens:
  - {WEB_WLHOST}: the server where Oracle Utilities Smart Grid Gateway and the message components have been installed
  - {WEB_WLPORT}: the port used by the Oracle Utilities Smart Grid Gateway application
  - {WEB_CONTEXT_ROOT}: the root directory at which the Oracle Utilities Smart Grid Gateway inbound web service
- AMI Head-end Server: This is the application server running the head-end system software. The AMI head-end server is referenced using the following tokens:
  - {Headend_MR_Server_DG}: the URL for either the actual head-end system's MR_Server MultiSpeak implementation, or the URL to an emulator test harness being used (if applicable). For example, the Adapter Development Kit includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.
  - {Headend_CD_Server_DG}: the URL for either the actual head-end system's CD_Server MultiSpeak implementation, or the URL to an emulator test harness being

Copyright © 2026, Oracle and/or its affiliates.

used (if applicable). For example, the generic adapter includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.

  - {Headend_OD_Server_DG}: the URL for either the actual head-end system's OD_Server MultiSpeak implementation, or the URL to an emulator test harness being used (if applicable). For example, the generic adapter includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.

In addition, credentials for the WebLogic server must be specified in the build.properties file in the deploy folder:

- {WebLogic_UserID}: the user ID used to connect to the WebLogic server
- {WebLogic_Password}: the password for the {WebLogic_UserID} used to connect to the WebLogic server

Customizing BPEL Processes

This section provides an overview of how the sample BPEL processes provided with the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be customized to meet specific business requirements, such as using custom XSL transformations and adding/editing the steps involved in the process.

Using Custom XSL Files

This process references XSL files to transform the messages from the Oracle Utilities Smart Gird Gateway standard format into the format used by the “generic” head-end system (based on the Multispeak protocol) when sending messages to the head-end system, and from the head-end system format into the Oracle Utilities Smart Gird Gateway standard format when sending messages back to Oracle Utilities Smart Gird Gateway. These XSL file references can be changed to custom XSL files that transform the messages into and from the format used by your head-end system. Refer to Oracle BPEL and Oracle JDeveloper documentation for more information about referencing XSL files within a BPEL process.

Making Changes to the Process

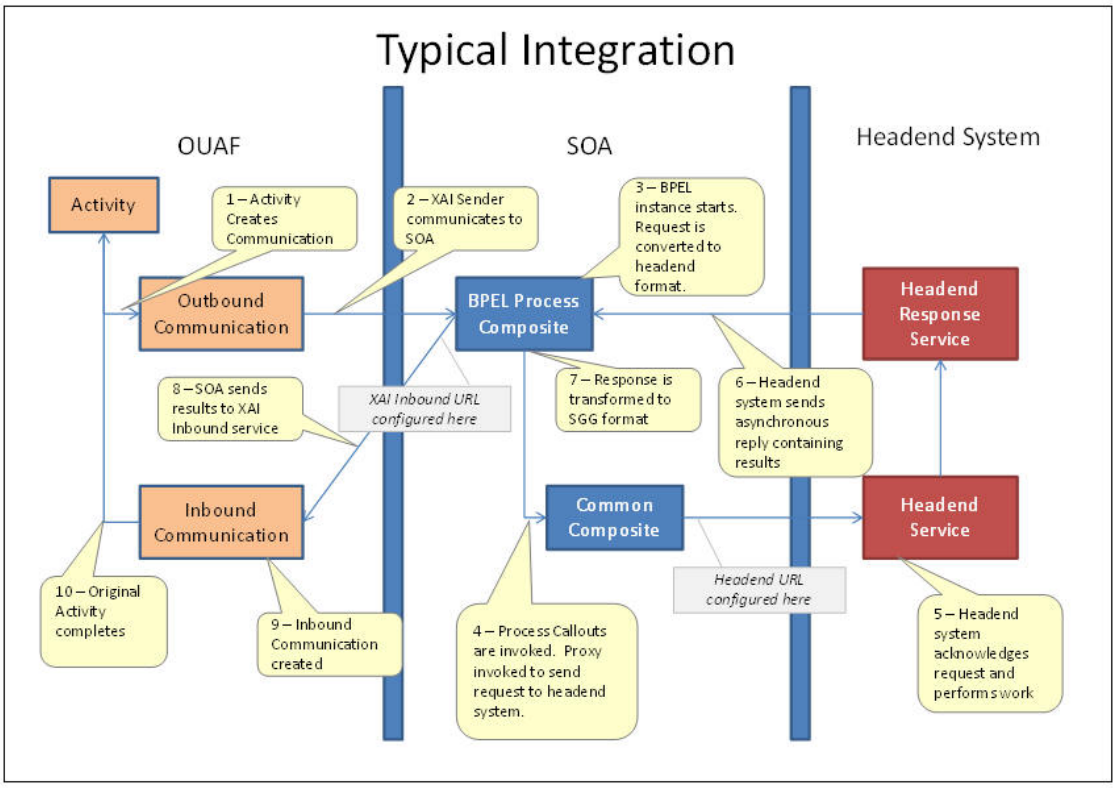
In addition to referencing custom XSL files for transformation, you can also add additional steps to the process to meet the specific business requirements of your implementation, or make other changes to the existing steps. Refer to Oracle BPEL and Oracle JDeveloper documentation for more information about adding and/or changing steps within BPEL process.

Note: Any changes made to this process for a particular AMI vendor and head-end system needs to be modeled based on the head-end system/AMI vendor’s business processes, including configuring the required request/response transactions required for a specific command (such a remote connect).

Commands

This section describes the general structure and flow of each SOA composite in the adapter development kit for the two-way commands. Most follow this basic model:

Copyright © 2026, Oracle and/or its affiliates.



- Common
- Commission / Decommission
- Connect / Disconnect
- Device Status Check
- On Demand Read

Common

The Common composite is a repository for files and logic that is used by other composites. For instance, the Proxies and default implementations of Process Callouts are defined in the Common composite.

Composite Properties

None.

Composite WSDLs

Callout Services

Process Callouts are customization points for integrators with identical inputs and outputs based on head-end formats. The Common composite contains default implementations in which incoming data is reflected back in an identical state. Each of the following is a Mediator component and uses an “echo”-type definition.

Process Callouts are usually associated with boolean composite properties which control whether they will be executed. They are activated by default so uses can easily inject their customizations, but setting the properties to false can be a performance optimization. When

Copyright © 2026, Oracle and/or its affiliates.

set to false, the associated Process Callout will not be made and execution will continue normally.

Service	Description
OA_CalloutService	Used when the asynchronous reply in DeviceStatusCheck arrives.
MR_CalloutService	Used during Commission/Decommission and OnDemandRead operations.
CB_CalloutService	Used by the asynchronous callback processes of OnDemandRead and Connect/Disconnect.
CD_CalloutService	Used by Connect/Disconnect.
OD_CalloutService	Used in DeviceStatusCheck.

Proxies

Placing all the proxy web services in the Common composite creates a place to conveniently set endpoints. Each is a Mediator component which passes the data to the head-end system without modification

Proxy Web Service	Description
OD_ServerProxy	Used to transmit a DeviceStatusCheck request to the head-end system.
CD_ServerProxy	Used to transmit a Connect/Disconnect request to the head-end system.
MR_ServerProxy	Used to transmit an OnDemandRead request to the head-end system.

Other

AuxiliaryRoutinesService: A container defining helpful, commonly used functions:

- **FindExpTime:** Many MultiSpeak functions contain an expTime element, which is used to deliver the length of time to wait for the command to complete before failing. (The included test harness does not support timeouts.) If the input to the command composite does not contain this timeout, the process will read from its properties the years, months, days, hours, minutes, and seconds to wait for a result. This web service utilizes a Java method to combine these inputs into the proper XML field used in the MultiSpeak API. The properties are also used to control the timeout wait period of asynchronous callbacks within a command's BPEL process.
- **FindTimeout:** When the expTime element and units are supplied as input, it is usually still necessary to compute the timeout used in asynchronous callbacks. This method accepts the MultiSpeak fields and replies in the “P0Y0M0DT0H0M0S” format understood by BPEL.

Commission / Decommission

The CommissionDecommission composite takes care of registering and de-registering a device with the head-end system.

Copyright © 2026, Oracle and/or its affiliates.

Composite Properties

Property Name	Default Value	Description
isExecutingCommissionReceived-Callout	true	Controls whether the Request Arrived callout executes.
isExecutingCommissionCompleted-Callout	true	Controls whether the Request Completed callout executes.
isExecutingDecommissionReceived-Callout	true	Controls whether the Request Arrived callout executes.
isExecutingDecommissionCompleted-Callout	true	Controls whether the Request Completed callout executes.

Composite WSDLs

Composite	Description
CommissionDecommission Service	Entry point for the CommissionDecommission command. The operations are synchronous, so no additional WSDLs are needed.
MR_ServerCallout	Reference describing the process callout.
MR_ServerProxy	Reference describing the proxy used to invoke the head-end system.

Process Flow

1. SGG/OUAF invokes MeterAddNotification/MeterRemoveNotification operation using CommissionDecommissionService.
2. Test whether MeterAddNotification or MeterRemoveNotification is invoked. In both cases, the following steps are common for both the operations.
3. Composite properties are loaded and local variables are initialized.
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in MR_ServerCallout.Assign updated data to head-end request.
6. Invoke MeterAddNotification/MeterRemoveNotification via the MR_ServerProxy to the head-end system.
7. If required, execute a process callout in MR_ServerCallout. Assign updated data to head-end response.
8. Transform Header and Body head-end responses to SGG/OUAF format.
9. Reply to SGG/OUAF with synchronous results.

Connect / Disconnect

The ConnectDisconnect composite is responsible for starting and stopping the recording of usage data for a meter.

Copyright © 2026, Oracle and/or its affiliates.

Composite Properties

Property Name	Default Value	Description
CB_CDCallbackEndpoint	http://[SERVER_NAME]/soa-infra/services/DG/ConnectDisconnect/CB_ServerService	The default endpoint to which MultiSpeak should send the asynchronous CB_Server callback.
isExecutingCDReceivedCallout	true	Controls whether the Request Received callout executes.
isExecutingCDCompletedCallout	true	Controls whether the Request Received Completed callout executes.
isExecutingCDStatesChangedArrivedCallout	true	Controls whether the CD States Changed Notification arrival callout executes.
CallbackTimeoutYears	0	The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutMonths	0	The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutDays	0	The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutHours	0	The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.

Copyright © 2026, Oracle and/or its affiliates.

Property Name	Default Value	Description
CallbackTimeoutMinutes	0	The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutSeconds	45	The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.

Composite WSDLs

Composite	Description
ConnectDisconnectService	Entry point for the ConnectDisconnect operation. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-ConnectDisconnectStateChgNotification.
CB_ServerService	MultiSpeak WSDL hosted to receive asynchronous CDStatesChangedNotification callbacks from the head end system.
CB_ServerCallout	Reference describing the process callout used when the asynchronous callback from the head end system arrives.
CD_ServerCallout	Reference describing the process callout.
CD_ServerProxy	Reference describing the proxy used to invoke the head end system.
AuxiliaryRoutines	Reference to common helper routines in the Common composite.
DG-ConnectDisconnectStateChgNotification	A reference to an inbound web service capable of processing asynchronous results from a Connect/Disconnect request.

Process Flow

1. SGG/OUAF invokes InitiateConnectDisconnect operation using ConnectDisconnectService.
2. Composite properties are loaded and local variables are initialized.
3. Test whether the input includes a expiration time
   - If Yes: Use AuxilaryRoutines to compute timeout.
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time.
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in CD_ServerCallout. Assign updated data to head end request.

Copyright © 2026, Oracle and/or its affiliates.

6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the results.
7. Invoke InitiateConnectDisconnect via the CD_ServerProxy to the head end system.
8. If required, execute a process callout in CD_ServerCallout. Assign updated data to head end response.
9. Transform Header and Body head end responses to SGG/OUAF format.
10. Reply to SGG/OUAF with synchronous results.
11. Check for ErrorObject in the head end response.
12. If error object not found:

    Await asynchronous response from head-end system for the period of expiration time calculated during start of this flow:

    - Receive CDStatesChangeNotification from CB_ServiceService.
    - Generate reply with no errors for CDStatesChangeNotification
    - Synchronously reply to the head end.
    - If required, execute a process callout in CB_ServerCallout. Assign updated data to the incoming request.
    - Transform Header and Body of incoming request to OUAF inbound web service format described in DG-ConnectDisconnectStateChgNotification.
    - If request contains responseURL
      - Invoke the inbound web service on given responseURL with the transformed incoming request.
    - Else
      - Invoke the inbound web service on definedURL with the transformed incoming request.

Device Status Check

DeviceStatusCheck is used to determine the health of a meter and to test that the meter can be reached on the network.

Composite Properties

Property Name	Default Value	Description
OD_ServerCallbackEndpoint	http://127.0.0.1:8001/soa-infra/services/DG/DeviceStatusCheck/OA_ServerService	The default endpoint to which MultiSpeak should send the asynchronous OA_Server callback.
isExecutingInitODRequest-ReceivedCallout	true	Controls whether the Request Arrived callout executes.
isExecutingInitODEventRequest-Completed	true	Controls whether the Request Completed callout executes.
isExecutingODEventNotification-Arrived	true	Controls whether the ODEventNotification arrival callout executes.

Copyright © 2026, Oracle and/or its affiliates.

Property Name	Default Value	Description
timeout.callback.years	0	The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
timeout.callback.months	0	The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
timeout.callback.days	0	The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
timeout.callback.hours	0	The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
timeout.callback.minutes	0	The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
timeout.callback.seconds	45	The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.

Composite WSDLs

Composite	Description
OD_ServerService	Entry point for DeviceStatusCheck. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-OutageDetectionEventNotification.
OA_ServerService	MultiSpeak WSDL hosted to received asynchronous ODEventNotification callbacks from the head end system.

Copyright © 2026, Oracle and/or its affiliates.

Composite	Description
OD_ServerProxy	Reference describing the proxy used to invoke the head end system to initiate a Device Status Check.
OA_ServerCallout	Reference describing the process callout used when the asynchronous callback from the head end system arrives.
OD_ServerCallout	Reference describing the process callout used when initiating a Device Status Check.
AuxilaryRoutines	Reference to common helper routines in the Common composite.
DG-OutageDetectionEventNotificationXAI	A reference to an inbound web service capable of processing asynchronous results from a Device Status Check request.

Process Flow

1. SGG/OUAF Initiates the Outage Detection service using OD_ServerService
2. Composite properties are loaded. Local variables are initialized. These mostly include boolean settings describing the state (such as whether or not the synchronous reply from the head-end system returned)
3. Test whether the input includes a expiration time (expTime element).
   - If Yes: Use AuxilaryRoutines to compute timeout
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time
4. Both the Header and Body inputs are transformed to MultiSpeak format. This can be done either through direct mapping or through XSLT. In this instance, XSLT is used to transform the header and the body separately.
5. If the composite property indicates the process callout should occur, execute the “request arrived” process callout in OD_CalloutService. Assign updated data to head end request.
6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the outage detection results. Some head end systems use other approaches, such as pre-configuring the URL for callbacks or allowing polling to find the results. In this case, the URL is configured in a composite property and corresponds to an implementation of the OA_Server WSDL. When the head end system has results, it will deliver them to this URL using the ODEventNotification method.
7. Invoke InitiateOutageDetectionEventRequest via the OD_ServerProxy to the head end system.
8. Split processing to handle simultaneous activities. The Flow activity does this in BPEL and it makes sense here because one process will handle a synchronous response and return it to the caller. The other process will sleep, or “dehydrate,” until the asynchronous ODEventNotification arrives or until the timeout value is reached.
   - Handle synchronous response to SGG/OUAF
     - If the composite property indicates the process callout should occur, execute the “request completed” process callout in OD_CalloutService. Assign updated data to head end response.
     - Transform Header and Body head end responses to SGG/OUAF format.
     - Reply to SGG/OUAF with synchronous results.

Copyright © 2026, Oracle and/or its affiliates.

- Set boolean variable indicating that the synchronous reply has been returned.

- Await asynchronous response from head-end system.
  - Receive ODEventNotification from OA_ServerService. If the timeout period elapses, raise an error and stop waiting.
  - Synchronously reply to the head end with no errors
  - If the composite property indicates the process callout should occur, execute the “data arrived” process callout in OA_CalloutService. Assign updated data to the incoming request.
  - Transform Header and Body to inbound web service format described in DG-OutageDetectionEventNotification.
  - Invoke the inbound web service with the new data.

9. Fault Handlers look for any remote Fault returned from the head-end system. The MultiSpeak API does not define named faults, but they are still possible to encounter. For example, a security error or network error could be raised.
   - If the synchronous reply has been sent back to the caller, Terminate the flow. This indicates an error has occurred and more research in Enterprise Manager is warranted.
   - If the synchronous reply has not been sent, stop further processing and compose a reply which contains the fault information.

On Demand Read

OnDemandRead interrogates a meter for the usage at the current point in time.

Composite Properties

Property Name	Default Value	Description
CB_ServerCallbackEndpoint	http:/[SERVER_NAME]/soa-infra/services/DG/OnDemandRead/CB_Server	The default endpoint to which MultiSpeak should send the asynchronous CB_Server callback
IsExecutingOnDemandReadRequestReceived	true	Controls whether the Request Received callout executes
IsExecutingOnDemandReadRequestReceivedResponse	true	Controls whether the Request Received Completed callout executes
IsExecutingReadingChangedNotification	true	Controls whether the Reading Changed Notification arrival callout executes
CallbackTimeoutYears	0	The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.

Copyright © 2026, Oracle and/or its affiliates.

Property Name	Default Value	Description
CallbackTimeoutMonths	0	The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutDays	0	The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutHours	0	The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutMinutes	0	The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.
CallbackTimeoutSeconds	45	The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service.

Composite WSDLs

Composite	Description
OnDemandReadService	Entry point for the OnDemandRead operation. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-ReadingChangedNotification.
CB_ServerService	MultiSpeak WSDL hosted to receive asynchronous ReadingChangedNotification callbacks from the head end system.
MR_ServerProxy	Reference describing the proxy used to invoke the head end system to initiate a On Demand Read.

Copyright © 2026, Oracle and/or its affiliates.

Composite	Description
MR_ServerCallout	Reference describing the process callout used when initiating a On Demand Read.
CB_ServerCallout	Reference describing the process callout used when the asynchronous callback from the head end system arrives.
AuxilaryRoutines	Reference to common helper routines in the Common composite.
DG-ReadingChangedNotificationXAI	A reference to an inbound web service capable of processing asynchronous results from a On Demand Read request.

Process Flow

1. SGG/OUAF invokes InitiateMeterReadingsByMeterID operation using OnDemandReadService.
2. Composite properties are loaded. Local variables are initialized.
3. Test whether the input includes a expiration time
   - If Yes: Use AuxilaryRoutines to compute timeout
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in MR_ServerCallout. Assign updated data to head end request.
6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the results.
7. Invoke InitiateMeterReadingsByMeterID via the MR_ServerProxy to the head end system.
8. If required, execute a process callout in MR_ServerCallout. Assign updated data to head end response.
9. Transform Header and Body head end responses to SGG/OUAF format
10. Reply to SGG/OUAF with synchronous results
11. Check for ErrorObject in the head end response.
12. If error object not found
    - Await asynchronous response from head end for the period of expiration time calculated during start of this flow
      - Receive ReadingChangedNotification from CB_ServiceService.
      - Generate reply with no errors for ReadingChangedNotification.
      - Synchronously reply to the head end.
      - If required, execute a process callout in CB_ServerCallout. Assign updated data to the incoming request.
      - Transform Header and Body of incoming request to OUAF inbound web service format described in DG-ReadingChangedNotificationXAI.
      - If request contains responseURL
        * Invoke the inbound web service on given responseURL with the transformed incoming request.

Copyright © 2026, Oracle and/or its affiliates.

- Else
  - Invoke the inbound web service on definedURL with the transformed incoming request.

Working with Enterprise Manager

Oracle Enterprise Manager (OEM) is useful for troubleshooting and diagnosing issues communicating with the head end system. In particular, security and other communications issues can be discerned.

To locate the instance of a service that is showing a problem, open the **Dashboard** view of the service. Each run instance is time-indexed. The **State** column in the first table contains the most important information for each instance. Completed messages show up with a green check icon. Terminated instances are indicated with a grey stop icon. Instances that are still running are marked as such. Runtime exceptions are in the bottom window.

When you click on the instance ID, a new window opens showing the entire process flow. This view is particularly useful in debugging.

Note the **State** column showing the status of each instance within the flow. SOA processes are made up of several calls to SOA components and web services. The **Instance** column shows a rough ordering of the operations. A typical approach to troubleshooting would find the “lowest” instance of an error or termination. This would be the error that is deepest within the process and is usually the source of the problem. In the above case, the deepest error is in the very first composite for the process. More information can be found by clicking into the link within the **Instance** column.

The level of detail is dependent on the auditing level set on the server. In this case, the error is in the response from the head end system.

OEM also contains a centralized location to control security. On a **Dashboard** screen for a composite, the **Policies** tab shows the OWSM policies attached to the composite. Typically, a log policy is placed on any inbound or outbound communication. Also, all services or references are delivered with attached basic http security policies.

MultiSpeak Implementation

This section lists the subset of MultiSpeak schemas and WSDLs utilized by the Adapter Development Kit.

MultiSpeak Head End System Service Definitions

- CB_Messages.wsdl
- CB_Server.wsdl
- CD_Messages.wsdl
- CD_PortTypes.wsdl
- CD_Server.wsdl
- MR_Messages.wsdl
- MR_PortTypes.wsdl
- MR_Server.wsdl
- OA_Messages.wsdl
- OA_Server.wsdl

Copyright © 2026, Oracle and/or its affiliates.

- OA_ServerCollisions.wsdl
- OD_Messages.wsdl
- OD_PortTypes.wsdl
- OD_Server.wsdl

MultiSpeak Head End XML Schemas

- cpsm.xsd
- gml.xsd
- xlink.xsd
- mspcommon.xsd
- CB_Server.xsd
- CD_Server.xsd
- MR_Server.xsd
- OA_Server.xsd
- OD_Server.xsd

Configuring an Adapter Development Kit Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway to communicate with the smart meters.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the head-end system in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder

Copyright © 2026, Oracle and/or its affiliates.

Inbound Web Service Name	Description
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
DG-ConDisconStChgNotification	Initiate Connect Disconnect Response
DG-OutageDetectionEventNotification	Initiate Outage Detection Response
DG-ReadingChangedNotification	Reading Changed Notification

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace**: No
- **Debug**: No
- **Active**: Yes

Operations

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
DG-COMM	Generic Adapter Commission
DG-DCOMM	Generic Adapter Decommission
DG-INTOUTDET	Initiate Outage Detection Request
DGXAISender	Used for Meter Decommission

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

Copyright © 2026, Oracle and/or its affiliates.

- **HTTP Header**: SOAPAction: **<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Generic/**<SERVICE>**

  where:

  – **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)

  – **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager

  – **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager

  – **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed

  – **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed

  – **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
DG-COMM	Generic Adapter Commission
DG-DCOMM	Generic Adapter Decommission

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the head-end system must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System — Generic:

- **External System**: Generic

Copyright © 2026, Oracle and/or its affiliates.

- **Description**: Generic
- **Outbound Message Types**:

Outbound Message Type	Description	Message Sender
DG-COMM	Generic Adapter Commission	DG-COMM
DG-DCOMM	Generic Adapter Decommission	DG-DCOMM

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: DG-Request xsl
- **Response XSL**: DG-Response.xsl

Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the head-end system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - Generic:

- **Service Provider**: Generic
- **Description**: Generic
- **External Reference ID**: Generic
- **External System**: Generic
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to the head-end system.

The following types of processing methods must be configured for the head-end system service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

Copyright © 2026, Oracle and/or its affiliates.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from the head-end system.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from the head-end system.

UOM Translation

UOM translation processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the head-end system service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	DG-MeterAddNotification	Commission
Device Decommission	Device Removal	DG-MeterRemoveNotification	Decommission
Device Status Check	Device Status Check	DG-InitiateOutageDetection	Get Status
On-Demand Read (Scalar)	On-Demand Read (Scalar)	DG-InitiateMeterByMeterId	Initiate Meter Read by Meter ID
Remote Connect	Remote Connect	DG-InitiateConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	DG-InitiateConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of configuring your adapter is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands.

The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a

Copyright © 2026, Oracle and/or its affiliates.

production environment using the actual head end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoints URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Adapter Development Kit endpoint override DVM (ADK–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Landis+Gyr adapter and the command used with each:

DVM Key	Commands
MR_Server	Device Commissioning Device Decommissioning On-Demand Read (Scalar) On-Demand Read (Interval)
CD_Server	Remote Connect Remote Disconnect
OD_Server	Device Status Check

To define an override Endpoint URI for an adapter built from the Adapter Development Kit, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the ADK–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

   The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

Copyright © 2026, Oracle and/or its affiliates.

Enter an optional note for the session in the **Publish Session** dialog.

Changes take effect immediately upon publishing the session.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Configuring Adapter Development Kit Extendable Lookups

> **Note**
>
> This section applies to to **both** middleware and native implementations of the Smart Grid Gateway Adapter Development Kit,

This section outlines some of the extendable lookups that must be configured for use with a customized adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system.

Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system.
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

Head-End UOM Code to Standard UOM Mapping

Usage received from the head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The head-end system UOM Code to Standard UOM Mapping extendable lookup is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Each value defined for the head-end system UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the head-end system
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)

Interval Status Code to Condition Mapping

Interval usage received from the head-end system can include interval status codes that indicate the status or condition of the interval value. These interval status codes must be

Copyright © 2026, Oracle and/or its affiliates.

mapped to standard condition codes in the system. The Generic Interval Status Code to Condition Mapping extendable lookup is used to determine how to map head-end system interval status codes to standard status codes when receiving usage from the head-end system.

Each value defined for the Generic Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The head-end system interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

Other Extendable Lookups

Business Object Name	Description
DG-CDReasonCodeLookup	ADK - Connect Disconnect Reason Code
DG-GenericTimeUnits	ADK - Time Units
DG-LoadActionCodeLookup	ADK - Load Action Code
DG-OutageEventTypeLookup	ADK - Outage Event Type
DG-ServiceTypeMappingLookup	ADK - Service Type Mapping

Using the Adapter Development Kit Test Harness

Oracle Utilities Smart Grid Gateway Adapter Development Kit includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. This section describes the test harness and its components.

- Locating the WSDL for the Test Harness
- Web Services
- Sample Meters File

Locating the WSDL for the Test Harness

Follow these procedures to locate the Adapter Development Kit test harness WSDL:

How to Use Enterprise Manager to Locate the WSDL

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the test harness composite:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the composite:
3. Click the UtilService WSDL URL link to see the WSDL in the browser, or right click and save it to your machine

Depending on your requirements, it may be necessary to download the associated schema found in the wsdl:types section. The URL can be pasted into a browser tab and downloaded in

Copyright © 2026, Oracle and/or its affiliates.

the same manner as the WSDL. The main schema has imported schemas that may also be required.

How to Use a Direct URL to locate the WSDL

The WSDL can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the following form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}?WSDL
```

So by default, the test harness WSDL can be found at

```
http://{server name}:{port number}/soa-infra/services/DG_Test/DGTestHarness/
UtilService?WSDL
```

Web Services

This section describes the web services included in the Adapter Development Kit test harness BPEL composite.

- General Services
- Locate Meter Services
- Meter Administration Services
- Meter Attribute Administration Services

General Services

This section describes the general services of the Adapter Development Kit test harness composite.

LoadMeterIndex

This web service loads the data store from the internal file. By default if the store is already in memory, it will *not* reload. This behavior can be overridden with the forceReload parameter.

Input: LoadMeterIndexInput

Part: payload

Element: LoadMeterIndexRequest

Parameter	Description
forceReload	A switch telling the system whether to reload the meter index from the configuration file. Default value is false.

Output: LoadMeterIndexOutput

Part: payload

Element: LoadMeterIndexResult

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description
loaded	A boolean value for whether or not the index was reloaded from the configuration file

Fault: UtilityFault (see UtilityFault for more details).

ViewAuditTrail

This web service returns the audit log for the entire session.

Input: ViewAuditTrailInput

Part: payload

Element: ViewAuditTrailRequest

Parameters: This is an empty request. There are no parameters.

Output: ViewAuditTrailOutput

Part: payload

Element: ViewAuditTrailResult

This element is an entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing what changed.

Fault: UtilityFault (see UtilityFault for more details).

UtilityFault

Fault with similar mapping to SGG/OUAF faults:

Typically, the faultCode, faultString, faultActor, and detail/text elements will be populated.

Locate Meter Services

This section describes the locate meter web services of the Adapter Development Kit test harness composite.

FindMeters

This web service queries the data store for one or more meters. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input: FindMetersInput

Part: payload

Element: FindMetersRequest

Parameter	Description
id	The meter ID for which to search

Copyright © 2026, Oracle and/or its affiliates.

Parameter	Description
isRegex	The provided id can be a regex value when this parameter is true. Hint: to search for all meters in the system, use ".*" for the ID.

Output: FindMetersOutput

Part: payload

Element: FindMetersResult

Zero or more meter objects can be returned from the search

Fault: See UtilityFault. Unlike other methods, FindMeters does not throw an exception if the meter is not found.

IsMeterDefined

This web service queries whether a particular meter is defined in the data store.

Input: IsMeterDefinedInput

Part: payload

Element: IsMeterDefinedRequest

Parameter	Description
id	The meter ID for which to search

Output: IsMeterDefinedOutput

Part: payload

Element: IsMeterDefinedResult

Whether or not the provided ID is part of the index.

Fault: See UtilityFault. Thrown when meter id is not found.

GetMeter

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input: GetMeterInput

Part: payload

Element: GetMeterRequest

Parameter	Description
id	The meter ID for which to search

Output: GetMeterOutput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: GetMeterResult

The meter object requested by the ID.

Fault: See UtilityFault. Thrown when meter id is not found.

Meter Administration Services

This section describes the meter administration services of the Adapter Development Kit test harness composite.

AddMeters

This web service adds a set of meters to the in-memory data store. This will not permanently add it to the control file.

Input: AddMetersInput

Part: payload

Element: AddMetersRequest

Parameter	Description
id	The identification code for the meter.
utility	An informational string.
serviceType	One of the valid ServiceType values (see schema). "Electric" is the only option at this time.
isCommissioned	Whether or not the meter is in a commissioned state.
loadActionCode	One of the possible LoadActionCode values used in Connect and Disconnect (see schema).
outageEventType	One of the possible OutageEventType values used in Device Status Check (see schema).
executionStatus	One of the possible ExecutionStates (see schema). These values control how the meter will respond to commands.
updateIfExisting	Whether or not to update the meter with the provided values if it already exists in the index.
Comment	An informational string describing the purpose of the meter.
Channels	A listing of unit of measures supported by this meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output: AddMetersOutput

Part: payload

Element: AddMetersResult

Whether or not each meter was added to the index.

Fault: See UtilityFault.

Copyright © 2026, Oracle and/or its affiliates.

RemoveMeter

This web service removes a meter from the in-memory data store. This will not permanently remove it from the control file.

Input: RemoveMeterInput

Part: payload

Element: RemoveMeterRequest

Parameter	Description
id	The ID for the meter to be removed.

Output: RemoveMeterOutput

Part: payload

Element: RemoveMeterResult

Whether or not the meter was removed from the index.

Fault: See UtilityFault. Thrown when meter id is not found.

AddMeterChannel

This web service adds a new channel to a single meter.

Input: AddMeterChannelInput

Part: payload

Element: AddMeterChannelRequest

Parameter	Description
id	The identification code for the meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output: AddMeterChannelOutput

Part: payload

Element: AddMeterChannelResult

Whether or not the channel was added to the index.

Fault: See UtilityFault. Thrown when meter id is not found.

RemoveMeterChannel

This web service removes a Channel from a meter.

Input: RemoveMeterChannelInput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: RemoveMeterChannelRequest

Parameter	Description
id	The ID for the meter to be removed.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.

These three parameters are combined to locate a unique channel

Output: RemoveMeterChannelOutput

Part: payload

Element: RemoveMeterChannelResult

Whether or not the channel was removed from the meter.

Fault: See UtilityFault. Thrown when meter id is not found.

ReadScalarMeter

This web service generates a scalar reading for each channel of a given meter.

Input: ReadScalarMeterInput

Part: payload

Element: ReadScalarMeterRequest

Parameter	Description
id	The ID for the meter to be read.

Output: ReadScalarMeterOutput

Part: payload

Element: ReadScalarMeterResult

Zero or more scalar readings for the given meter.

Parameter	Description
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.
value	A random number representing the scalar reading.

Fault: See UtilityFault. Thrown when meter id is not found.

Meter Attribute Administration Services

Copyright © 2026, Oracle and/or its affiliates.

This section describes the meter administration services of the Adapter Development Kit test harness composite.

GetOutageEventType

This web service queries the outage event type for a given meter. The OutageEventType is used by DeviceStatusCheck.

Input: GetOutageEventTypeInput

Part: payload

Element: GetOutageEventTypeRequest

Parameter	Description
id	The ID for the meter for which the OutageEventType should be retrieved.

Output: GetOutageEventTypeOutput

Part: payload

Element: GetOutageEventTypeResult

The value of the OutageEventType attribute for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetOutageEventType

This web service updates the outage event type for a given meter.

Input: SetOutageEventTypeInput

Part: payload

Element: SetOutageEventTypeRequest

Parameter	Description
id	The ID for the meter for which the OutageEventType should be set.
value	The new value of OutageEventType to set on the meter.

Output: SetOutageEventTypeOutput

Part: payload

Element: SetOutageEventTypeResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See UtilityFault. Thrown when meter id is not found.

GetLoadActionCode

This web service queries the load action code for a given meter. This is the Connect/Disconnect behavior.

Input: GetLoadActionCodeInput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: GetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the LoadActionCode should be retrieved.

Output: GetLoadActionCodeOutput

Part: payload

Element: GetLoadActionCodeResult

The value of the LoadActionCode attribute for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetLoadActionCode

This web service updates the load action code for a given meter.

Input: SetLoadActionCodeInput

Part: payload

Element: SetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the LoadActionCode should be set.
value	The new value of LoadActionCode to set on the meter.

Output: SetLoadActionCodeOutput

Part: payload

Element: SetLoadActionCodeResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See UtilityFault. Thrown when meter id is not found.

IsCommissioned

This web service queries the commissioning status for a given meter. This is the Commission/Decommission behavior.

Input: IsCommissionedInput

Part: payload

Element: IsCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be retrieved.

Output: IsCommissionedOutput

Copyright © 2026, Oracle and/or its affiliates.

Part: payload

Element: IsCommissionedResult

The value of the Commissioned status attribute for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetCommission

This web service updates the commissioning status for a given meter.

Input: SetCommissionedInput

Part: payload

Element: SetCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be set.
value	The new value of Commissioned status to set on the meter.

Output: SetCommissionedOutput

Part: payload

Element: SetCommissionedResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See UtilityFault. Thrown when meter id is not found.

GetExecutionStatus

This web service queries the status of the property controlling the overall execution of the command.

Input: GetExecutionStatusInput

Part: payload

Element: GetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be retrieved.

Output: GetExecutionStatusOutput

Part: payload

Element: GetExecutionStatusResult

The value of the ExecutionStatus attribute for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

Copyright © 2026, Oracle and/or its affiliates.

SetExecutionStatus

This web service updates the property controlling the overall completion of the command.

Input: SetExecutionStatusInput

Part: payload

Element: SetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be set.
value	The new value of ExecutionStatus to set on the meter.

Output: SetExecutionStatusOutput

Part: payload

Element: SetExecutionStatusResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See UtilityFault. Thrown when meter id is not found.

Sample Meters File

The Adapter Development Kit includes an XML file and schema that can be used for configuring one or more meters for use with the test harness. The file, metersdb.xml, is located in the Test/DGTestHarness directory and can be edited with an appropriate XML editor such as XML Spy. This section describes the attributes in the metersdb.xml file.

The meterdb.xml file contains one or more **Meter** elements that have the following attributes:

Meter Attribute	Definition
id	The meter identifier. This value should match the amiDeviceID setting in the Oracle Utilities Application Framework.
utility	This is informational only.
serviceType	This is informational only.
ServiceType	An enumeration that is described in the schema.
isCommissioned	A boolean value that describes whether or not the meter is commissioned or decommissioned. The associated MultiSpeak commands are MeterAddNotification and MeterRemoveNotification.
loadActionCode	This value is for the Connect and Disconnect commands and has enumeration values described in the schema. The value is returned in the CDStatesChangedNotification MultiSpeak command.
outageEventType	This value is used by DeviceStatusCheck and gives the status returned by the ODEventNotification call. Its enumeration values are described in the schema.

Copyright © 2026, Oracle and/or its affiliates.

Meter Attribute	Definition
executionStatus	This value does not reflect a MultiSpeak command, but is instead intended to give a state of the operation. There are four valid values: **Success:** When a meter has this status the operation will complete without error. **ResponseTimeout:** When a meter has this status an asynchronous reply will never arrive (not relevant for Commission/Decommission commands). **SyncOperationFailure:** When a meter has this status the initial communication to the simulated head-end system will produce an error. **AsyncOperationFailure:** When a meter has this status, the asynchronous callback from the head-end system will arrive, but will indicate an error (not relevant for Commission/Decommission commands).

A meter can also contain the following elements:

- **Comment:** A field which is for informational use only and is meant to indicate the purpose of the meter.
- **Channels:** Used in reading the meter for On Demand Read commands. A channel contains the following attributes:

Channel Attribute	Definition
uomCode	The units of measure code that should be returned when the meter is read.
uomName	The units of measure name that should be returned when the meter is read.
description	A longer form. When the meter is read, a random number is generated for this value.
decimals	A value to indicate how many places to the right of the decimal that random number should have.

The Adapter Development Kit Native Format

> **Note**
>
> The Adapter Development Kit Native Format can be used with **both** middleware and native Smart Grid Gateway Adapter implementations.

The Adapter Development Kit supports loading usage and event data exported from the AMI head-end system in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). Processing of the ADK native format is supported by the following OSB projects:

- **SGG-DG-SEEDER-BASE**
- **SGG-DG-SEEDER-CM**

Copyright © 2026, Oracle and/or its affiliates.

Refer to Initial Measurements and Device Events and OSB Project Summary for more information about these OSB projects.

Format Details

The ADK native format is an XML format that contains zero or more initial measurements and/or device events, as follows:

- The collection of initial measurements and/or device events are encapsulated within in an <SGGIMDsEvents> element.
- Each initial measurement or device event is defined in the format of the initial measurement and device event seeder business objects, encapsulated by the following elements:
  - **Initial Measurements**: <D1-InitialLoadIMD>
  - **Device Events**: <D1–DeviceEventSeeder>
- The format can support any number of initial measurements and/or device events.
- Initial measurements can be either scalar or interval measurements.

See Adapter Development Kit Native Format Example for an example of this format. See Adapter Development Kit Native Format Schema for the native format XML schema.

Adapter Development Kit Native Format Example

The following is an example of the ADK native format. This example contains 2 initial measurements (1 scalar and 1 interval) and 2 device events (PowerOutage and PowerRestored).

```
<?xml version="1.0" encoding="UTF-8" ?>
<SGGIMDsEvents xmlns="http://oracle.com/SGGIMDsEvents">
  <D1-InitialLoadIMD dateTimeTagFormat="xsd" xmlns="">
    <preVEE>
      <imdType>D1IL</imdType>
      <mcIS>D1SC</mcIS>
      <externalId>da_164_Scalar_withMetadata.xml-2015-09-10-11-19-03-297</
externalId>
      <dvcIdN>DL_06</dvcIdN>
      <externalUOM>KWH</externalUOM>
      <enDt>2005-01-01T00:00:00Z</enDt>
      <enQty>4544</enQty>
    </preVEE>
    <serviceProviderExternalId>Itron</serviceProviderExternalId>
  </D1-InitialLoadIMD>
  <D1-InitialLoadIMD dateTimeTagFormat="xsd" xmlns="">
    <preVEE>
      <imdType>D1IL</imdType>
      <mcIS>D1IN</mcIS>
      <externalId>da_164_Multile_withMetadata_A.xml-2015-09-10-11-19-03-297</
externalId>
      <dvcIdN>ZZ-D-OSB-INT-ITRON-0002</dvcIdN>
      <spi>900</spi>
      <externalUOM>KWH</externalUOM>
      <mcm>725</mcm>
      <stDt>2003-01-01T23:45:00Z</stDt>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <enDt>2003-01-02T00:00:00Z</enDt>
      <msrs>
        <mL>
          <s>1</s><q>4722</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>CHardwareFailure</st>
            </stsL>
            <stsL>
              <s>2</s>
              <st>COverflow</st>
            </stsL>
          </sts>
        </mL>
      </msrs>
    </preVEE>
    <serviceProviderExternalId>Itron</serviceProviderExternalId>
  </D1-InitialLoadIMD>
  <D1-DeviceEventSeeder dateTimeTagFormat="xsd" xmlns="">
    <externalSenderId>Itron</externalSenderId>
        <deviceIdentifierNumber>ZZ-D-OSB-INT-ITRON-0002</
deviceIdentifierNumber>
    <externalEventName>PowerOutage</externalEventName>
    <eventDateTime>2001-02-02T00:00:00Z</eventDateTime>
    <eventInformation>
      <externalEventIdentifier>2147483642</externalEventIdentifier>
      <externalStatusValue>string</externalStatusValue>
      <externalEventCategory>Communication</externalEventCategory>
    </eventInformation>
  </D1-DeviceEventSeeder>
  <D1-DeviceEventSeeder dateTimeTagFormat="xsd" xmlns="">
    <externalSenderId>Itron</externalSenderId>

<externalSourceIdentifier>da_164_Multile_withMetadata_A.xml-2015-09-10-11-19-0
3-297</externalSourceIdentifier>
    <deviceIdentifierNumber>ZZ-D-OSB-INT-ITRON-0002</deviceIdentifierNumber>
    <externalEventName>PowerRestored</externalEventName>
    <eventDateTime>2001-02-02T01:00:00Z</eventDateTime>
    <eventInformation>
      <externalEventIdentifier>2147483642</externalEventIdentifier>
      <externalStatusValue>string</externalStatusValue>
      <externalEventCategory>Communication</externalEventCategory>
    </eventInformation>
  </D1-DeviceEventSeeder>
</SGGIMDsEvents>
```

Adapter Development Kit Native Format Schema

The following is the XML schema of the ADK native format.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```


Copyright © 2026, Oracle and/or its affiliates.

```
  <xsd:element name="SGGIMDsEvents">
<xsd:complexType>
      <xsd:all>
 <xsd:element name="D1-DeviceEventSeeder" minOccurs="0">
  <xsd:complexType>
   <xsd:all>
    <xsd:element name="deviceEventId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="14" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="bo" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="boStatus" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="sender" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalSenderId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="36" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deviceEventType" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalEventName" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="254" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="eventDateTime" nillable="true" type="xsd:dateTime"
```


Copyright © 2026, Oracle and/or its affiliates.

```
minOccurs="0"/>
    <xsd:element name="eventEndDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="externalTimeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="20" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deviceId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deprecatedDeviceId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="creationDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="statusUpdateDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
    <xsd:element name="statusReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="rawEventInformation" type="xsd:anyType"
minOccurs="0" />
    <xsd:element name="externalSourceIdentifier" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="120" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="eventInformation" minOccurs="0">
     <xsd:complexType>
      <xsd:all>
       <xsd:element name="externalEventIdentifier" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:maxLength value="30" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="externalEventCategory" minOccurs="0">
        <xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalEventSeverity" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalDeviceType" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalServiceLocationId" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalCommunicationModuleIdentifier"
minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalGatewayIdentifier" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalStatusValue" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalStatusDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
        <xsd:element name="externalCommandId" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:element>
    <xsd:element name="externalEventDescription" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalEventReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalStatusReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="sourceTimeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="10" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="timeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="10" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="dateTimesInStandard" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:enumeration value="D1NO" />
       <xsd:enumeration value="D1YS" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
   </xsd:all>
  </xsd:complexType>
 </xsd:element>
 <xsd:element name="version" minOccurs="0">
  <xsd:simpleType>
   <xsd:restriction base="xsd:decimal">
    <xsd:minExclusive value="-99999" />
    <xsd:maxExclusive value="99999" />
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:element>
 <xsd:element name="deviceIdentifierNumber" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="newDeviceEvent" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="14" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="nextRetryDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="retryUntilDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
    <xsd:element name="processData" minOccurs="0">
     <xsd:complexType>
      <xsd:all>
       <xsd:element name="errorEncountered" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="D1NO" />
          <xsd:enumeration value="D1YS" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="dateTimesInStandard" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="D1NO" />
          <xsd:enumeration value="D1YS" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="logs" minOccurs="0">
        <xsd:complexType>
         <xsd:all>
          <xsd:element name="logsList" minOccurs="0" >
           <xsd:complexType>
            <xsd:all>
             <xsd:element name="logsEntry" minOccurs="0">
              <xsd:complexType>
               <xsd:all>
                <xsd:element name="sequence" minOccurs="0">
                 <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                   <xsd:minExclusive value="-999" />
                   <xsd:maxExclusive value="999" />
                  </xsd:restriction>
                 </xsd:simpleType>
                </xsd:element>
                <xsd:element name="mo" minOccurs="0">
                 <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:maxLength value="12" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue1" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue2" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue3" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue4" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue5" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="logEntryType" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="D1TD" />
                    <xsd:enumeration value="F1CR" />
                    <xsd:enumeration value="F1ER" />
                    <xsd:enumeration value="F1EX" />
                    <xsd:enumeration value="F1ST" />
                    <xsd:enumeration value="F1SY" />
                    <xsd:enumeration value="F1TD" />
                    <xsd:enumeration value="F1US" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="logDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
                 <xsd:element name="boStatus" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="12" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="description" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="60" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="user" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="8" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="logMessage" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="4000" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="characteristicType" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="20" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="characteristicValue" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="16" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="adhocValue" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="254" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue1" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue2" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue3" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue4" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue5" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageCategory" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:decimal">
   <xsd:minExclusive value="-99999" />
   <xsd:maxExclusive value="99999" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageNumber" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:decimal">
   <xsd:minExclusive value="-99999" />
   <xsd:maxExclusive value="99999" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageParm1" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="30" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageParm2" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="30" />
  </xsd:restriction>
 </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      </xsd:element>
      <xsd:element name="messageParm3" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm4" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm5" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm6" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm7" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm8" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm9" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
     </xsd:all>
    </xsd:complexType>
   </xsd:element>
  </xsd:all>
   <xsd:attribute name="action" type="xsd:string" use="optional" />
 </xsd:complexType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          </xsd:element>
         </xsd:all>
        </xsd:complexType>
       </xsd:element>
      </xsd:all>
     </xsd:complexType>
    </xsd:element>
   </xsd:all>
  </xsd:complexType>
 </xsd:element>
  <xsd:element name="D1-InitialLoadIMD" minOccurs="0">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="initialMeasurementDataId" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="14"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="bo" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="fromDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="toDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="boStatus" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="12"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="statusReason" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="dataSource" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1RE"/>
              <xsd:enumeration value="D1ST"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="timeZone" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
              <xsd:maxLength value="10"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="creationDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="comments" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="254"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isTraceOn" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isIntervalDateTimePopulated" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="serviceProvider" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isAutomatedRetry" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="nextRetryDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="retryUntilDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="veeGroupRole" nillable="true" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1ES"/>
              <xsd:enumeration value="D1IL"/>
              <xsd:enumeration value="D1IS"/>
              <xsd:enumeration value="D1MO"/>
              <xsd:enumeration value="D1PR"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="startReadingDateTime" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="startReading" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:decimal">
              <xsd:minExclusive value="-9999999999.999999"/>
              <xsd:maxExclusive value="9999999999.999999"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="startReadingCondition" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="preVEE" minOccurs="0">
          <xsd:complexType>
            <xsd:all>
              <xsd:element name="simdId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="14"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="dvcIdN" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="mcId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="12"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="mcIdN" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="uom" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalUOM" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="tou" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalTOU" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="sqi" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalSQI" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="stDt" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
              <xsd:element name="stQty" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                    <xsd:minExclusive value="-9999999999.999999"/>
                    <xsd:maxExclusive value="9999999999.999999"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="enDt" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
              <xsd:element name="enQty" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                    <xsd:minExclusive value="-9999999999.999999"/>
                    <xsd:maxExclusive value="9999999999.999999"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="imdType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1ES"/>
      <xsd:enumeration value="D1GA"/>
      <xsd:enumeration value="D1IL"/>
      <xsd:enumeration value="D1MO"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="mcIS" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1IN"/>
      <xsd:enumeration value="D1SC"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="inShift" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1NS"/>
      <xsd:enumeration value="D1SH"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="mcm" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minExclusive value="-999999999999.999999"/>
      <xsd:maxExclusive value="999999999999.999999"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="nd" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minExclusive value="-999999999"/>
      <xsd:maxExclusive value="999999999"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="tz" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="externalTimeZone" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="20"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="spi" nillable="true" type="xsd:int"
minOccurs="0"/>
              <xsd:element name="ccond" type="xsd:string" minOccurs="0"/>
              <xsd:element name="sts" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="stsL" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="s" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-99999"/>
                                <xsd:maxExclusive value="99999"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                          <xsd:element name="st" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:maxLength value="6"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="msrs" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="mL" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="s" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-99999"/>
                                <xsd:maxExclusive value="99999"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                          <xsd:element name="dt" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
                          <xsd:element name="q" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-9999999999.999999"/>
                                <xsd:maxExclusive value="9999999999.999999"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="ue" minOccurs="0">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:string">
                                  <xsd:enumeration value="D1UE"/>
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="fc" type="xsd:string"
minOccurs="0"/>
                            <xsd:element name="r" minOccurs="0">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:decimal">
                                  <xsd:minExclusive value="-9999999999.999999"/>
                                  <xsd:maxExclusive value="9999999999.999999"/>
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="rc" type="xsd:string"
minOccurs="0"/>
                            <xsd:element name="sts" minOccurs="0">
                              <xsd:complexType>
                                <xsd:sequence>
                                  <xsd:element name="stsL" minOccurs="0"
maxOccurs="unbounded">
                                    <xsd:complexType>
                                      <xsd:sequence>
                                        <xsd:element name="s" minOccurs="0">
                                          <xsd:simpleType>
                                            <xsd:restriction base="xsd:decimal">
                                              <xsd:minExclusive value="-99999"/>
                                              <xsd:maxExclusive value="99999"/>
                                            </xsd:restriction>
                                          </xsd:simpleType>
                                        </xsd:element>
                                        <xsd:element name="st" minOccurs="0">
                                          <xsd:simpleType>
                                            <xsd:restriction base="xsd:string">
                                              <xsd:maxLength value="6"/>
                                            </xsd:restriction>
                                          </xsd:simpleType>
                                        </xsd:element>
                                      </xsd:sequence>
                                    </xsd:complexType>
                                  </xsd:element>
                                </xsd:sequence>
                              </xsd:complexType>
                            </xsd:element>
                          </xsd:sequence>
                        </xsd:complexType>
                      </xsd:element>
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:all>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rawData" type="xsd:anyType" minOccurs="0"/>
<xsd:element name="processData" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="isShiftedStartEnd" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isShiftedIntervals" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isTimeZoneConverted" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isErrorEncountered" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="servicePointId" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="12"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="installationConstant" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:decimal">
            <xsd:minExclusive value="-999999.999999"/>
            <xsd:maxExclusive value="999999.999999"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="deviceId" minOccurs="0">
        <xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:restriction base="xsd:string">
                      <xsd:maxLength value="12"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="allowNonZeroIntervalsForEstimate"
minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="D1NO"/>
                      <xsd:enumeration value="D1YS"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="disableReEstimate" minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="D1NO"/>
                      <xsd:enumeration value="D1YS"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="reEstimationActivity" minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:maxLength value="14"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="logs" minOccurs="0">
                  <xsd:complexType>
                    <xsd:all>
                      <xsd:element name="logsList" minOccurs="0">
                        <xsd:complexType>
                          <xsd:sequence>
                            <xsd:element name="logsEntry" minOccurs="0">
                              <xsd:complexType>
                                <xsd:sequence>
                                  <xsd:element name="sequence" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:decimal">
                                        <xsd:minExclusive value="-999"/>
                                        <xsd:maxExclusive value="999"/>
                                      </xsd:restriction>
                                    </xsd:simpleType>
                                  </xsd:element>
                                  <xsd:element name="mo" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:string">
                                        <xsd:maxLength value="12"/>
                                      </xsd:restriction>
                                    </xsd:simpleType>
                                  </xsd:element>
                                  <xsd:element name="pkValue1" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue2" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue3" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue4" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue5" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="logEntryType"
minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="D1TD"/>
                                    <xsd:enumeration value="F1CR"/>
                                    <xsd:enumeration value="F1ER"/>
                                    <xsd:enumeration value="F1EX"/>
                                    <xsd:enumeration value="F1ST"/>
                                    <xsd:enumeration value="F1SY"/>
                                    <xsd:enumeration value="F1TD"/>
                                    <xsd:enumeration value="F1US"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="logDateTime"
nillable="true" type="xsd:dateTime" minOccurs="0"/>
                              <xsd:element name="boStatus" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="12"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                <xsd:element name="description" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="60"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="user" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="8"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="logMessage" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="4000"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="characteristicType"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="20"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="characteristicValue"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="16"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="adhocValue" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="254"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue1" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue2" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue3" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue4" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue5" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageCategory"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:decimal">
                                      <xsd:minExclusive value="-99999"/>
                                      <xsd:maxExclusive value="99999"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageNumber"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:decimal">
                                      <xsd:minExclusive value="-99999"/>
                                      <xsd:maxExclusive value="99999"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm1"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm2"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:element>
                                <xsd:element name="messageParm3"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm4"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm5"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm6"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm7"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm8"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm9"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:element>
                              </xsd:sequence>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:all>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="deviceEventTypes" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="deviceEventTypesList" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="deviceEventType" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:maxLength value="30"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:all>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="boStatusDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="isReprocessPerformed" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="serviceProviderExternalId" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="60"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="syncIMDOtherInfo" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="meterReadSource" type="xsd:string"
minOccurs="0"/>
              <xsd:element name="reviewHiLo" type="xsd:string" minOccurs="0"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="version" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:decimal">
              <xsd:minExclusive value="-99999"/>
              <xsd:maxExclusive value="99999"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>

 </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Smart Grid Gateway Adapter Native Implementations

This section describes Smart Grid Gateway native implementations. Native implementations can be used with **either** on-premises implementations of Oracle Utilities Smart Grid Gateway, or with Oracle Utilities cloud services such as Oracle Utilities Meter Solution Cloud Service or Oracle Utilities Customer Cloud Service.

- Supported Head End Systems
- Payload Processing
- Smart Meter Commands
- Master Configurations
- Native Implementation Configuration Checklists

Supported Head End Systems

Oracle Utilities Meter Solution products support integration with a number of vendor-specific head end systems via Oracle Smart Grid Gateway Adapters. The table below lists the supported versions of head end system for each Adapter available with Oracle Utilities Meter Solution products.

Adapter	Protocol	Head End System Version
Adapter for Itron OpenWay	Proprietary	Itron OpenWay 3.70, 3.90, 4.1, 6.10 OpenWay Collection Manager 5.4
Adapter for Itron Field Collection System	Proprietary	Field Collection System 4.63

Copyright © 2026, Oracle and/or its affiliates.

Adapter	Protocol	Head End System Version
Adapter for Landis+Gyr	MultiSpeak 3.0 CIM 1.0v2 CIM 2.0	Gridstream Command Center 7.5, 8.1, 8.4, 8.5, 9.0.4
Adapter for Sensus RNI	MultiSpeak 4.1	Sensus RNI 4.4, 4.5.1, 4.10, 4.12, 4.14, 4.15, 4.16
Adapter for Silver Spring Networks (Itron AMM)	Proprietary MultiSpeak 4.1	UtilityIQ Version 4.4, 4.7, 4.10, 4.14, 5.0, 5.1, 5.2, 6.1
MV-90 Adapter for Itron	Binary file format	N/A SGG supports the .mv9 binary, mainframe data format.

Payload Processing

In native implementations, Oracle Smart Grid Gateway adapters process usage and device event payloads via batch processing. Files are sent from a head end system to a file system (for on-premises implementations) or an Oracle Cloud Object Storage location (for cloud services), where they are processed via batch processes.

> **Note**
>
> The total length of the file name for payload files, including the name, delimiter, and extension, should be no greater than 96 characters.

This process requires set up and configuration of the following:

- **Key Rings** and **Key Pairs**: Records used to link an Oracle Cloud Object Storage location to records in Oracle Utilities cloud services. Used with cloud services only.
- **Object Storage Locations**: One or more Oracle Object Storage buckets where payload files are sent from the head end system for processing. Used with cloud services only.
- **File Storage Configuration Extendable Lookup Values**: An extendable lookup value that defines the file location used for payload processing and allows it to be referenced by batch processes
- **Head End Systems**: A record that defines how specific types of payload processing are handled, such the business object to use to create initial measurements and device events
- **Adapter Extendable Lookups**: Lookup value that define details used by data mapping and processing
- **SGG Payload Processing Configuration Extendable Lookup Values**: An extendable lookup value that defines payload processing details for a specific head end system
- **Payload Processing Batch Controls**: A batch control used for payload process (based on a template batch control provided with the base package)

See the following sections below for more details about each of these.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Itron OpenWay Adapter uses an alternative method for processing usage and device event payloads. See Itron OpenWay Payload Processing for more information.

- Creating Key Rings and Pairs
- Creating Object Storage Locations
- Creating File Storage Extendable Lookup Values
- Creating Head End Systems
- Configuring Adapter Extendable Lookups
- Creating SGG Payload Processing Extendable Lookup Values
- Creating Payload Processing Batch Controls
- Mapping Additional Fields
- Itron OpenWay Payload Processing
- Adapter Development Kit Custom Payload Processing
- Adapter Development Kit Common Information Model (CIM) Payload Processing

Creating Key Rings and Pairs

> **Note**
>
> Key Rings and Pairs are used with cloud services **only**.

When using Smart Grid Gateway adapters with cloud services, connection between the Oracle Utilities cloud service and Oracle Object Storage requires an API signature key. See **API Key Management** in the *Oracle Utilities Cloud Services Object Storage Setup Guide* for more information.

API key rings and key pairs are maintained in the **Key Ring** portal. This portal contains the following zones:

- **Key Ring**: Displays basic information about the key ring
- **Key Pairs**: Displays a list of key pairs for the current key ring

Key rings are defined by the following:

- **Key Ring**: A unique code for the key ring
- **Key Ring Class**: Signature (default)
- **Status**: The current status of the key ring. Note: Key pairs can only be generated for Active key rings. Once a key ring has been deactivated, you can no longer create key pairs for that ring.
- **Description**: A name for the key ring (this will be referenced in the File Location extendable lookup)

Once the key ring is created, you need to generate and activate the key pair. Click **Generate Key** to generate a key pair for the key ring.

Copyright © 2026, Oracle and/or its affiliates.

Key Pairs are defined by the following:

- **Sequence**: The sequence of the key pair (the order in which the key pair was created)
- **Creation Date/Time**: The date/time when the key pair was created
- **Key Status**: The current status of the key pair. Key pairs are inactive when first created.
- **Public Key**: Click **View** to open a dialog box containing the public key.
- **Action**: Click **Activate** to activate an inactive key pair.

Once the key ring and key pair have been created, the key should be pasted into the **User API Key** in **Oracle Identification and Access Management** (IAM) See the **User API Keys** section in the **Security and Access Management** section of the **Managing Object Storage** chapter in the *Oracle Cloud Infrastructure Services* documentation. To copy the public key click **View** in the **Actions** column in the **Key Pairs** zone, and select and copy the text in the **View Public Key** dialog box.

The last step is to activate the key. Click **Activate** in the **Actions** column in the **Key Pairs** zone. A dialog box opens displaying the following message: “Warning(s): Activating a key assumes that you have already registered the public key with the appropriate third parties. Press Cancel to abort.” Note: Be sure to copy the public key to **Oracle Identification and Access Management** before activating it. Click **OK** to activate the key. The **Key Status** column will change to “Active”.

Creating Object Storage Locations

> **Note**
>
> Object Storage Locations are used with cloud services **only**.

When using Smart Grid Gateway adapters with cloud services, you need to create one or more buckets in Oracle Object Storage where payload files will be sent for processing. Refer to the *Oracle Utilities Cloud Services Object Storage Setup Guide* and the *Oracle Cloud Infrastructure* documentation for more information about creating buckets in Oracle Object Storage.

After creating the buckets in Oracle Object Storage, make a note of the following, as this information is needed in later configuration steps:

- Namespace
- User Oracle Cloud ID (User OCID)
- Tenancy Oracle Cloud ID (Tenancy OCID)
- Compartment Oracle Cloud ID (Compartment OCID)

Suggested Buckets for Payload Processing

This section outlined suggested sets of Object Storage buckets for each of the Smart Grid Gateway adapters.

General Guidelines

As a general rule, each adapter should have separate buckets for each of the following:

- Usage payloads

Copyright © 2026, Oracle and/or its affiliates.

- Usage archive
- Usage errors
- Device Event payloads
- Device Event archive
- Device Event errors

If an adapter uses payloads in more than one format (such as in CSV and XML formats), you should also consider creating separate buckets for each file format type.

> **Note**
>
> If possible, archive buckets should be created in an “Archive” storage tier.

Adapter Development Kit

The table below lists suggested Object Storage buckets for the Adapter Development Kit:

Bucket Name	Storage Tier	Smart Grid Gateway Description
adk-csv	Standard	SGG ADK Adapter - CSV input
adk-csv-arch	Archive	SGG ADK Adapter - CSV archive
adk-csv-error	Standard	SGG ADK Adapter - CSV errors
adk-seeder	Standard	SGG ADK Adapter - IMD Seeder input
adk-seeder-arch	Archive	SGG ADK Adapter - IMD Seeder archive
adk-seeder-error	Standard	SGG ADK Adapter - IMD Seeder errors
adk-xml	Standard	SGG ADK Adapter - XML input
adk-xml-arch	Archive	SGG ADK Adapter - XML archive
adk-xml-error	Standard	SGG ADK Adapter - XML errors

Adapter for Itron OpenWay

The table below lists suggested Object Storage buckets for the Adapter for Itron OpenWay:

Bucket Name	Storage Tier	Smart Grid Gateway Description
itronexception	Standard	SGG Itron OW Adapter - event input
itronexception-arch	Archive	SGG Itron OW Adapter - event archive
itronexception-error	Standard	SGG Itron OW Adapter - event errors
itronxml	Standard	SGG Itron OW Adapter - XML input
itronxml-arch	Archive	SGG Itron OW Adapter - XML archive
itronxml-error	Standard	SGG Itron OW Adapter - XML errors

MV90 Adapter for Itron

The table below lists suggested Object Storage buckets for the MV90 Adapter for Itron

Bucket Name	Storage Tier	Smart Grid Gateway Description
mv90-usage	Standard	SGG MV90 Adapter - usage input

Copyright © 2026, Oracle and/or its affiliates.

Bucket Name	Storage Tier	Smart Grid Gateway Description
mv90-usage-arch	Archive	SGG MV90 Adapter - usage archive
mv90-usage-error	Standard	SGG MV90 Adapter - usage errors

Adapter for Landis+Gyr

The table below lists suggested Object Storage buckets for the Adapter for Landis+Gyr:

Bucket Name	Storage Tier	Smart Grid Gateway Description
lg-cim-event	Standard	SGG Landis+Gyr Adapter - CIM input
lg-cim-event-arch	Archive	SGG Landis+Gyr Adapter - CIM archive
lg-cim-event-error	Standard	SGG Landis+Gyr Adapter - CIM errors
lg-event	Standard	SGG Landis+Gyr Adapter - event input
lg-event-arch	Archive	SGG Landis+Gyr Adapter - events archive
lg-event-error	Standard	SGG Landis+Gyr Adapter - event errors
lg-usage	Standard	SGG Landis+Gyr Adapter - usage input
lg-usage-arch	Archive	SGG Landis+Gyr Adapter - usage archive
lg-usage-error	Standard	SGG Landis+Gyr Adapter - usage errors

Adapter for Sensus

The table below lists suggested Object Storage buckets for the Adapter for Sensus:

Bucket Name	Storage Tier	Smart Grid Gateway Description
sensus-event	Standard	SGG Sensus Adapter - event input
sensus-event-arch	Archive	SGG Sensus Adapter - events archive
sensus-event-error	Standard	SGG Sensus Adapter - event errors
sensus-usage	Standard	SGG Sensus Adapter - usage input
sensus-usage-arch	Archive	SGG Sensus Adapter - usage archive
sensus-usage-error	Standard	SGG Sensus Adapter - usage errors

Adapter for Silver Spring Networks

The table below lists suggested Object Storage buckets for the Adapter for Silver Spring Networks:

Bucket Name	Storage Tier	Smart Grid Gateway Description
ssn-csv	Standard	SGG SSN Adapter - CSV input
ssn-csv-arch	Archive	SGG SSN Adapter - CSV archive
ssn-csv-error	Standard	SGG SSN Adapter - CSV errors
ssn-ssnxml	Standard	SGG SSN Adapter - XML input
ssn-ssnxml-arch	Archive	SGG SSN Adapter - XML archive
ssn-ssnxml-error	Standard	SGG SSN Adapter - XML errors

Copyright © 2026, Oracle and/or its affiliates.

Creating File Storage Extendable Lookup Values

Payload processing requires defining details of file storage locations. These are file system or Object Storage locations where payload files are posted for processing. File storage configuration information is captured as values for the File Storage Configuration (F1-FileStorage) extendable lookup. These records will be referenced by the batch controls that will process the payloads.

Values for this extendable lookup are defined by the following:

- **Value**: A unique code for the extendable lookup value. This value will be referenced as a batch control parameter value.
- **Description**: A description of the extendable lookup value
- **Status**: The current status of the value. Select “Active”.
- **File Storage Details**: This section defines details for the file storage location, including:
  - **File Adapter**: The type of file adapter for the location. Options include:
    - Native File Storage: Used for on-premises implementations. Additional details for this type of adapter include:
      - **File Path**: A valid file path to the file location on the file system where payload files will be found
    - Oracle Cloud Object Storage: Used with Oracle Utilities cloud services. Additional details for this type of adapter include:
      - **User**: The user Oracle Cloud ID (ODIC) for the object storage location
      - **Tenancy**: The tenancy Oracle Cloud ID (ODIC) for the object storage location
      - **Compartment**: The compartment Oracle Cloud ID (ODIC) for the object storage location
      - **Namespace**: The namespace for the object storage location
      - **Key Ring**: The Key Ring you created earlier
      - **Region**: The region of the object storage tenancy for the connection (Values for this field are defined in the F1_REGION_FLG lookup.
      - **Bucket Name Prefix**: Prefix used by Object Storage Bucket configuration

Refer to **External File Storage** in the *Oracle Utilities Application Framework Administrative User Guide* and the *Oracle Utilities Cloud Services Object Storage Setup Guide* for more information about using Oracle Object Storage with cloud service implementations.

Creating Head End Systems

The next step is to create a record for the head end system. This record defines processing methods for the head end system.

Processing Methods for Payload Processing

The following processing methods should be configured to support payload processing:

- **Device Event Mapping**: defines how head end-specific device events are mapped to standard device event names. The Device Event Seeder business object uses this

Copyright © 2026, Oracle and/or its affiliates.

processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

- **Initial Measurement Creation**: defines the business object(s) and other data used to create initial measurements. When using Legacy Measurement Processing, the IMD Seeder business object uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head end system. When using Direct Measurement Processing, specify an IMD Type(s). See Direct Measurement Processing in the *Business User Guide* for more information.
- **UOM Translation**: defines how head end-specific unit of measure (UOM) codes are mapped to standard UOM codes.

If usage payload include time of use periods or service quantity identifiers, you should also configure **SQI Translation** and **TOU Translation** processing methods.

See Head End Systems and Understanding Processing Methods in the *Administrative User Guide* for more information about creating and configuring a head end system and its processing methods.

Configuring Adapter Extendable Lookups

Smart Grid Gateway adapters use extendable lookups to define data mapping and processing rules specific to each adapter. These include:

- **Device Event Mapping**: used to map head end specific device event codes to standard event names
- **UOM Code to Standard UOM Mapping**: used to map head end specific units of measure (UOM) to standard UOM codes
- **Status Code Mapping**: used to map head end specific interval status codes to standard Condition codes.

Some adapters have additional extendable lookups used to define categories, statuses, and other data. See the following topics for information about adapter-specific extendable lookups:

- Configuring Adapter Development Kit Extendable Lookups
- Configuring Itron OpenWay Extendable Lookups
- Configuring Landis+Gyr Extendable Lookups
- Configuring MV90 Extendable Lookups
- Configuring Sensus Extendable Lookups
- Configuring Silver Spring Networks Extendable Lookups
- Configuring Itron Field Collection System Extendable Lookups

Creating SGG Payload Processing Extendable Lookup Values

You also have to create an extendable lookup value to define how usage and event payloads will be processed.

Payload processing configuration information is captured in the SGG Payload Processing Configuration (D1-SGGPayloadProcess) extendable lookup.

Values for this extendable lookup are defined by the following:

Copyright © 2026, Oracle and/or its affiliates.

- **Value**: A unique code for the extendable lookup value. This value will be referenced as a batch control parameter value.
- **Description**: A description of the lookup value
- **Status**: The current status of the value. Set this to “Active”.
- **Processing Details**: This section defines details for payload processing, including
  - **Payload Handler Class Name**: The Java class name for the processing handler to be used. See Payload Handler Classes and Parameters below for more information.
  - **Head-End System**: The head end system for which the configuration will apply. Select from the drop-down list.
  - **Populate Raw**: A checkbox that indicates if the payload data is populated as raw data.
  - **Payload Processing Result Type**: The type of payload to which the configuration applies. Options include:
    * Device Events
    * Initial Measurements
    * Initial Measurements and Device Events
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. If checked, the **IMD Seeder BO (Interval)** and **IMD Seeder BO (Scalar)** fields are hidden. See Direct Measurement Processing in the *Business User Guide* for more information. This checkbox is enabled by default.
  - **Passthrough Date/Time Format**: The date/time format for passthrough measurements. Valid values are "OUAF" and "XSD". When XSD is selected the data will be exported in UTC. Used when using Measurement Data Filtering with Direct Measurement Processing. See Direct Measurement Processing in the *Business User Guide* for more information.
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. If checked, the **IMD Seeder BO (Interval)** and **IMD Seeder BO (Scalar)** fields are hidden. See Direct Measurement Processing in the *Business User Guide* for more information..
  - **IMD Seeder BO (Interval)**: The seeder business object used to create interval initial measurements. The base package includes the “IMD Seeder” (D1-IMDSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”.
  - **IMD Seeder BO (Scalar)**: The seeder business object used to create scalar initial measurements. The base package includes the “IMD Seeder” (D1-IMDSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”.
  - **Device Event Seeder BO**: The seeder business object used to create device events. The base package includes the “Device Event Seeder” (D1-DeviceEventSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Device Events” or “Initial Measurements and Device Events”.
  - **Filter IMD**: A checkbox that indicates if initial measurements should be filtered based on unit of measure. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”. See Filtering Payloads by Unit of Measure or Device Event Mapping for more information. Note that filtering by unit of measure can be used in parallel with Measuring Component Filtering.
  - **Filter Device Events**: A checkbox that indicates if device events should be filtered. Applicable only if the Payload Processing Result Type is “Device Events” or “Initial

Copyright © 2026, Oracle and/or its affiliates.

Measurements and Device Events". See **Filtering Payloads** below for more information.

- **Dynamic Parameters**: A list of parameter names and values that support additional processing rules, based on the specified Payload Handler Class. See Payload Handler Classes and Parameters below for specific parameters supported by each adapter.
- **User Exit Interceptors**: Defines custom Groovy Library Scripts to be executed via user exit calls when parsing and transforming incoming usage and device data in custom and non-standard formats. User exit interceptors are defined as follows:

Payload Processing User Exit Type	Groovy Library Script	Groovy Library Method
On Get Parser	Parsing script	getParser
On Get Transformer	Transforming script	getTransformer
On Result List	Handler script	getHandler

See Adapter Development Kit Custom Payload Processing for more information about the On Get Parser and On Get Transformer User Exit Interceptors. See Mapping Additional Fields for information about the On Result List User Exit Interceptor.

> **Note**
>
> On Get Parser and On Get Transformer User Exit Interceptors are applicable only when the **Payload Handler Class Name** is set to `com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits`.

- Payload Handler Classes and Parameters
- Filtering Payloads by Unit of Measure or Device Event Mapping
- Upload Statistics Activities

Payload Handler Classes and Parameters

This section provides valid Payload Handler Class Names and associated dynamic parameters for supported head end systems and payload types.

Adapter Development Kit

As delivered, the Adapter Development Kit supports payload processing of usage and event data exported from an AMI head end system in the "native" initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). See The Adapter Development Kit Native Format for more information.

Payload Type: ADK CSV

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.CSVPayloadHandler

Payload Type: ADK XML

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.XMLPayloadHandler

Payload Type: ADK Seeder

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.SeederPayloadHandler

Copyright © 2026, Oracle and/or its affiliates.

Processing Custom and Non-Standard Formats

The Adapter Development Kit can be configured to support payload processing of data from AMI head end systems in custom and non-standard formats. Processing payloads of this type requires a specific **Payload Handler Class** (see below), and creation of Groovy Library Scripts to parse and transform data into the “native” format. See **Adapter Development Kit Custom Payload Processing** and **Payload Processing User Exit Interceptor Scripts** for more information.

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits

The Adapter Development Kit also supports payload processing of data in the Common Information Model (CIM) format. See Adapter Development Kit Common Information Model (CIM) Payload Processing for more information.

Adapter for Itron OpenWay

Payload Type: Itron OpenWay XML

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d8.processing.ItronXMLPayloadHandler
- **Dynamic Parameters**:

Dynamic Parameter Name	Description and Valid Values
intervalValueDecPlaces	Optional value. Specifies the number of digits after the decimal place for the quantity value. If this parameter is missing, the processing logic defaults the number of digits after the decimal place to 6.
filterRegisterSource	Optional value. Determines if source register data should be filtered. Valid values are ‘true’ and ‘false’ (default)

Payload Type: Itron OpenWay Exception

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d8.processing.ExceptionPayloadHandler

MV90 Adapter for Itron

Payload Type: MV90

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d5.processing.MV90PayloadHandler
- **Dynamic Parameters**:

Dynamic Parameter Name	Description and Valid Values
usePhysicalChannel	Optional value. Determines whether the physical channel ID is passed to the IMD seeder to create the measuring component identifier number. If this is set to false, the MV90 LOGCHAN field is used.
fieldForDvcIdN	Optional value. Specifies which field is used as the value for the device ID. Valid values include: • DC_RECID • DC_CUSTID • DC_METERID (default)

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Parameter Name	Description and Valid Values
MV90ScalarChannelSuffix	Optional value. Holds a suffix value to be added to the measuring component identifier number when a scalarIMD is created for register reads. Default value is "_S"
processMV90ScalarData	Optional value. Determines if register reads are processed. Valid values are 'true' and 'false' (default).
sendStatusAsIs	Optional value. Determines if the status codes will be passed as they came from the head end. If this set to false or not provided, then the default existing behavior when a status code is parsed and status types passed to the IMD seeder in which bits are turned on. Valid values are 'true' and 'false' (default).
MV90DateFormat	Optional value. Holds the date format used to parse provided date-time fields. The code supports the limited list of values for the format. If this parameter is missing or if the value supplied to it does not match the values from this list then the format used will be MDDYYhhmm, the default value. Valid values include: • MMDDYYhhmm (default) • MMYYDDhhmm • DDMMYYhhmm • DDYYMMhhmm • YYMMDDhhmm • YYDDMMhhmm
MV90IntervalValueDecPlaces	Optional value. Specifies the number of digits after the decimal place for the quantity value. If this parameter is missing, the processing logic defaults the number of digits after the decimal place to 6.
MV90StrtmtrAndStopmtrToDecimal	Optional value. Determines if the MV90 STRTMTR (Start Reading Meter) and STOPMTR (Stop Reading Meter) will be converted to a decimal. The processing logic will treat these fields as decimal use the setting to determine how many decimal points are required.
usePulseMultiplierAndOffset	Optional value. If set to "true", specifies the pulse values are multiplied by the pulse mutlipler. Valid values include: • true • false (default) **Note** This setting only applies to MV90 files that have a Pulse datatype.

Adapter for Itron Field Collection System (FCS)

Payload Type: Itron FCS

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d9.processing.FCSPayloadHandler
- **Dynamic Parameters:**

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Parameter Name	Description and Valid Values
uploadParametersLookup	Used to specify the FCS Upload Parameters extendable lookup value use to map specific types of FCS readings to read conditions, and specific types of FCS failures (based on read conditions in the FCS payload) to device event types.

Adapter for Landis+Gyr

Payload Type: L+G Usage

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGUsagePayloadHandler
- **Dynamic Parameters:**

Dynamic Parameter Name	Description and Valid Values
dateTimeInUTC	Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are 'true' and 'false' (default)

Payload Type: L+G Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGEventPayloadHandler
- **Dynamic Parameters:**

Dynamic Parameter Name	Description and Valid Values
dateTimeInUTC	Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are 'true' and 'false' (default)

Payload Type: L+G CIM Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGCIMEventPayloadHandler
- **Dynamic Parameters:** N/A

Adapter for Sensus

Payload Type: Sensus RNI Usage

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d6.processing.SensusUsagePayloadHandler
- **Dynamic Parameters:**

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Parameter Name	Description and Valid Values
dateTimeInUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are ‘true’ and ‘false’ (default).
useExternalTOU	Applies to Scalar usage only. Indicates whether or not an externally referenced TOU period should be used when processing measurement data. When set to true the TOU period Sensus provides will be passed to the IMD Seeder as an external TOU “preVEE/externalTOU”. When not provided or set to false then the TOU period Sensus provides will be passed to the IMD Seeder as the external MC identifier number “preVEE/mcIdN”. Valid values are ‘true’ and ‘false’ (default).
sendStatusAsIs	Determines if the status codes will be passed as they received from the head end. When set to true status codes will be sent as they are received from the head end. When not provided or set to false the status codes from the head will be processed through a bit mask to produce multiple individual statuses. Valid values are ‘true’ and ‘false’ (default).
sendIntervalDateTimes	When set to true the “preVEE/msrms/mL/dt” element of the IMD seeder for interval records will be populated with the interval date/time and the element “preVEE/enDt” will be populated with the interval date/time of the last interval. When not provided or set to false the “preVEE/msrms/mL/dt” element of the IMD seeder for interval records will not be populated and the element “preVEE/enDt” will be populated with: start date + spi * number of intervals. Valid values are ‘true’ and ‘false’ (default).
scalarExpanding	Optional Value. Defines the number of register reads to process in scalar payloads. When defined, Sensus payload processing uses this setting to ensure that all the scalar reads included in a payload will be processed. Default value is 2.

Payload Type: Sensus RNI Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d6.processing.SensusEventPayloadHandler
- **Dynamic Parameters:**

Dynamic Parameter Name	Description and Valid Values
dateTimeInUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are ‘true’ and ‘false’ (default)

Copyright © 2026, Oracle and/or its affiliates.

Adapter for Silver Spring Networks

Payload Type: Silver Spring Networks CSV

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d7.processing.SSNCSVPayloadHandler

Payload Type: Silver Spring Networks XML

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d7.processing.SSNXMLPayloadHandler

Common Parameters

The following parameters can be used by all of the supported head end system adapters.

Dynamic Parameter Name	Description and Valid Values
suppressPayloadStatistics	Optional parameter. Determines if the PayloadStatistics business object will not be created while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadStatistics business object is created at the start of the payload processing. Valid values are ‘true’ and ‘false’ (default).
suppressPayloadError	Optional parameter. Determines if the PayloadErrorNotification business object will not be created when an error occurs while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadErrorNotification business object is created for every error occurred during the payload processing. If this set to false, the error description will be written into the application log file. Valid values are ‘true’ and ‘false’ (default).
suppressPayloadSummary	Optional parameter. Determines if the PayloadSummary business object will not be created while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadSummary business object is created at the end of the payload processing. Valid values are ‘true’ and ‘false’ (default).
createFilePerCommit	Optional parameter. Determines how files are created when streaming files out to a file system or Object Storage. When set to true, streamed out files are written per commit. Valid values are ‘true’ and ‘false’ (default).
commitSizeInKB	Optional Parameter. Specifies the total amount of memory (in KB) allocated for Device Event and IMD seeders before they are committed. Seeders are kept in memory until their size exceeds the value defined for this parameter or the end of payload is reached. Default value is 100. Note: This parameter (either the default value or a specified value) is ignored when the **chunkSize** parameter is set.

Copyright © 2026, Oracle and/or its affiliates.

Dynamic Parameter Name	Description and Valid Values
chunkSize	Optional parameter. Specifies the size in bytes of a portion (chunk) of the payload that will be processed in a single batch “thread”. If parameter is set, the **commitSizeInKB** parameter is ignored. This parameter is supported by the following payload formats: • Landis+Gyr Usage • Landis+Gyr Events • MV90 Usage • Sensus RNI Usage • Sensus RNI Events • Silver Spring Networks Usage (CSV) • Adapter Development Kit Usage sample format (CSV) • Adapter Development Kit via User Exits (if an instance returned from the getParser method implements the `com.splwg.d1.domain.sgg.dg.processing.ChunkProcessor` interface).

Filtering Payloads by Unit of Measure or Device Event Mapping

When **Filter IMD** is selected, only measurements whose <externalUOM> matches one of the values defined in the head end system’s UOM Code to Standard UOM Mapping extendable lookup are passed into the system for processing.

When **Filter Device Events** is selected, only device events whose <externalEventName> matches one of the values defined in the head end system’s Device Event Mapping extendable lookup are passed into the system for processing.

Upload Statistics Activities

Upload statistics activities are activities that contain statistics related to the processing of usage and device events sent from a head end system.

There are three types of upload statistics activities:

- **Payload Statistics**: Contains statistics related to a specific payload (file) containing one or more initial measurements or device events.
- **Payload Error Notification**: Contains details concerning processing errors encountered in an individual payload (file) containing one or more initial measurements or device events.
- **Payload Summary**: Contains processing summary statistics for an individual payload (file) containing one or more initial measurements or device events. Payload Summary used to update related payload statistics upon the completion of payload processing.

Upload statistics activities are created during processing of payload files as follows:

- When processing begins for a payload, a **Payload Statistics** activity is created to record the process.
- •If an error occurs during processing, a **Payload Error Notification**activity is created.

Copyright © 2026, Oracle and/or its affiliates.

- •When payload processing is complete, a **Payload Summary**activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload, including (the start and end time of the processing, the total processing time, the number of initial measurements or device events processed, and the number of initial measurement or device event errors (if any).

Suppressing Upload Statistics Activities

Creation of upload statistics activities can be suppressed through the use of the following dynamic parameters in the SGG Payload Processing Configuration Extendable Lookup:

- suppressPayloadStatistics
- suppressPayloadSummary
- suppressPayloadError (**Note**: If this set to false, the error description will be written into the application log file.)

See **Common Parameters** in Payload Handler Classes and Parametersfor more information about these parameters.

Creating Payload Processing Batch Controls

The last step is to create payload processing monitor batch controls based on templates. You can create separate batch controls for each head end system or for specific payload types, or any combination of both. For example, you might create a separate batch control for Itron OpenWay usage payloads and device event payloads. Creating specific batch controls allows more control when setting up batch process scheduling for payload processing. For instance, you may want to run usage payload processing every hour, but device payload processing every two hours.

The base package includes the following Payload Processing template batch controls:

- Payload Processing Monitor Template (D1-PLPRM): Used with both Legacy Measurement Processing and Direct Measurement Processing
- Payload Processing Monitor Template with Export (D1-PLPSO): Used with Direct Measurement Processing and required when using Measuring Component Filtering. When used with filtering, this batch process can also send initial measurement payloads to a file system or an object storage location.
- SGG ML Payload Processing Monitor Template (D1-PLMLP): Used with Machine Learning Anomaly Scoring to retrieve measurements following machine learning processing (see Configuring Machine Learning Anomaly Scoring for more information).

To create a payload processing monitor batch control, search for the appropriate template batch control, click **Duplicate**, enter a unique code for the duplicate batch control and click **OK**.

This new batch control can be used to process payloads received from head end systems.

Payload Processing Batch Control Parameters

Payload processing batch processing is controlled by the following parameters:

Parameter Name	Description	Detailed Description	Required?
includeFiles	Include Files	The pattern for types of files to pick up during polling. The parameter supports the "glob" syntax for using wildcard characters.	True

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Detailed Description	Required?
fileLocation	File Location	The file location or Object Storage bucket to poll files from. This should be a file path or the **Value** of the File Location Extendable Lookup value for the appropriate bucket (usage or event). See Referencing URIs for more information.	True
processingConfiguration	Processing Configuration	The **Value** of the SGG Payload Processing Configuration Extendable Lookup value	True
errorLocation	Error Location	The Object Storage bucket where defected portions of payload and related rejection details are stored. This should be the **Value** of the File Location Extendable Lookup value for the appropriate “error” bucket.	True
postProcessingAction	Post Processing Action	Defines what happens to a file after it has been processed: delete (default), archive or rename. • delete: The processed file is deleted. • archive: The processed file is moved/copied to the location specified for the **archiveLocation** parameter. The file is renamed by appending the run date time value to the end of file name. The extension for archived files is .gz. If this option is selected, an archive bucket should be provided for the **archiveLocation** parameter • rename: The processed file is renamed using the extension specified for the **processedFileExtension** parameter, but the file is not moved.	False
archiveLocation	Archive Location	The Object Storage bucket where processed files are moved to. Required if the Post Processing Action is "archive". This should be the **Value** of the File Location Extendable Lookup value for the appropriate “archive” bucket.	False
processedFileExtension	Processed File Extension	The text to be appended to the file name after the file has been processed. Default is "PROCESSED".	False
maxRaiseSize	Maximum Raise Size	This parameter defines the maximum number of files that the batch program would submit for processing on each polling cycle. Default is No Maximum.	False

Copyright © 2026, Oracle and/or its affiliates.

Parameter Name	Description	Detailed Description	Required?
streamOutFileLocation	Stream Out File Location	The file location or Object Storage bucket where converted initial measurement are posted for distribution to external systems. This should be a file path or the **Value** of the File Location Extendable Lookup value for the appropriate bucket (usage). See Referencing URIs for more information. Used with D1-PLPSO.	True
streamOutFileNamePrefix	Stream Out File Name Prefix	The prefix for files created in the Stream Out File Location. The file format will be <Stream Out File Name Prefix><batch run>_<thread>_<thread count>_<timestamp>{<_file sequence>}.json.gz. If the Stream Out File Location is set then this parameter is required. Used with D1-PLPSO.	True
manifestOption	Manifest Option	This parameter specifies that a manifest file be created. See Manifest File for more information. Used with D1-PLPSO.	False
encryptExternalKey	Encryption External Key	The key ring used for the external key provided by the recipient of the file. If populated, the batch output file will be encrypted using a runtime generated key. The generated key will be encrypted using the active public key stored on the key ring. The recipient will use their private key to decrypt the generated key so the file may be decrypted. Only external public key rings may be used. See File Encryption and Signing Keys for more information. Used with D1-PLPSO.	False
encryptSignatureKey	Encryption Signature Key	The key ring for the internal key used to sign the file. If populated, the batch output file will be signed using the active private key stored on the key ring. The assumption is that the corresponding public key has been shared with the recipient. Only internal file encrypting / signing key rings may be used. See File Encryption and Signing Keys for more information. Used with D1-PLPSO.	False
DIST-THD-POOL	Thread Pool	Thread pool name to use if the DEFAULT thread pool is not desired.	False

> **Note**
>
> Refer to Creating Object Storage Locations for lists of suggested Object Storage buckets for each adapter.

Copyright © 2026, Oracle and/or its affiliates.

Mapping Additional Fields

Measurements and device event files received from head end systems can include additional fields containing data to be imported into Oracle Utilities Smart Grid Gateway. These additional fields must be mapped to elements within the measurements and device events processed by Smart Grid Gateway.

This mapping can be performed through use of a custom Groovy script, via the “On Result List” User Exit Interceptor specified in the SGG Payload Processing Configuration Extendable lookup.

Payload Processing Overview

This section provides an overview of the payload processing logic when mapping data in incoming measurements or device events.

Oracle Utilities Meter Data Framework Steps

1. The file upload process program starts reading the file.
2. The upload process creates an instance of the Payload Handler class specified on the SGG Payload Processing Configuration extendable lookup. This class handles the interaction with the User Exit Interceptor script.
3. The upload process invokes the “On Result List” User Exit Interceptor. This executes a “handler” Groovy Library Script that returns an instance of a class that implements the com.splwg.d1.domain.sgg.dg.processing.PayloadProcessingCMHandler interface. This "CM payload handler" is initialized with a stream opened for the incoming file and a business object instance of SGG Payload Processing Configuration extendable lookup.
4. The upload process parses data from the incoming document into the “Plain XML” format and transforms the Plain XML data into a set of Initial Measurements (IMDs) or/and Device Events in the “native” XML format and returns each as child nodes added to the Result List parent node.
5. The upload process invokes the CM payload handler, passing the Result List parent node.
6. The CM payload handler transforms data the Result List parent node based on the logic in the Handler script. See **CM Payload Handler Steps**, below.
7. The upload process creates business object instances for all child nodes of the Result List.
8. The process steps (4-7) are repeated until the process returns NULL on step 4 (when there are no further records to process).

CM Payload Handler Steps

1. The CM payload handler reads the given measurement or device event inside the Result List node.
2. The CM payload handler transforms specific data elements inside the measurement or device event (such as replacing a <externalUOM> element with the standard <UOM> element) based on the logic in the handler script, and updates the Result List accordingly.

CM Handler Payload Processing User Exit Interceptor Scripts

Mapping data is performed via a Groovy Library Script that is invoked via a user exit during payload processing. The script used in this process is a referred to as a “handler” script.

Handler Script

Copyright © 2026, Oracle and/or its affiliates.

The “handler” script is responsible for transforming and mapping data elements within measurements and device events.

This script should include the gethandler method that returns an instance of a class that implements the ccom.splwg.d1.domain.sgg.processing.PayloadProcessingCMHandler interface.

To view details of the PayloadProcessingCMHandler interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select PayloadProcessingCMHandlerfrom the list of interfaces.

The “handler” script should be defined for the “On Result List” Payload Processing User Exit Type in the User Exit Interceptors section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

Configuration Steps

Configuration of Smart Grid Gateway adapters to support mapping of non-standard data involves the following steps:

1. Create a “handler script” that transforms and maps specific data elements inside the measurement or device event (such as replacing a <externalUOM> element with the standard <UOM> element). See CM Handler Payload Processing User Exit Interceptor Scripts for more information.
2. Create an SGG Payload Processing Configuration extendable lookup value. The **Payload Handler Class Name** should reference one of the usage or event “PayloadHandler” classes listed in the Payload Handler Classes and Parameters section. See Creating SGG Payload Processing Extendable Lookup Values for more information about creating these extendable lookup values..
3. Specify the script for the “On Result List” **Payload Processing User Exit Type** in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.
4. Create other objects and data used with payload processing, including:
    - Key Rings and Pairs (See Creating Key Rings and Pairs)
    - Object Storage Locations (See Creating Object Storage Locations)
    - File Storage Extendable Lookup values (See Creating File Storage Extendable Lookup Values)
    - Head End System (See Creating Head End Systems)
    - Payload Processing Batch Controls (See Creating Payload Processing Batch Controls)

- Sample Implementation

Sample Implementation

The following sample illustrates how additional fields can be mapped into the XML format sent to Smart Grid Gateway. In this case, the unit of measure for incoming measurement data is defined by an <externalUOM>, which must be mapped to the standard <UOM> element.

Copyright © 2026, Oracle and/or its affiliates.

Sample Script

The below Groovy Script will loop through a list of “D1-IMDSeeder” nodes and check the “externalUom” for a value (GD_KWH in this example), and if equal then the <UOM> element will be created with the <externalUOM> values and the <externalUOM> element will be removed.

Script Type: Groovy Library Script

Step 10: Groovy Imports

```
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import com.splwg.d1.domain.sgg.processing.PayloadProcessingCMHandler;
import com.splwg.shared.logging.Logger;
import com.splwg.shared.logging.LoggerFactory;
import com.splwg.base.support.scripting.XMLUtils;
```

Step 20: Groovy Library Interface

```
PayloadProcessingCMHandler getHandler()
```

Step 30: Groovy Members

```
public PayloadProcessingCMHandler getHandler() {
  return new CMhandler();
}

public class CMhandler implements PayloadProcessingCMHandler{

  org.w3c.dom.Document onList(org.w3c.dom.Document resultList,
org.w3c.dom.Document plainXML) {

  NodeList imdList=resultList.getDocumentElement().getElementsByTagName("D1-
IMDSeeder");
  for(int i=0;i<imdList.getLength();i++){
    Element imdEl = (Element) imdList.item(i);

    //check if msrs element exists
    NodeList msrsList = imdEl.getElementsByTagName("msrs");
    if (msrsList.length > 0) {
      //get preVEE
      NodeList preVEENodeList = imdEl.getElementsByTagName("preVEE");
      Element preVEENode = (Element) preVEENodeList.item(0);
      //get externaluom
      NodeList uomList = preVEENode.getElementsByTagName("externalUOM");

      Element uomElm = uomList.length>0 ? uomList.item(0) : null;
      String uom = uomElm!=null ? uomElm.getTextContent() : "";
      //get mcIdN
```


Copyright © 2026, Oracle and/or its affiliates.

```
      NodeList mcIdNList = preVEENode.getElementsByTagName("dvcIdN");
      if ( uom.compareTo("GD_KWH")==0 && mcIdNList.length>0 ) {
        Element externalUOMElm = resultList.createElement("uom");
        externalUOMElm.setTextContent(uom);
        //replace uom with externalUOM
        preVEENode.replaceChild(externalUOMElm, uomElm);
      }
    }
  }
  return resultList;
  }
}
```

Sample File

Below is a sample Landis+Gyr usage file. Note that this file has a"Units" value of "GD_KWH".

```
Record Type~Record Version~Time Stamp~Premise ID~ESIID~Provisioned~Meter
ID~Purpose~Commodity~Units~Calculation
Constant~Interval~Count~FirstIntervalDateTime~Data
MEPMD01~20080519~11092010012000AM~~~~GD_LL_SN100~OK~E~GD_KWH~~000100~24~100620
10010000AM~~0.2316~~0.1416~~0.2904~~0.2628~~0.240~~0.2796~~0.2232~~0.1692~~0.0
552~~0.24096~~0.276~~0.1572~~0.0612~~0.3168~~0.132~~0.1092~~0.2532~~0.2844~~0.
198~~0.0756~~0.3084~~0.15240~~0.1788~~0.1716~~
```

Plain XML Output

Below is the Plain XML output for the above Landis+Gyr usage file. Note that this file has a <Units> element with a value of "GD_KWH".

```
<MeterReads
xmlns="http://xmlns.oracle.com/LandisGyrUsage">
    <MeterRead>
        <Origin>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</Origin
>
        <ServProvExtRefId>L+G</ServProvExtRefId>
        <RecordType>MEPMD01</RecordType>
        <RecordVersion>20080519</RecordVersion>
        <TimeStamp>11092010012000AM</TimeStamp>
        <MeterID>GD_LL_SN100</MeterID>
        <Purpose>OK</Purpose>
        <Comodity>E</Comodity>
        <Units>GD_KWH</Units>
        <Interval>000100</Interval>
        <Count>24</Count>
        <FirstIntervalDateTime>10062010010000AM</FirstIntervalDateTime>
        <Data>
            <Row v="0.2316"/>
            <Row v="0.1416"/>
            <Row v="0.2904"/>
            <Row v="0.2628"/>
            <Row v="0.240"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <Row v="0.2796"/>
            <Row v="0.2232"/>
            <Row v="0.1692"/>
            <Row v="0.0552"/>
            <Row v="0.24096"/>
            <Row v="0.276"/>
            <Row v="0.1572"/>
            <Row v="0.0612"/>
            <Row v="0.3168"/>
            <Row v="0.132"/>
            <Row v="0.1092"/>
            <Row v="0.2532"/>
            <Row v="0.2844"/>
            <Row v="0.198"/>
            <Row v="0.0756"/>
            <Row v="0.3084"/>
            <Row v="0.15240"/>
            <Row v="0.1788"/>
            <Row v="0.1716"/>
        </Data>

<RawData>MEPMD01~20080519~11092010012000AM~~~~GD_LL_SN100~OK~E~GD_KWH~~000100~
24~10062010010000AM~~0.2316~~0.1416~~0.2904~~0.2628~~0.240~~0.2796~~0.2232~~0.
1692~~0.0552~~0.24096~~0.276~~0.1572~~0.0612~~0.3168~~0.132~~0.1092~~0.2532~~0
.2844~~0.198~~0.0756~~0.3084~~0.15240~~0.1788~~0.1716~~</RawData>
    </MeterRead>
</MeterReads>
```

Result List Input

Below is the Result List input sent to the handler script. Note that this file has an <externalUOM> element with a value of "GD_KWH".

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultList>
    <D1-IMDSeeder>
        <fromDateTime>2010-10-06-00.00.00</fromDateTime>
        <preVEE>
            <dvcIdN>GD_LL_SN100</dvcIdN>
            <externalId>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</
externalId>
            <externalUOM>GD_KWH</externalUOM>
            <stDt>2010-10-06-00.00.00</stDt>
            <spi>3600</spi>
            <imdType>D1IL</imdType>
            <msrs>
                <mL>
                    <s>1</s><q>0.2316</q>
                </mL>
                <mL>
                    <s>2</s><q>0.1416</q>
                </mL>
                <mL>
                    <s>3</s><q>0.2904</q>
                </mL>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<mL>
    <s>4</s><q>0.2628</q>
</mL>
<mL>
    <s>5</s><q>0.240</q>
</mL>
<mL>
    <s>6</s><q>0.2796</q>
</mL>
<mL>
    <s>7</s><q>0.2232</q>
</mL>
<mL>
    <s>8</s><q>0.1692</q>
</mL>
<mL>
    <s>9</s><q>0.0552</q>
</mL>
<mL>
    <s>10</s><q>0.24096</q>
</mL>
<mL>
    <s>11</s><q>0.276</q>
</mL>
<mL>
    <s>12</s><q>0.1572</q>
</mL>
<mL>
    <s>13</s><q>0.0612</q>
</mL>
<mL>
    <s>14</s><q>0.3168</q>
</mL>
<mL>
    <s>15</s><q>0.132</q>
</mL>
<mL>
    <s>16</s><q>0.1092</q>
</mL>
<mL>
    <s>17</s><q>0.2532</q>
</mL>
<mL>
    <s>18</s><q>0.2844</q>
</mL>
<mL>
    <s>19</s><q>0.198</q>
</mL>
<mL>
    <s>20</s><q>0.0756</q>
</mL>
<mL>
    <s>21</s><q>0.3084</q>
</mL>
<mL>
    <s>22</s><q>0.15240</q>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                </mL>
                <mL>
                    <s>23</s><q>0.1788</q>
                </mL>
                <mL>
                    <s>24</s><q>0.1716</q>
                </mL>
            </msrs>
        </preVEE>
        <serviceProviderExternalId>L+G</serviceProviderExternalId>
    </D1-IMDSeeder>
</ResultList>
```

Result List Output

Below is the Result List output sent returned by the handler script. Note that <externalUOM> element had been replaced with a <UOM> element with a value of "GD_KWH".

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultList>
    <D1-IMDSeeder>
        <fromDateTime>2010-10-06-00.00.00</fromDateTime>
        <preVEE>
            <dvcIdN>GD_LL_SN100</dvcIdN>
            <externalId>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</
externalId>
            <uom>GD_KWH</uom>
            <stDt>2010-10-06-00.00.00</stDt>
            <spi>3600</spi>
            <imdType>D1IL</imdType>
            <msrs>
                <mL>
                    <s>1</s><q>0.2316</q>
                </mL>
                <mL>
                    <s>2</s><q>0.1416</q>
                </mL>
                <mL>
                    <s>3</s><q>0.2904</q>
                </mL>
                <mL>
                    <s>4</s><q>0.2628</q>
                </mL>
                <mL>
                    <s>5</s><q>0.240</q>
                </mL>
                <mL>
                    <s>6</s><q>0.2796</q>
                </mL>
                <mL>
                    <s>7</s><q>0.2232</q>
                </mL>
                <mL>
                    <s>8</s><q>0.1692</q>
                </mL>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <mL>
                        <s>9</s><q>0.0552</q>
                    </mL>
                    <mL>
                        <s>10</s><q>0.24096</q>
                    </mL>
                    <mL>
                        <s>11</s><q>0.276</q>
                    </mL>
                    <mL>
                        <s>12</s><q>0.1572</q>
                    </mL>
                    <mL>
                        <s>13</s><q>0.0612</q>
                    </mL>
                    <mL>
                        <s>14</s><q>0.3168</q>
                    </mL>
                    <mL>
                        <s>15</s><q>0.132</q>
                    </mL>
                    <mL>
                        <s>16</s><q>0.1092</q>
                    </mL>
                    <mL>
                        <s>17</s><q>0.2532</q>
                    </mL>
                    <mL>
                        <s>18</s><q>0.2844</q>
                    </mL>
                    <mL>
                        <s>19</s><q>0.198</q>
                    </mL>
                    <mL>
                        <s>20</s><q>0.0756</q>
                    </mL>
                    <mL>
                        <s>21</s><q>0.3084</q>
                    </mL>
                    <mL>
                        <s>22</s><q>0.15240</q>
                    </mL>
                    <mL>
                        <s>23</s><q>0.1788</q>
                    </mL>
                    <mL>
                        <s>24</s><q>0.1716</q>
                    </mL>
                </msrs>
            </preVEE>
            <serviceProviderExternalId>L+G</serviceProviderExternalId>
        </D1-IMDSeeder>
    </ResultList>
```


Copyright © 2026, Oracle and/or its affiliates.

Itron OpenWay Payload Processing

By default, payload processing for the Itron OpenWay adapter works differently than it does for other adapters, but it can be configured to work like other adapters in which payloads are saved to a staging area in Oracle Object Storage and then processed via batch processing. For large volumes, saving to the staging area provides better performance because it allows the web service to simply drop off the payload, and the collection of saved payloads can then be processed in multiple batch processing threads.

Configuration of Itron payload processing requires the same set up and configuration as other adapters. Some specifics regarding configuration for Itron processing include the following:

- **Object Storage Locations**: Itron OpenWay payload processing uses object storage locations (including Key Rings, Key Pairs and File Storage Configuration extendable lookup values) to capture usage and event payloads or exceptions received from the head end system.
- **Head End Systems and Extendable Lookups**: Itron payload processing uses head end system and extendable lookup configuration in the same manner as other adapters.
- **SGG Payload Processing Configuration Extendable Lookup Values**: These values define payload processing details for the Itron head end system (which is referenced on the payload processing batch control).
- **Payload Processing Batch Controls**: A batch control used for payload process (based on a template batch control provided with the base package)
- **Master Configuration**: Configuration settings for Itron OpenWay payload processing, including specifying the use of a payload staging area in Object Storage. See Itron OpenWay Master Configuration for more information about configuring the system to process Itron OpenWay payloads and exceptions.

Deferring Batch Processing

We recommend configuring a monitor process on the Pending status of the Initial Load initial measurement business objects so that the VEE and measurement creation processes can be deferred to batch processing.

Deferring processing has the following additional benefits:

1. It can ensure chronological processing for scalar reads which provides more efficient processing. Processing scalar reads out of order can result in additional work as reconciliation initial measurements will be created and processed to true up the consumption calculations.
2. It can ensure measuring components are processed in order based on their relationships. For example, if a scalar channel is configured as the check channel for an interval channel, the scalar channel will be processed first so that its data is available for sum checks and any interval gap filling required.

Configuration

Configuring deferred processing involves adding the “IMD Monitor – Physical Devices” Batch Control as the Monitor Process on the Pending status to the following business objects:

- D1-InitialLoadIMDInterval
- D1-InitialLoadIMDScalar

When initial measurement processing is deferred it will be necessary to schedule the “IMD Monitor – Physical Devices” Batch Control to run more frequently. It should be scheduled such

Copyright © 2026, Oracle and/or its affiliates.

that it runs with a higher thread count immediately after the Itron Scheduled Reads cycle has completed and with fewer threads throughout the rest of day to pick up any other initial measurements that might be received (such as those received via On Demand Read commands).

Default Processing — Not Recommended

By default, payload processing for the Itron OpenWay adapter works differently than it does for other adapters. Instead of using batch processing, Itron payloads are processed as they are received. Payloads and exceptions sent from the Itron OpenWay head end system are received via inbound web services that invoke business components that process the payloads and exceptions as they are received rather than posting them to object storage locations. See Itron OpenWay Inbound Web Services for more information about the inbound web services used to receive payloads and exceptions from the Itron OpenWay head end system. This approach is not recommended.

Adapter Development Kit Custom Payload Processing

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process custom payload formats, including payloads from non-supported head end systems.

- Introduction
- Custom Payload Processing Overview
- Payload Processing User Exit Interceptor Scripts
- Configuration Steps
- Sample Implementation

Introduction

As delivered, the Adapter Development Kit supports payload processing of usage and event data exported from an AMI head end system in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects).

The Adapter Development Kit can also be configured to support payload processing of data from AMI head end systems in custom and non-standard formats. This involves creation of Groovy Library Scripts to:

- Parse the payload into a “plain XML” format, and
- Transform the plain XML format into the “native” ” initial measurement and device event data formats

These scripts are executed as User Exit Interceptors configured on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Configuration Extendable Lookup Values.

Custom Payload Processing Overview

This section provides an overview of the payload processing logic when converting incoming data into the native format.

Copyright © 2026, Oracle and/or its affiliates.

Oracle Utilities Meter Data Framework Steps

1. The file upload process program starts reading the file.
2. The upload process creates an instance of the Payload Handler class. This class handles the interaction with the User Exit Interceptor scripts.
3. The upload process invokes the “On Get Parser” User Exit Interceptor. This executes a “parser” Groovy Library Script that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface. This “payload parser” is initialized with a stream opened for the incoming file and a business object instance of SGG Payload Processing Configuration extendable lookup.
4. The upload process invokes the “On Get Transformer” User Exit Interceptor. This executes a “transformer” Groovy Library Script that returns an instance of a class that implements `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface. This “payload transformer” is initialized with a string representing the origin of the payload and business object instance of SGG Payload Processing Configuration extendable lookup.
5. The upload process invokes the payload parser to parse data from the incoming document into the “Plain XML” format. See Payload Parser Steps, below.
6. The upload process invokes the payload transformer, passing the Plain XML data and a Result List parent node.
7. The payload transformer transforms the Plain XML data into a set of Initial Measurements (IMDs) or/and Device Events in the “native” XML format and returns each as child nodes added to the Result List parent node. See Payload Transformer Steps, below.
8. The upload process creates business object instances for all child nodes of the Result List.
9. The process steps (5-9) are repeated until the payload parser returns NULL on step 5 (when there are no further records to process).

Payload Parser Steps

1. The payload parser reads an input stream until it hits the logical end of portion of data (each usage or event record), which could be transformed into Plain XML.
2. The payload parser parses the data and converts into the Plain XML format.
3. The process is performed for each call from the upload process.

Payload Transformer Steps

1. The payload transformer reads the given Plain XML.
2. The payload transformer generates a set of XML nodes representing Initial Measurements (IMDs) or Device Events. Names for nodes are taken from the provided SGG Payload Processing Configuration.
3. The payload transformer adds the generated nodes to the given Result List node.

Payload Processing User Exit Interceptor Scripts

Conversion of data from custom and non-standard formats into the “native” format is performed via a pair of Groovy Library Scripts that are invoked via user exits during payload processing (see Custom Payload Processing Overview). The two scripts used in this process include a “parser” script, and a “transformer” script.

Copyright © 2026, Oracle and/or its affiliates.

Parser Script

The “parser” script is responsible for parsing the incoming data and converting it into the Plain XML format. This script should include the `getParser` method that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface.

To view details of the *PayloadParser* interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select *PayloadParser* from the list of interfaces.
The “parser” script should be defined for the “On Get Parser” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

The base package includes two sample parser scripts that can be used as examples when developing custom parser scripts. See Sample Implementation.

Transformer Script

The “transformer” script is responsible for converting data the Plain XML format into the “native” format. This script should include the `getTransformer` method that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface.

To view details of the *PayloadTransformer* interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select *PayloadTransformer* from the list of interfaces.

The “transformer” script should be defined for the “On Get Transformer” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

The base package includes two sample transformer scripts that can be used as examples when developing custom transformer scripts. See Sample Implementation.

Configuration Steps

Configuration of the Adapter Development Kit to support conversion of non-standard formats into the “native” format involves the following steps:

1. Create a “parser script” that converts data from the incoming format into Plain XML format. See Payload Processing User Exit Interceptor Scripts for more information.
2. Create a “transformer script” that converts data from the Plain XML format to the “native” format.
   See Payload Processing User Exit Interceptor Scripts for more information.
3. Create an SGG Payload Processing Configuration extendable lookup value with a **Payload Handler Class Name** of “com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits”.
   See Creating SGG Payload Processing Extendable Lookup Values for more information.
4. Specify the parser script for the “On Get Parser” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup.

Copyright © 2026, Oracle and/or its affiliates.

See Creating SGG Payload Processing Extendable Lookup Values for more information.

5. Specify the transformer script for the “On Get Transformer” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup.
   See Creating SGG Payload Processing Extendable Lookup Values for more information.
6. Create other objects and data used with payload processing, including:
   - Key Rings and Pairs (See Creating Key Rings and Pairs)
   - Object Storage Locations (See Creating Object Storage Locations)
   - File Storage Extendable Lookup values (See Creating File Storage Extendable Lookup Values)
   - Head End System (See Creating Head End Systems)
   - Payload Processing Batch Controls (See Creating Payload Processing Batch Controls)

Sample Implementation

The Smart Grid Gateway Adapter Development Kit includes sample parser and transformer scripts that demonstrate how custom payload processing works. These sample scripts include samples for processing data in comma separated values (CSV) format and in XML format.

- Comma Separated Values Format
- XML Format

Comma Separated Values Format

This page describes the comma separated values format sample implementation.

Sample File Format

In this sample, data is stored as a comma separated values file.

Note: Values in this example cannot contain a literal comma character. This restriction is added for the simplicity of file parser implementation. The parsing logic can be re-implemented to avoid this restriction. It will not affect the rest of functionality.

The sample CSV file contains the following record types:

Interval Usage: Used for interval usage measurements

Device Events: Used for device events

Trail record: Used to provide the read date/time and number of interval usage and device event records in the file

Interval Usage Format

#	Field Name	Type	Definition
1	Record Type ID	Constant value “U”	Record type identifier
2	Start Date/Time	Time in the Unix Time format.	Meter read’s start date/time
3	End Date/Time	Time in the Unix Time format.	Meter read’s end date/time
4	Device Id	Arbitrary Text	Device identifier

Copyright © 2026, Oracle and/or its affiliates.

#	Field Name	Type	Definition
5	Interval Duration	Numeric Integer	The time interval between readings in seconds (seconds per interval).
6	UOM	Arbitrary Text	Describes the units of the data values
7	Data Entry(s)	Numeric Floating-Point and optional Arbitrary Text separated by ":"	Each record can contain unlimited number of data entry fields. Each data entry is a set of two fields – a reading value and an optional reading status flag. Fields are separated by ":" character. The separator is not present when the reading status flag is not provided.

Device Event Format

#	Field Name	Type	Definition
1	Record Type ID	Constant value "E"	Record type identifier
2	Event Date/Time	Time in the Unix Time format.	The date and time at which the event occurred
3	Device Id	Arbitrary Text	Device identifier
4	Event Name	Arbitrary Text	The primary identification name of the event

Trail Record Format

#	Field Name	Type	Definition
1	Record Type ID	Constant value "T"	Record type identifier
2	Creation Date/Time	Time in the Unix Time format.	Meter read's start date/time
3	Total Records	Numeric Integer	Displays number of records in file

Sample File

```
U,1,86400,DEVICE_DG_0,900,KWH,1:S,2.5,-3.99:R,4:B
E,86400,DEVICE_DG_1,Power Outage
U,2,86400,DEVICE_DG_2,900,KWH,1:S,2,3:R,4:B,5,6,7:A
E,86402,DEVICE_DG_3,Tamper attempt suspected
U,3,86400,DEVICE_DG_4,900,KWH,1:S
T,86401,5
```

Sample Scripts

The sample implementation includes the following sample scripts designed for use with the above CSV format:

DG_SmplPrCSV: This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface. The class contains the following logic:

Copyright © 2026, Oracle and/or its affiliates.

1. Parses an incoming payload in the CSV Format.
2. Transforms incoming data to Usage or Event related structures based on incoming data type
3. Returns one by one that structures in the Plain XML format

DG_SmPrCSVCh: This script is provided as a sample to demonstrate how to parse the CSV format in chunks (created via the `chunkSize` parameter) by utilizing a helper. This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.ChunkProcessor` and `com.splwg.d1.domain.sgg.dg.processing.GenericPayload` interfaces.

The ChunkProcessor interface distinguishes a chunking-aware processor from the rest of implementations. A custom parser implemented in groovy can implement this interface. The `GenericPayload` interface defines methods needed to get access to a payload via specific helpers. Via this interface, a custom parser gets access to an instance of `LineReaderChunked`.

If the `chunkSize` dynamic parameter is not set in an SGG Payload Processing configuration the entire payload will be processed. See Common Parameters for more information about the `chunkSize` parameter.

The class contains the following logic:

1. Reads a line from incoming payload by utilizing the `LineReaderChunked.readLineWithinChunk()` method
2. Parses a line in the incoming payload (in the CSV Format) that was read in the previous step.
3. Transforms incoming data to Usage or Event related structures based on incoming data type
4. Returns one by one that structures in the Plain XML format

DG_SmplTrCSV: This sample script includes one method in the Library Interface – `getTransformer()`. The method returns an instance of class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface. The class contains the following logic:

1. Transforms incoming message in the Plain XML format into the “native” format (the IMD Seeder or Device Event Seeder XML nodes).
2. Adds new XML nodes as child nodes to the given Result List.

Use the **Script** portal to view these scripts in more detail.

Parsing

The parser script performs the following data mapping.
Interval Usage to Plain XML Mapping

The following table shows the mapping between fields in incoming interval data and child elements of Payload/Usage element in the Plain XML format:

Interval Usage Field	Plain XML Element
Record Type ID	RecordType
Start Date/Time	StartDateTime

Copyright © 2026, Oracle and/or its affiliates.

Interval Usage Field	Plain XML Element
End Date/Time	EndDateTime
Device Id	DeviceId
Interval Duration	IntervalDuration
UOM	UOM
Data Entry(s)	Intervals/ Interval/ Value [Status]
Record content from incoming file.	RawData

Device Event to Plain XML Mapping

The following table shows the mapping between fields in incoming interval data and child elements of Payload/Event element in the Plain XML format:

Device Event Field	Plain XML Element
Record Type ID	RecordType
Event Date/Time	DateTime
Device Id	DeviceId
Event Name	Name
Record content from incoming file.	RawData

XML Schema of Plain XML Format

The Plain XML consists of the Interval Usage and Device Event related elements at the same time.

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/GenericAdapter"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ga="http://xmlns.oracle.com/GenericAdapter">
  <xs:element name="Payload" type="ga:PayloadType"/>
  <xs:complexType name="PayloadType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Usage" type="ga:UsageType"/>
        <xs:element name="Event" type="ga:EventType"/>
      </xs:choice>
      <xs:element name="RawData"  type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UsageType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeUsageType"/>
      <xs:element name="StartDateTime" type="xs:integer"/>
      <xs:element name="EndDateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="IntervalDuration" type="xs:integer"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <xs:element name="UOM" type="ga:NonEmptyString"/>
      <xs:element name="Intervals" type="ga:IntervalList"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeUsageType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="U"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="IntervalList">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="Interval" type="ga:IntervalType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IntervalType">
    <xs:sequence>
      <xs:element name="Value" type="xs:float"/>
      <xs:element name="Status" type="ga:NonEmptyString" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeEventType"/>
      <xs:element name="DateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="Name" type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeEventType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="E"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="NonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

> **Note**
>
> **Note**: A part of incoming file that been used to generate current Plain XML structure will be placed into the Payload/RawData element.

Transformation

The transformer script performs the following mapping.

Plain XML to Seeder XML Mapping

The payload transformer creates the IMD or/and Device Event Seeder XML elements as following:

Copyright © 2026, Oracle and/or its affiliates.

Usage Mapping: If name of Payload/* node is equal to 'Usage' (interval usage), the script creates an XML element with a name defined in the SGG Payload Processing Configuration field “IMD Seeder BO (Interval)” and the structure according to the following mapping table:

Plain XML Element	IMD Seeder Element	Notes
RecordType	N/A	
DeviceId	dvcIdN	
UOM	externalUOM	
IntervalDuration	spi	
StartDateTime	stDt	Convert to OUAF "dateTime" type
EndDateTime	enDt	Convert to OUAF "dateTime" type
Intervals/		
Interval/		
Value [Status]	msrs	Value
msrs/mL/q		
Status ->		
msrs/mL/sts/stsL/st		
N/A	imdType	Const value 'D1IL'
	externalId	Value stored in setOrigin method.
	serviceProviderExternalId	Value stored in setConfiguration method.
RawData	rawData	If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked.

Device Event Mapping: If name of Payload/* node is equal to 'Event' (device event data) create an XML element with a name defined in the SGG Payload Processing Configuration field “Device Event Seeder BO” and the structure according to the following mapping table:

Plain XML Element	Device Event Seeder Element	Note
RecordType	N/A	
DateTime	eventDateTime	Convert to OUAF "dateTime" type
DeviceId	externalUOM	
Name	externalEventName	
	externalSourceIdentifier	Value stored in setOrigin method.
	externalSenderId	Value stored in setConfiguration method.
RawData	rawEventInformation	If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked.

XML Format

This page describes the XML format sample implementation.

Sample File Format

The file format is described in the Online IMD Upload hint on the **Online IMD and Event Upload** screen. (Select **Menu**, select **Communication**, select **Load IMDs/Events (XML)**).

Copyright © 2026, Oracle and/or its affiliates.

Sample File

```
<deviceList>
  <device>
    <headEnd>L&G</headEnd>
    <headEndExternalId>L+G</headEndExternalId>
    <deviceId></deviceId>
    <deviceIdentifierNumber>DEV-OUSGG-OSB-DEMO-001</deviceIdentifierNumber>
    <initialMeasurementDataList>
      <initialMeasurementData>
        <preVEE>
          <mcIdN></mcIdN>
          <uom>KWH</uom>
          <stDt>2010-05-19-00.00.00</stDt>
          <enDt>2010-05-19-00.30.00</enDt>
          <spi>900</spi>
          <msrs>
            <mL>
              <s>1</s>
              <q>0.2316</q>
            </mL>
            <mL>
              <s>2</s>
              <q>0.1416</q>
            </mL>
          </msrs>
        </preVEE>
      </initialMeasurementData>
      <initialMeasurementData>
        <initialMeasurementDataId>testValue</initialMeasurementDataId>
          <preVEE>
            <mcIdN></mcIdN>
            <uom>KWH2</uom>
            <stDt>2010-05-19-00.30.00</stDt>
            <enDt>2010-05-19-01.00.00</enDt>
            <spi>900</spi>
            <msrs>
              <mL>
                <s>1</s>
                <q>1.2316</q>
              </mL>
              <mL>
                <s>2</s>
                <q>2.1416</q>
              </mL>
            </msrs>
          </preVEE>
      </initialMeasurementData>
    </initialMeasurementDataList>
  </device>
</deviceList>
```

Sample Scripts

The sample implementation includes the following sample scripts designed for use with the above XML format:

Copyright © 2026, Oracle and/or its affiliates.

DG_SmplPrXML: This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface. The class contains the following logic:

1. Parses an incoming payload in XML format
2. Fetches information about current device
3. Breaks an incoming data into separate initial measurement data structures
4. Extends these structures with device information
5. Returns one by one these structures in the Plain XML format

DG_SmplTrXML: This sample script includes one method in the Library Interface – `getTransfprmer()`. The method returns an instance of class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface. The class contains the following logic:

1. Transforms incoming message in the Plain XML format into the IMD Seeder XML nodes
2. Adds new XML nodes as child nodes to the given Result List

Use the **Script** portal to view these scripts in more detail.

Parsing

The XML parser script maps data from the above XML format into the Plain XML format.

XML Schema of Plain XML Format

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/GenericAdapter"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ga="http://xmlns.oracle.com/GenericAdapter">
  <xs:element name="Payload" type="ga:PayloadType"/>
  <xs:complexType name="PayloadType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Usage" type="ga:UsageType"/>
        <xs:element name="Event" type="ga:EventType"/>
      </xs:choice>
      <xs:element name="RawData"  type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UsageType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeUsageType"/>
      <xs:element name="StartDateTime" type="xs:integer"/>
      <xs:element name="EndDateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="IntervalDuration" type="xs:integer"/>
      <xs:element name="UOM" type="ga:NonEmptyString"/>
      <xs:element name="Intervals" type="ga:IntervalList"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeUsageType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="U"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="IntervalList">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="Interval" type="ga:IntervalType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IntervalType">
    <xs:sequence>
      <xs:element name="Value" type="xs:float"/>
      <xs:element name="Status" type="ga:NonEmptyString" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeEventType"/>
      <xs:element name="DateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="Name" type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeEventType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="E"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="NonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Transformation

The transformer script performs the following mapping.

Plain XML to Seeder XML Mapping

The payload transformer script does the following:

1. Creates an XML element with a name defined in the SGG Payload Processing Configuration field “IMD Seeder BO (Interval)” and copies all elements from the Plain XML as child nodes of new element.
2. Inserts the “externalId” element populated with value stored in `setOrigin` method.
3. Inserts the “serviceProviderExternalId” element populated with value stored in `setConfiguration` method.
4. If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked, inserts the “rawData” element populated with the textual XML representation of Plain XML element.

Adapter Development Kit Common Information Model (CIM) Payload Processing

Copyright © 2026, Oracle and/or its affiliates.

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process payload formats in the Common Information Model (CIM) format.

- CIM Payload Processing Introduction and Overview
- Creating CIM Payload Processing Extendable Lookup Values
- CIM Event Processing Inbound Web Service
- CIM Payload Processing Data Mapping and Extendable Lookups
- CIM Payload Processing Sample Payloads

CIM Payload Processing Introduction and Overview

The Adapter Development Kit supports payload processing of data in the Common Information Model (CIM) format. The CIM formats and standards supported by the Adapter Development Kit are based on the *International Standard* (IEC 61968-9, Edition 2.0), available here. Reference to specific data elements from the CIM usage and event payload formats used in this section are based on this document. Refer to this document for detailed information about the CIM standard and payload formats.

Usage payloads are processed in the same manner as other formats and other adapters. Payload files are sent from a head end system to a file system (for on-premises implementations) or an Oracle Cloud Object Storage location (for cloud services), where they are processed via batch processes. See Payload Processing for more information about this process and the related configuration. Events can also be processed in this manner.

Event payload can also be processed through an Inbound Web Service, which sends the payload to a file location where it is processed via batch processing. See Event Processing Inbound Web Service for more information. This approach is most often used when receiving real-time events from a head end system.

This section outlines the specific configuration necessary to process CIM-based payloads.

Creating CIM Payload Processing Extendable Lookup Values

Payload processing of CIM usage and events requires configuration of values for the SGG Payload Processing Configuration extendable lookup. This section outlines the specific configuration for usage and event payload processing extendable lookups for CIM usage and events. See Creating SGG Payload Processing Extendable Lookup Values for more information about this extendable lookup including parameters not listed in this section.

Usage

Extendable lookup values for CIM usage payload processing should include the following parameters:

- **Payload Handler Class Name**: com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits
- **Payload Processing Result Type**: Initial Measurements
- **IMD Seeder BO (Interval)**: IMD Seeder (D1-IMDSeeder) (Applicable only If processing interval usage)
- **IMD Seeder BO (Scalar)**: IMD Seeder (D1-IMDSeeder) (Applicable only If processing scalar usage)
- **Dynamic Parameters**: As appropriate for your implementation

Copyright © 2026, Oracle and/or its affiliates.

- **User Exit Interceptors**:

Payload Processing User Exit Type	Groovy Library Script	Groovy Library Method
On Get Parser	DG_CimUPrXML (SGG ADK Generic CIM XML Parser)	getParser
On Get Transformer	DG_CimUTrXML (SGG ADK Generic CIM XML Transformer)	get Transformer

Device Events

Extendable lookup values for CIM event payload processing should include the following parameters:

- **Payload Handler Class Name**: com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits
- **Payload Processing Result Type**: Device Events
- **Device Event Seeder BO**: Device Event Seeder (D1-DeviceEventSeeder)
- **Dynamic Parameters**: As appropriate for your implementation
- **User Exit Interceptors**:

Payload Processing User Exit Type	Groovy Library Script	Groovy Library Method
On Get Parser	DG_CimEvntPr (CIM Events Parser using XMLReader)	getParser
On Get Transformer	DG_CimEvntTr (CIM Events Transformer)	get Transformer

CIM Event Processing Inbound Web Service

Event processing can be handled via the EventMessage operation of the ADK CIM Events and Alarms (DG-CIMEvents) SOAP-based inbound web service. This approach is most often used when a head end system sends real-time events to Smart Grid Gateway.

When invoking this webservice, the event payload should be included in the web service call. Refer to the International Standard document for detailed information about the event data to be included in the web service call.

The ADK CIM Events and Alarms (DG-CIMEvents) service sends the payload to a file location where it is processed via batch processing.

The file location is defined in the **Arrived Data Save Locations** section of the ADK - Master Configuration, which includes the following parameters:

- **File Storage**: A File Storage Configuration (F1-FileStorage) extendable lookup value that defines the staging area to which the payloads will be saved. See **Creating File Storage Extendable Lookup Values** (LINK) for more information.
- **Output Bucket/Folder Name**: The bucket name (if saving to Oracle Object Storage), or a directory below the specified file storage location (if saving to a local file system).

Copyright © 2026, Oracle and/or its affiliates.

- **Filename Generation Script**: The name of a service script used to compose the filename when saving the incoming payload. The Default Filename Generator (D1DEFFNGEN) script is used by default if not specified.
- **XSL Transformer Template**: The name of an XSL used to create the event payload from the incoming inbound message data. Select the "ADK - CIM EndDeviceEvents Output" (DG-CIMEndDvcEvtsOutput.xsl) XSL.
- **File Extension**: An extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

CIM Payload Processing Data Mapping and Extendable Lookups

Unit of Measure Mapping

Unit of measure mapping is based on the ReadingType string in the CIM payload and uses the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup. Refer to the International Standard document for detailed information about the ReadingType data element.

For example, a payload with a ReadingType string of "0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0" could be mapped to the "Kilowatt Hour" unit of measure as follows:

- **Head-End Unit of Measure**: 0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0
- **Unit of Measure**: Kilowatt Hour

Interval Size Mapping

Interval size (expressed as seconds per interval or SPI) mapping is based on the measuringPeriod value extracted from the ReadingType string and uses the ADK CIM Measuring Period Lookup (DG-CIMMeasuringPeriodLookup) extendable lookup. The measuringPeriod is the third element in the ReadingType string. Refer to the International Standard document for detailed information about the measuringPeriod and ReadingType elements.

For example, the measuringPeriod in a payload with a ReadingType string of "0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0" is "2" (the third value in the string). This could be mapped to an SPI of 900 (15 minutes) as follows:

- **Measuring Period**: 2
- **Interval Size**: 00:15:00

Condition Code Mapping

Condition code mapping is based on the ReadingQualityType string in the CIM payload (if present) and uses the the ADK - Interval Sts Code to Cond Map (DG-IntStsCodeToCondMapLookup) extendable lookup. Refer to the International Standard document for detailed information about the ReadingQualityType data element.

For example, a payload with a ReadingQualityType string of "3.8.8" could be mapped to the "System Estimate" condition code as follows:

- **Interval Status**: 3.8.8
- **Condition**: System Estimate

Copyright © 2026, Oracle and/or its affiliates.

Event Type Mapping

Event type mapping is based on the EndDeviceEventType string in the CIM payload and uses the the ADK - Device Event Mapping (DG-DeviceEventMappingLookup) extendable lookup. Refer to the International Standard document for detailed information about the EndDeviceEventType data element.

For example, a payload with a EndDeviceEventType string of "3.12.0.257"" could be mapped to the "Tamper Detected" standard event name as follows:

- **Head-End System Event Name**: 3.12.0.257
- **Standard Event Name**: Tamper Detected

CIM Payload Processing Sample Payloads

This section provides sample CIM usage and event payloads. Refer to the International Standard document for detailed information about the data elements in these examples.

Usage

The following example represents a meter reading containing four KWH interval values and four KW scalar readings.

```
<mr:MeterReadings
xmlns:mr="http://iec.ch/TC57/2011/MeterReadings#">
    <mr:MeterReading>
        <mr:Meter>
            <mr:Names>
                <mr:name>GDDGCIM_INT_CLOUD_TESTHARNESS_01</mr:name>
                <mr:NameType>
                    <mr:description>This is an endpoint serial number</
mr:description>
                    <mr:name>meterID</mr:name>
                    <mr:NameTypeAuthority>
                        <mr:description>AssetManagementSystem</mr:description>
                        <mr:name>com.company.assets</mr:name>
                    </mr:NameTypeAuthority>
                </mr:NameType>
            </mr:Names>
        </mr:Meter>
        <mr:IntervalBlocks>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T09:30:47Z</mr:timeStamp>
                <mr:value>25.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:15:00Z</mr:end>
                    <mr:start>2022-02-03T00:00:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:00:47Z</mr:timeStamp>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                <mr:value>26.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:30:00Z</mr:end>
                    <mr:start>2022-02-03T00:15:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:30:47Z</mr:timeStamp>
                <mr:value>27.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:45:00Z</mr:end>
                    <mr:start>2022-02-03T00:30:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:30:47Z</mr:timeStamp>
                <mr:value>28.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T01:00:00Z</mr:end>
                    <mr:start>2022-02-03T00:45:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </mr:IntervalBlocks>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:15:00Z</mr:timeStamp>
            <mr:value>9.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:30:00Z</mr:timeStamp>
            <mr:value>10.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:45:00Z</mr:timeStamp>
            <mr:value>11.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T01:00:00Z</mr:timeStamp>
            <mr:value>12.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:valuesInterval>
            <mr:end>2022-02-03T01:00:00Z</mr:end>
            <mr:start>2022-02-03T00:00:00Z</mr:start>
        </mr:valuesInterval>
    </mr:MeterReading>
</mr:MeterReadings>
```


Copyright © 2026, Oracle and/or its affiliates.

Events

The following sample represents a single "tamper" device event. Note that this sample is the event data ONLY. This is NOT an example of the data sent to the ADK CIM Events and Alarms inbound web service.

```
 <end:EndDeviceEvents
xmlns:end="http://iec.ch/TC57/2011/EndDeviceEvents#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <end:EndDeviceEvent>
        <end:createdDateTime>2022-02-10T21:54:53.404+01:00</
end:createdDateTime>
        <end:issuerID>External System 1</end:issuerID>
        <end:issuerTrackingID>123</end:issuerTrackingID>
        <end:severity>5</end:severity>
        <end:Assets>
            <end:mRID>01f8b2e5-a677-4f2c-a6c0-1e79b409c55e</end:mRID>
            <end:Names>
                <end:name>METERID</end:name>
                <end:NameType>
                    <end:name>meterID</end:name>
                    <end:NameTypeAuthority>
                        <end:name>Utility ABC</end:name>
                    </end:NameTypeAuthority>
                </end:NameType>
            </end:Names>
        </end:Assets>
        <end:EndDeviceEventType ref="3.12.0.257"/>
    </end:EndDeviceEvent>
</end:EndDeviceEvents>
```

Smart Meter Commands

This section provides information about implementing smart meter commands with native implementations and Oracle Utilities cloud services, including Oracle Utilities Meter Solution Cloud Service.

- Device Communication Overview
- Smart Meter Command Flows
- Sending Outbound Communications
- Receiving Inbound Communications
- Adapter Development Kit - MultiSpeak Commands
- Adapter Development Kit - Common Information Model Commands
- Adapter Development Kit - Creating Custom Commands
- Cloud Service Test Harness
- Web Service Authentication

Device Communication Overview

Copyright © 2026, Oracle and/or its affiliates.

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The head-end system receives the message, acts on the request, and returns a reply.

See Communications in the *Business User Guide*, and About Communications in the *Administrative User Guide* for basic information about command activities and communications.

Native Implementation Command Processing

When using native implementation of Smart Grid Gateway adapters, including with Oracle Utilities cloud services, command processing works slightly differently than it does in midddleware (on-premises) implementations.

The table below provides a brief description of the communication process, and lists example objects used by the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay. Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Example Data
1.	A user initiates a remote connect command for a device. A remote connect activity business object is instantiated for the command.	Activity Business Object: Remote Connect (D1-RemoteConnect)
2.	The remote connect command activity business object creates an outbound communication. The specific type of outbound communication business object created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Connection Ready” status of the command activity business object’s lifecycle).	Outbound Communication Business Object: Itron - Reconnect Meter (Remote Connect) (D8-ReconnectMeterDR)
3.	The outbound communication creates an outbound message. The specific type of outbound message created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle).	Outbound Message Type Business Object: Outbound Message for Itron connect command (D8-ItronRemoteConnectOutbndMsg)
4.	The outbound message is sent to the head end system via an External System, Message Sender, and Outbound Message Type. The head end system sends a synchronous response to acknowledge receipt of the request.	External System: Itron OpenWayHead End System Message Sender: Itron – Remote Connect (D8-Reconnect) Outbound Message Type: Itron – Remote Connect (D8-RemoteConnect)

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Example Data
5.	As the status of the command request is updated, the head end system sends periodic messages. These messages are received by an Inbound Web Service which creates instances of an inbound communication business object. The specific type of inbound communication business object created is determined by the Inbound Web Service.	Inbound Communication Business Object: Itron – StatusChanged (D8-StatusChanged)
6.	When the command request has been completed, it sends a status changed inbound communication with a “Completed” status. This transitions the outbound communication business object to its “Completed” state.	Inbound Communication Business Object: Itron – StatusChanged (D8-StatusChanged)
7.	The remote connect command activity business object creates a “results” outbound communication. The specific type of outbound communication business object created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Connection Ready” status of the command activity business object’s lifecycle).	Outbound Communication Business Object: Itron - Reconnect Meter Results (D8-ReconnectMeterResultDR)
8.	The outbound communication creates an outbound message. The specific type of outbound message created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle).	Outbound Message Type Business Object: Outbound Message for Itron connect result command (D8-ItronRemoteCntRsltOutMsg)
9.	An Enter Algorithm on the “Evaluate Response” status of the outbound message business object’s lifecycle evaluates the response.	Enter Algorithm: Evaluate Get Reconnect Results Response (D8-EVGRCMRST)
10.	The “result” outbound communication creates a completion event to update the status of the device to indicate it has been connected. The specific type of completion event business object created is specified in an Enter algorithm on the “Create Completion Event” Status of the outbound communication business object’s lifecycle.	Algorithm: Create Connect Completion Event from Result (D8-CRCNCER) Completion Event Business Object: Connect Device (D1-ConnectDevice)

Copyright © 2026, Oracle and/or its affiliates.

Step	Process	Example Data
11.	The outbound communication updates the “Connect/Disconnect Completion Flag” and the original activity business object. This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle.	

In the case of commands that also request usage readings, such as On Demand Read or Scheduled Read, usage readings are sent separately via DataArrived messages separately to the DataSubscriberService Inbound Web Service, where they can be routed to Object storage for payload processing.

Smart Meter Command Flows

This section provides information about the smart meter command flows used by supported Smart Grid Gateway adapters.

- Itron OpenWay Command Flows
- Landis+Gyr Command Flows
- Sensus Command Flows
- Silver Spring Networks Command Flows
- Silver Spring Networks MultiSpeak Command Flows
- Filtering On-Demand Read Payloads

Itron OpenWay Command Flows

The table below lists the communication flows used with each Itron OpenWay command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
  - **Result Outbound Communication** The outbound communication business object used to request the result of a command request
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Commissioning	Itron - Add Meter Definition (Commission) (D8-AddMeterDefinitionsDR)		Device Commissioning Completion Event
Device Decommissioning	Itron - Deregister Meter (Decommission) (D8-DeregisterMeterDR)		Device Decommissioning Completion Event
Remote Connect	Itron - Reconnect Meter (Remote Connect) (D8-ReconnectMeterDR) **Result**: Itron - Reconnect Meter Result (D8-ReconnectMeterResultDR)	Itron - StatusChanged (D8-StatusChanged)	Connect Device Completion Event
Remote Disconnect	Itron - Meter Remote Disconnect (D8-DisconnectMeterDR) **Result**: Itron - Disconnect Meter Result (D8-DisconnectMeterResultDR)	Itron - StatusChanged (D8-StatusChanged)	Disconnect Device Completion Event
On-Demand Read (Scalar)	Itron - Contingency Read (Scalar) (D8-ReadScalarDR) **Result**: Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR)	Itron - StatusChanged (D8-StatusChanged)	Create IMD Completion Event
On-Demand Read (Interval)	Itron - Contingency Read (Interval) (D8-ReadIntervalDR) **Result**: Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR)	Itron - StatusChanged (D8-StatusChanged)	Create IMD Completion Event
Scheduled Read (Scalar)	Itron - Interrogate By Group (D8-InterrogateByGroupDR) **Result**: Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR)	Itron - StatusChanged (D8-StatusChanged)	Create IMD Completion Event
Scheduled Read (Interval)	Itron - Interrogate By Group (D8-InterrogateByGroupDR) **Result**: Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR)	Itron - StatusChanged (D8-StatusChanged)	Create IMD Completion Event
Device Status Check	Itron - Ping By Endpoints (Status Check) (D8-PingByEndpointsDR)	Itron - StatusChanged (D8-StatusChanged)	

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
	Result: Itron - Ping By Endpoints Result (D8- PingByEndpointsResultDR)		
Multi-Device Status Check	Itron – Ping By Endpoints Multi-Device (D8-PingByEndpointsMultiDvcDR)	Itron – StatusChanged (D8-StatusChanged)	
	Result: Itron – Ping By Endpoints Multi Dvc Result (D8-PingByEndpointsMDResultDR)		
Load Check	Itron – Detect Load Side Voltage (D8-DetectLoadSideVoltgeByMtrDR)	Itron – StatusChanged (D8-StatusChanged)	
	Result: Itron – Detect Load Side Voltage Result (D8-DetLoadSideVoltageMtrRsltDR)		
Read Disconnect State	Itron - Read Disconnect State (D8-ReadDisconStateByMtrDR)	Itron – StatusChanged (D8-StatusChanged)	
	Result: Itron - ReadDisconnectStateByMeter Result (D8-ReadDisconStByMtrResultDR)		

Batch Processing

The Inbound Communication Wait - Monitor (D1-IBCOM) batch control is defined on the "Pending" state of the Itron - StatusChanged (D8-StatusChanged) business object. The system should be configured to run this process frequently (every few minutes) to ensure timely processing of inbound communications created by the response from the head end system.

Landis+Gyr Command Flows

The table below lists the communication flows used with each Landis+Gyr command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Commissioning	L+G Add Meter to Inventory(MultiSpeak) (D3-AddMeterToInventryMultiSpDR)		Device Commissioning Completion Event
Device Decommissioning	L+G Meter Remove Notification (MultiSpeak) (D3-MtrRmvNotifMultiSpeakDR)		Device Decommissioning Completion Event
Meter Retire	L+G Meter Retire Notification (D3-MeterRetireNotificationDR)		
Meter Exchange	L+G Meter Exchange Notification (D3-MeterExNotificatnMultiSpDR)		
Remote Connect	L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR)	Con/Discon State Change Notif-Direct Route (D3-ConnectDisconStateChgNtfDR)	Connect Device Completion Event
Remote Disconnect	L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR)	Con/Discon State Change Notif-Direct Route (D3-ConnectDisconStateChgNtfDR)	Disconnect Device Completion Event
Demand Reset	L+G Schedule Demand Reset (Multispeak) (D3-ScheduleDemandResetDR)	Reading Changed Notification Direct Route (D3-ReadingChgNotificationDR)	
On-Demand Read (MultiSpeak)	L+G MultiSpeak Initiate Meter Read By Meter Num Direct Route (D3-InitiateMRByMtrNbrDR)	Reading Changed Notification Direct Route (D3-ReadingChgNotificationDR)	Create IMD Completion Event
On-Demand Read (Interval) [CIM]	CIM Meter On Demand Read Direct Route (Interval) (D3-CIMGetLPDataDR)	CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR)	Create IMD Completion Event
On-Demand Read (Scalar) [CIM]	CIM Meter On Demand Read Direct Route (Scalar) (D3-CIMMeterOnDemandReadDR)	CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR)	Create IMD Completion Event
Device Status Check [CIM]	CIM Ping Direct Route (D3-CIMPingDR)	CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR)	

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
On-Demand Read (Interval) [CIM 2.0]	L+G CIM 2.0 On-Demand Read DR (Interval) (D3-CIMGetMtrReadIntervalDR)	L+G CIM 2.0 - On-Demand Read Response DR (D3-CIMMtrReadResponseDR)	Create IMD Completion Event
On-Demand Read (Scalar) [CIM 2.0]	L+G CIM 2.0 - On-Demand Read DR (Scalar) (D3-CIMGetMtrReadScalarDR)	L+G CIM 2.0 - On-Demand Read Response DR (D3-CIMMtrReadResponseDR)	Create IMD Completion Event
Device Status Check [CIM 2.0]	L+G CIM 2.0 - Ping DR (D3-CIM2PingDR)	L+G CIM 2.0 - Ping Response DR (D3-CIMPingResponseDR)	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CRRCCEDR (Create IMD Completion Event -Direct Route) algorithm on the Create Completion Events status of the D3-ReadingChgNotificationDR (L+G Reading Changed Notification Direct Route) business object.

The D3-VALDVCFDR (Check if Device has any Filtering errors Direct Route) algorithm on the Validate status of the D3-ReadingChgNotificationDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the same parameter on the D3-CRRCCEDR algorithm. The “Opt Out Of Filtering” parameter should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

CIM Commands

For CIM-based on-demand read commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CRCIMCE (Create CIM Completion Event Direct Route) algorithm on the Create Completion Events status of the D3-CIMAsyncReplyMeterReadingDR (CIM AsyncReply MeterReads Direct Route) business object.

The D3-VALCCIMCE (Check if Device has Filtering errors CIM Completion Event Direct Route) algorithm on the Validate status of the D3-CIMAsyncReplyMeterReadingDR (CIM AsyncReply MeterReads Direct Route) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CRCIMCE algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Sensus Command Flows

Copyright © 2026, Oracle and/or its affiliates.

The table below lists the communication flows used with each Sensus command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Commissioning (MultiSpeak)	Sensus -Meter Add Notification Direct Route (D6-MeterAddNotificationDR)		Device Commissioning Completion Event
Device Decommissioning (MultiSpeak)	Sensus -Meter Remove Notification Direct Route (D6-MeterRemoveNotificationDR)		Device Decommissioning Completion Event
Remote Connect (MultiSpeak)	Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR)	Sensus Conn/Discon State Chg Direct Route (D6-ConnectDisconStateChgNtfDR)	Connect Device Completion Event
Remote Disconnect (MultiSpeak)	Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR)	Sensus Conn/Discon State Chg Direct Route (D6-ConnectDisconStateChgNtfDR)	Disconnect Device Completion Event
On-Demand Read (MultiSpeak)	Sensus - Initiate Meter Read By Meter ID (D6-InitiateMeterByMeterIdDR)	Sensus - Reading Changed Notification (D6-ReadingChgNotificationDR)	Create IMD Completion Event
Device Status Check (MultiSpeak)	Sensus Initiate Outage Detection DR (D6-InitiateOutageDetectionDR)	Sensus Outage Detect Event Notif Direct Route (D6-ODEventNotificationDR)	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D6-CCOMPEVDR (Create IMD Completion Event Direct Route) algorithm on the Create Completion Events status of the D6-ReadingChgNotificationDR (Sensus - Reading Changed Notification DR) business object.

Copyright © 2026, Oracle and/or its affiliates.

The D6-VALDVCFDR (Check if Device has any Filtering errors Direct Route) algorithm on the Validate status of the D6-ReadingChgNotificationDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D6-CCOMPEVDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Silver Spring Networks Command Flows

The table below lists the communication flows used with each Silver Spring Networks command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
  - **Get Status Outbound Communication**: The outbound communication business object used to request the status of a command request
  - **Result Outbound Communication**: The outbound communication business object used to request the result of a command request
- **Outbound Messages**: The outbound message business objects for one or more outbound messages created by the outbound communication. These should be configured on the Silver Spring Networks external system.
- **Inbound Communications**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Command	Outbound Communication	Outbound Messages	Completion Event
Device Commissioning (see Note below)	SSN - Replace Location (D7-ReplaceLocationDR)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Replace Location**: SSN - Replace Location (Comm) OB Message (D7-ReplaceLocationOBMsg)	Device Commissioning Completion Event
Device Decommissioning	SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Replace Device at Location**: SSN - Replace Device At Loc O/B Message: (D7-ReplaceDeviceAtLocOBMsg)	Device Decommissioning Completion Event

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Outbound Messages	Completion Event
Remote Connect	SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Connect Disconnect**: SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg)	Connect Device Completion Event
	Get Status: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)	
	Result: SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR)	SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg)	
Remote Disconnect	SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Connect Disconnect**: SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg)	Disconnect Device Completion Event
	Get Status: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)	
	Result: SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR)	SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg)	
On-Demand Read (Scalar)	SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Add Meter Read Job Scalar**: SSN - Add Meter Read Job Scalar OB Msg Type (D7-AddMeterReadJobScalarOBMsg)	Create IMD Completion Event

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Outbound Messages	Completion Event
	Get Status: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)	
	Result: SSN - Meter Read Results (Scalar) (D7-MeterReadResultsScalar)	SSN - GetMeterReadResults(Scalar) By JobId (D7-GetSclrMtrRdRstByJobIdOBMsg)	
On-Demand Read (Interval)	SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Add Meter Read Job Interval**: SSN - AddMeterReadJob Interval OB Msg Type (D7-AddMeterReadJobIntervalOBMsg)	Create IMD Completion Event
	Get Status: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)	
	Result: SSN - Meter Read Results (Interval) (D7-MeterReadResultsInterval)	SSN - GetMeterReadResults(Interval) By JobId (D7-GetIntMtrRdRsltByJobIdOBMsg)	
Device Status Check	SSN - Add Ping Job - Polling (D7-AddPingJobDRP)	**Find Device**: SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg) **Add Ping Job**: SSN - Add Ping Job OB Message (D7-AddPingJobOBMsg)	
	Get Status: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)	
	Result: SSN - Ping Results (D7-PingResultsDR)	SSN - Get Ping Results O/B Message (D7-GetPingResultsOBMsg)	

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Service points must be provisioned in the Silver Spring Networks (Itron AMM) system before the Device Commissioning command can be used.

Configuring Communication Types for Job Status Polling

The following outbound communication business objects (from the table above) all have corresponding communication types that must be configured before outbound communications based on those business objects can be created. The table below lists the communication type business objects used with each.

Outbound Communication Business Object	Communication Type Business Object
SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP)	SSN - Add Remote Prov Job Polling DR Type (D7-AddRemoteProvDRPType)
SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP)	SSN - Add Meter Read Job (Scalar) Polling DR Type (D7-AddMeterReadJobScalrDRPType)
SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP)	SSN - Add Meter Read Job (Interval) Polling DR Type (D7-AddMeterReadJobIntDRPType)
SSN - Add Ping Job - Polling (D7-AddPingJobDRP)	SSN - Add Ping Job Polling DR Type (D7-AddPingJobDRPType)
Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Polling DR Type (D7-GetJobStatusDRPType)

Refer to **Communication Types** for more information about communication types.

Get Job Status Polling Parameters and Polling Modes

The **Get Job Status Polling** parameters on these communication types are used to define how polling works with outbound communications of the corresponding type. These include **Retry Frequency** parameters (the minimum amount of time (expressed as hours:minutes:seconds) the system will wait between polling iterations for jobs created by communications of this type) and **Maximum Retries** parameters (the maximum number of times the system will try to poll for the status of jobs created by communications of this type), which must be configured in order to use these outbound communications.

There are two modes of polling:

- **Initial** polling occurs immediately and within the same transaction as any previous portion of the command. This setting is useful when the head end is expected to quickly complete the job (within a few seconds). To use Initial polling, define values for the **Initial Polling Retry Frequency** and **Initial Polling Maximum Retries** parameters.
- **Out of Process** polling occurs within its own transaction triggered either separately by a user or by a batch process. This mode is useful when the head end system is likely to take more than a few seconds to complete the job. The **Polling Retry Frequency** and **Polling Maximum Retries** parameters are used with Out of Process polling. If these parameters are not configured on the communication type, the values defined in the **SSN - Get Job Status Configuration** section of the Silver Spring Networks Version Master Configuration are used in their place.

Refer to the **Get Job Status Polling** embedded help on the above communication types for more information about polling modes and parameters.

Copyright © 2026, Oracle and/or its affiliates.

Polling and Batch Processing

When a communication enters the "Polling Ready" state, it polls the head end system via a Get Status Outbound Communication. If the initial polling parameters have been exceeded (or the initial polling parameters were not defined), the communication transitions to the "Polling in Progress" state. Communications in this state can be transitioned using the Outbound Communication Polling - Monitor (D7-OCPL) batch process (or the **Poll Job Status** button in the **Outbound Communication** zone in the **Communication** portal.)

If using the Outbound Communication Polling - Monitor (D7-OCPL) batch process, the system should be configured to run the process frequently (every few minutes).

Polling and the creation of Get Job Status outbound communications requires the following "Get Job Status" processing method be defined for the Silver Spring Networks head end system:

- **Processing Role**: Get Job Status
- **Default Business Object**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)

See **Silver Spring Networks Processing Methods** for more information about processing methods used with Silver Spring Networks.

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D7-CCOMPEVSR (SSN - Create Completion Event Scalar) algorithm on the Create Completion Events status of the D7-MeterReadResultsScalar (SSN - Meter Read Results (Scalar)) business object (for scalar on-demand read commands) and/or on the D7-CCEVTINDR (SSN - Create Completion Event Interval) algorithm on the Create Completion Events status of the D7-MeterReadResultsInterval (SSN - Meter Read Results (Interval)) business object (for interval on-demand read commands).

When filtering is enabled, the “Filter if invalid external MC identifier” parameter on the D7-CCOMPEVSR and D7-CCEVTINDR algorithms controls what should happen if a measuring component cannot be found using the MC Identifier Number sent from the head end system. When set to 'Yes' and no measuring component is found, the initial measurement will be filtered. When set to 'No' and no measuring component is found, the system will attempt to find the measuring component using the UOM and TOU. The value of the “Filter if invalid external MC identifier” on the D7-CCOMPEVSR and D7-CCEVTINDR algorithms must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either ‘yes’ or ‘no’.

Deprecated Outbound Communications

The outbound communication business objects listed in the table above replace the following outbound communication business objects, which will be deprecated in a future release.

Command	Deprecated Outbound Communication Business Object
Remote Connect/Remote Disconnect	SSN - Connect or Disconnect (D7-ConnectDisconnectDR)

Copyright © 2026, Oracle and/or its affiliates.

Command	Deprecated Outbound Communication Business Object
On-Demand Read (Scalar)	SSN - Add Meter Read Job (Scalar) (D7-AddMeterReadJobScalarDR)
On-Demand Read (Interval)	SSN - Add Meter Read Job (Interval) (D7-AddMeterReadJobIntervalDR)
Device Status Check	SSN - Add Ping Job (D7-AddPingJobDR)

Silver Spring Networks MultiSpeak Command Flows

The table below lists the communication flows used with each Silver Spring Networks MultiSpeak command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Commissioning (MultiSpeak)	SSN - MultiSpeak Meter Add Notification DR (D7-MSPMeterAddNotificationDR)		Device Commissioning Completion Event
Device Decommissioning (MultiSpeak)	SSN - MSP Meter Remove Notification DR (D7-MSPMeterRemNotifyDR)		Device Decommissioning Completion Event
Remote Connect (MultiSpeak)	SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR)	SSN - MSP Conn/Discon State Change DR (D7-MSPConDisconStaChgNtfDR)	Connect Device Completion Event
Remote Disconnect (MultiSpeak)	SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR)	SSN - MSP Conn/Discon State Change DR (D7-MSPConDisconStaChgNtfDR)	Disconnect Device Completion Event

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
On-Demand Read (MultiSpeak)	SSN - MSP Initiate Read By Meter ID DR (D7-MSPInitMtrByMtrIdDR)	SSN - MSP Reading Changed Notification DR (D7-MSPReadingChgNotifyDR)	Create IMD Completion Event
Device Status Check (MultiSpeak)	SSN - MSP Initiate Outage Detection DR (D7-MSPInitOutageDetectDR)	SSN - MSP OD Event Notification DR (D7-MSPODEventNotificationDR)	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the MSP Create IMD Completion Event DR (D1-MSCCEDR) algorithm on the Create Completion Events status of the MultiSpeak Reading Changed Notification DR (D1-MSPReadingChgNotifyDR) business object.

The MSP Check if Device has any Filtering errors DR (D1-MSVADVFDR) algorithm on the Validate status of the D1-MSPReadingChgNotifyDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D1-MSCCEDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Filtering On-Demand Read Payloads

By default, the On-Demand Read command is initiated for a specific meter (device) and returns readings (measurement data) for each register or channel (measuring component) on the meter. However, in some circumstances, an implementation may not want measurements returned for every channel. For instance, they may only want or need billing-related data, or only a subset of the data recorded by the meter.

Measurement payloads received via On-Demand Read commands can be filtered based on unit of measure (UOM) by adding custom logic to the XLS files used to send and receive On-Demand Read command.

This filtering can be implemented by editing the appropriate XSL to include a variable containing a list of "approved UOMs" list and logic that filters the payload based on the "approved list" variable.

Examples

The lines noted by “<!-- List of approved UOMS -->” and “<!-- UOM Filtering here -->” in the following excerpts from Landis+Gyr XSL files illustrate how the files can be edited to apply UOM filtering.

Copyright © 2026, Oracle and/or its affiliates.

L+G Reading Changed Notification Request (D3-ReadingChgNotifRequest.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet exclude-result-prefixes="soap11 soap12 ns1" xmlns:ns1="http://
www.multispeak.org/Version_3.0" xmlns:soap12="http://www.w3.org/2003/05/soap-
envelope" xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:import href="D3-MultiSpeakMsgHeader.xsl"/>
    <!-- List of approved UOMS -->
    <xsl:variable name="Approved_UOMs">|kWh|</xsl:variable>
  <xsl:template match="soap11:Envelope">
    <xsl:apply-templates select="//ns1:ReadingChangedNotification"/>
    ...
<xsl:template match="ns1:readingValues">
  <xsl:for-each select="./ns1:readingValue">
    <readingValue>
      <extensions>
        <xsl:value-of select="./ns1:extensions"/>
      </extensions>
      <extensionsList>
        <xsl:apply-templates select="./ns1:extensionsList"/>
      </extensionsList>
      <units>
        <xsl:value-of select="./ns1:units"/>
      </units>
      <value>
        <xsl:value-of select="./ns1:value"/>
      </value>
      <!-- UOM Filtering here -->
      <xsl:if test="contains($Approved_UOMs, concat('|', ./ns1:name , '|'))">
        <name>
          <xsl:value-of select="./ns1:name"/>
        </name>
      </xsl:if>
      <dateTime>
        <xsl:value-of select="./ns1:dateTime"/>
      </dateTime>
    </readingValue>
  </xsl:for-each>
</xsl:template>
...
```

SGG L+G CIM AsyncReply MeterReadings Request XSL (D3-CIMMeterReadingRequest.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet exclude-result-prefixes="xsl env sec msg mtr ede xsi"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ede="http://
iec.ch/TC57/2009/EndDeviceEvents#" xmlns:mtr="http://iec.ch/TC57/2009/
MeterReadings#" xmlns:msg="http://www.iec.ch/TC57/2008/schema/message"
xmlns:sec="http://www.landisgyr.com/iec61968/2010/03" xmlns:env="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:xsl="http://www.w3.org/1999/XSL/
Transform" version="1.0">
<xsl:output indent="yes" encoding="utf-8" method="xml"/>
  <!-- List of approved UOMS -->
```


Copyright © 2026, Oracle and/or its affiliates.

```
  <xsl:variable name="Approved_UOMs">|kWh|</xsl:variable>
<xsl:template match="env:Envelope">
  <xsl:apply-templates select="./env:Body/msg:ResponseMessage"/>
</xsl:template>
...

<xsl:template match="mtr:Readings">
  <xsl:if test="string-length(./*) > 0">
    <Readings>
      <timeStamp>
        <xsl:value-of select="./mtr:timeStamp"/>
      </timeStamp>
      <xsl:apply-templates select="./mtr:value" mode="copyif"/>
      <xsl:for-each select="./ReadingQualities[string-length(./mtr:quality) >
0]">
        <ReadingQualities>
          <quality>
            <xsl:value-of select="./mtr:quality"/>
          </quality>
        </ReadingQualities>
      </xsl:for-each>
      <xsl:if test="string-length(./mtr:ReadingType/@ref) > 0">
        <!-- UOM Filtering here -->
        <xsl:if test="contains($Approved_UOMs, concat('|', ./mtr:ReadingType/
@ref, '|'))">
          <ReadingTypeRef>
            <xsl:value-of select="./mtr:ReadingType/@ref"/>
          </ReadingTypeRef>
        </xsl:if>
      </xsl:if>
    </Readings>
  </xsl:if>
</xsl:template>
...
```

Head End System XSL Files

The table below lists the XSL files that can be used to filter payloads using this approach for each head end system.

XSL files can be edited using the **Managed Content** portal.

Head End System	XSL	Where Used
Landis+Gyr	D3-CIMMeterReadingRequest.xsl	Inbound Web Service (CIM Service Callback)
Landis+Gyr	D3-ReadingChgNotifRequest.xsl	Inbound Web Service (Reading Changed Notification (Callback))
Sensus	D6-ReadingChgNotifRequest.xsl	Inbound Web Service (Sensus MultiSpeak CB_Server Interface)
Silver Spring Networks	D7-GetMtrRdRsByJobIDResp.xsl	External System Message XSL (provided by implementation)

Sending Outbound Communications

Copyright © 2026, Oracle and/or its affiliates.

Sending outbound communications to a head end system involves configuring the following entities:

- **Outbound Message Types**: Define how outbound messages of a specific type are handled. Outbound message types are based on a specific outbound message business object that defines the data structure of the message. See Defining Outbound Message Types in the *Administrative User Guide* and Creating Outbound Message Types below for more information.
- **Message Senders**: Define details for sending messages to external systems. See Message Senders in the *Administrative User Guide* for more information.
- **External System**: Defines a set of outbound message types that can be sent to the head end system. See External Systems in the *Administrative User Guide* for more information.
- **Head End System**: Defines processing rules for the head end system, such as the specific activities and communications created for smart meter commands. See Head End Systems and Processing Methods in the *Administrative User Guide* and Configuration Processing Methods below for more information.
- Creating Outbound Message Types
- Creating Message Senders
- Creating an External System
- Configuring Processing Methods

Creating Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used. The tables below lists the outbound message types provided for each Smart Grid Gateway adapter.

Itron OpenWay Outbound Message Types

This table lists the outbound message types used with the Itron OpenWay adapter, along with the business objects used with each.

Command	Outbound Message Type / Business Object	"Result" Outbound Message Type / Business Object
Device Commissioning	Itron - Add Meter Definitions O/B Msg Type (D8-AMDOBM) Itron - Commissioning (D8-ItronCommissionOutboundMsg)	N/A
Device Decommissioning	Itron - Deregister Meter O/B Msg Type (D8-DREGOBM) Itron - Decommissioning (D8-ItronDecommissionOutbndMsg)	N/A
Remote Connect	Itron - Reconnect Meter O/B Msg Type (D8-RCNOBM) Itron - Remote Connect (D8-ItronRemoteConnectOutbndMsg)	Itron - Get Reconnect Meter Result O/B Msg Type (D8-GRCNOBM) Itron - Get Remote Connect Results (D8-ItronRemoteCntRsltOutMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Type / Business Object	"Result" Outbound Message Type / Business Object
Remote Disconnect	Itron - Disconnect Meter O/B Msg Type (D8-DISCOMOBM) Itron - Disconnect (D8-ItronRemoteDisconnectOutMsg)	Iron - Get Disconnect Meter Result O/B Msg Type (D8-GDMROBM) Itron - Get Disconnect Result (D8-ItronRemoteDcntRsltOutMsg)
On-Demand Read (Scalar)	Itron - Contingency Read O/B Msg Type (D8-CRDOBM) Itron - Contingency Read (D8-ItronOnDemandReadOutbndMsg)	Itron - Get Contingency Read Results O/B Msg Type (D8-GCRDOBM) Itron - Get Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg)
On-Demand Read (Interval)	Itron - Contingency Read O/B Msg Type (D8-CRDOBM) Itron - Contingency Read (D8-ItronOnDemandReadOutbndMsg)	Itron - Get Contingency Read Results O/B Msg Type (D8-GCRDOBM) Itron - Get Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg)
Scheduled Read (Scalar)	Itron - Scheduled Read O/B Msg Type (D8-SROBM) Itron - Schedule Read (D8-ItronInterrogateByGrpOBMsg)	Itron - Get Scheduled Read Result O/B Msg Type (D8-GSRROBM) Itron - Get Schedule Read Result (D8-ItronInterrogateByGrpRsltOB)
Scheduled Read (Interval)	Itron - Scheduled Read O/B Msg Type (D8-SROBM) Itron - Schedule Read (D8-ItronInterrogateByGrpOBMsg)	Itron - Get Scheduled Read Result O/B Msg Type (D8-GSRROBM) Itron - Get Schedule Read Result (D8-ItronInterrogateByGrpRsltOB)
Device Status Check	Itron - Ping By Endpoints O/B Msg Type (D8-DSCOBM) Itron - Device Status Check (D8-ItronDvcStatusChkOutbndMsg)	Itron - Get Ping By Endpoints Result O/B Msg Type (D8-GDSCROBM) Itron - Get Device Status Check Result (D8-ItronDvcStatusChkRstlOBMsg)
Multi-Device Status Check	Itron - Multi Ping By Endpoints O/B Msg Type (D8-MDSCOBM) Itron - Multi Device Status Check (D8-ItronDvcStChkMtDvcOutbndMsg)	Itron - Get Multi Ping By Endpoints Result O/B Msg Type (D8-GMDSCOBM) Itron - Get Multi Device Status Check Result (D8-ItronDvcStChkMtDvcRsltOBMsg)
Load Check	Itron - Detect Load Side Voltage O/B Msg Type (D8-DLSVOBM) Itron - Detect Load Side Voltage (D8-ItronDetectLoadSideVolOBMsg)	Itron - Get Detect Load Side Voltage Results O/B Msg Type (D8-GDLSVOBM) Itron - Get Detect Load Side Voltage Result (D8-ItronDetLSVoltMtrRsltOutMsg)

Landis+Gyr Outbound Message Types

This table lists the outbound message types used with the Landis+Gyr adapter, along with the business objects used with each.

Command	Outbound Message Type / Business Object
Device Commissioning	Add Meter to Inventory (L+G) (D3-AMIOMTYP) LG - Add Meter To Inventory (D3-AddMeterToInventoryOBMsg)
Device Decommissioning	Notify Remove Meter (L+G) (D3-MRENOMTYP) LG - Meter Remove Notification (D3-MeterRemoveNotifOBMsg)
Meter Retire	Notify Retire Meter (L+G) (D3-MRNOMTYP) LG - Meter Retire Notification (D3-MeterRetireNotifOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Type / Business Object
Meter Exchange	Notify Exchange Meter (L+G) (D3-MENOBMTYP) LG -Meter Exchange Notification (D3-MeterExchNotificatnOBMsg)
Remote Connect	Initiate Connect Disconnect by Meter Number (L+G) (D3-INCODIOBM) LG - Initiate Connect Disconnect (D3-InitConnectDisconnectOBMsg)
Remote Disconnect	Initiate Connect Disconnect by Meter Number (L+G) (D3-INCODIOBM) LG - Initiate Connect Disconnect (D3-InitConnectDisconnectOBMsg)
Demand Reset	Schedule Demand Reset (L+G) (D3-SDROMTYP) LG - Schedule Demand Reset (D3-ScheduleDemandResetOBMsg)
On-Demand Read (MultiSpeak)	Initiate Read by Meter Number (L+G) (D3-INMRMNOBM) LG - Initiate Meter Read By Meter Number (D3-InitMtrReadByMtrNumOBMsg)
On-Demand Read (Interval) [CIM]	Initiate On-Demand Read (CIM L+G) (D3-CIMODR) LG - CIM AMIRequest GetMeterReads (D3-CIMAMIRequestGetMtrRdOBMsg)
On-Demand Read (Scalar) [CIM]	Initiate On-Demand Read (CIM L+G) (D3-CIMODR) LG - CIM AMIRequest GetMeterReads (D3-CIMAMIRequestGetMtrRdOBMsg)
Device Status Check [CIM]	Initiate Device Status Check (CIM L+G) (D3-CIMDSC) LG - CIM AMI Request Get Meter Reads (D3-CIMAMIRequestGetMtrRdOBMsg)
On-Demand Read (Interval) [CIM 2.0]	L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR) L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)
On-Demand Read (Scalar) [CIM 2.0]	L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR) L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)
Device Status Check [CIM 2.0]	L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR) L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)

Sensus Outbound Message Types

This table lists the outbound message types used with the Sensus adapter, along with the business objects used with each.

Command	Outbound Message Type / Business Object
Device Commissioning	Notify Add Meter (Sensus) (D6-SMANOMTYP) Sensus - Meter Add Notification (D6-MeterAddNotificationOBMsg)
Device Decommissioning	Notify Remove Meter (Sensus) (D6-SMRNOMTY) Sensus -Meter Remove Notification OB Msg (D6-MeterRemoveNotificatnOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Type / Business Object
Remote Connect	Initiate Connect Disconnect by Meter Number (Sensus) (D6-INCODIOBM) Sensus - Initiate Connect Disconnect (D6-InitConnectDisconnectOBMsg)
Remote Disconnect	Initiate Connect Disconnect by Meter Number (Sensus) (D6-INCODIOBM) Sensus - Initiate Connect Disconnect (D6-InitConnectDisconnectOBMsg)
On-Demand Read (MultiSpeak)	Initiate Read by Meter Id (Sensus) (D6-SIMROMTY) Sensus - Initiate Meter Read By Meter ID (D6-InitMtrRdMtrIdOBMsg)
Device Status Check	Initiate Outage Detection (Sensus) (D6-SIODTOMTY) Sensus - Initiate Outage Detection (D6-InitOutageDetectionOBMsg)

Silver Spring Networks Outbound Message Types

This table lists the outbound message types used with the Silver Spring Networks adapter, along with the business objects used with each.

Command	Outbound Message Type / Business Object	"Result" Outbound Message Type / Business Object
Device Commissioning	Replace Location (Commission SSN) (D7-RLOCOBMTY) SSN - Replace Location Command (D7-ReplaceLocationOBMsg)	N/A
Device Decommissioning	Replace Device (Decommission SSN) (D7-RDVCOBMTY) SSN - Replace Device At Locaction (D7-ReplaceDeviceAtLocOBMsg)	N/A
Remote Connect	Connect Disconnect Device (SSN) (D7-CDCOBMTYP) SSN - Connect Disconnect (D7-ConnectDisconnectOBMsg)	Get Connect Disconnect Result (SSN) (D7-GCDROBMTY) SSN - Get Connect Disconnect Result (D7-GetCntDiscntResultOBMsg)
Remote Disconnect	Connect Disconnect Device (SSN) (D7-CDCOBMTYP) SSN - Connect Disconnect (D7-ConnectDisconnectOBMsg)	Get Connect Disconnect Result (SSN) (D7-GCDROBMTY) SSN - Get Connect Disconnect Result (D7-GetCntDiscntResultOBMsg)
On-Demand Read (Scalar)	Add Scalar Read Job (SSN) (D7-AMRJSCOBT) SSN - Add Meter Read Job Scalar (D7-AddMeterReadJobScalarOBMsg)	Get Scalar Read Results by Job Id (SSN) (D7-GMRRBJOMT) SSN - Get Meter Read Results(Scalar) (D7-GetSclrMtrRdRstByJobIdOBMsg)
On-Demand Read (Interval)	Add Interval Read Job (SSN) (D7-AMRJINOBT) SSN - Add Meter Read Job Interval (D7-AddMtrReadJobIntervalOBMsg)	Get Interval Read Results by Job Id (SSN) (D7-GIMRRJOMT) SSN - Get Meter Read Results(Interval) (D7-GetIntMtrRdRsltByJobIdOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Type / Business Object	"Result" Outbound Message Type / Business Object
Device Status Check	Add Meter Ping Job (SSN) (D7-APJOBM) SSN - Add Ping Job (D7-AddPingJobOBMsg)	Get Device Ping Job Results (SSN) (D7-GPJROM) SSN - Get Ping Results (D7-GetPingResultsOBMsg)

Silver Spring Networks MultiSpeak Outbound Message Types

This table lists the outbound message types used with the Silver Spring Networks adapter for MultiSpeak commands, along with the business objects used with each.

Command	Outbound Message Type / Business Object
Device Commissioning	SSN - MultiSpeak Notify Meter Add (D7-MSNMA) SSN - MultiSpeak Meter Add Notification (D7-MSPMeterAddNotifyOBMsg)
Device Decommissioning	SSN - MultiSpeak Notify Meter Remove (D7-MSNMR) SSN - MultiSpeak Meter Remove Notification (D7-MSPMeterRemoveNotifyOBMsg)
Remote Connect	SSN - MSP Initiate Con/Discon Meter Number (D7-MSICDBMN) SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg)
Remote Disconnect	SSN - MSP Initiate Con/Discon Meter Number (D7-MSICDBMN) SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg)
On-Demand Read (MultiSpeak)	SSN - MSP Initiate Meter Read by Meter Id (D7-MSMRBMI) SSN - MSP Initiate Meter Read By Meter ID (D7-MSPInitMtrRdMtrIdOBMsg)
Device Status Check	SSN - MultiSpeak Initiate Outage Detection (D7-MSIOD) SSN - MultiSpeak Initiate Outage Detection (D7-MSPInitOutageDetectionOBMsg)

Creating Message Senders

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

> **Note**
>
> When creating message senders as part of the command communication process used with Smart Grid Gateway adapters, the **HTTP URL** must point to the appropriate server and head end system version. If an implementation has more than one version of a given head end system, each can have its own server and URL.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When creating message senders as part of the command communication process used with native implementations, the **Message Class** MUST be set to **SOAPSNDR** (Sender for real-time HTTP / SOAP messages).

See Message Senders in the *Administrative User Guide* for more information about creating message senders.

Creating an External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

External System Outbound Message Types

You must configure an outbound message type on the external system for each type of message you will send to the head end system. An external system's outbound message types are defined by the following:

- **Outbound Message Type**: The outbound message type created for the message (see Creating Outbound Message Types)
- **Processing Method**: The method by which the message will be sent. This should be set to "Real-time" Smart Grid Gateway outbound communications.
- **Message Sender**: The message sender created for the message (see Creating Message Senders)
- **Message XSL / Response XSL**: XSL files used to transform your message and the response to the message into the appropriate format used by the head end system (Message XSL) and Smart Grid Gateway (Response XSL). Each Smart Grid Gateway adapter includes XSL files for each type of outbound communication.

See External Systems in the *Administrative User Guide* for more information about creating external systems.

Itron OpenWay XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Itron OpenWay adapter.

Outbound Message Business Object	Message XSL	Response XSL
Outbound Message for Itron Commission (D8-ItronCommissionOutboundMsg)	D8-CommissionRequest.xsl	D8-CommissionResponse.xsl
Outbound Message for Itron Decommission (D8-ItronDecommissionOutbndMsg)	D8-DecommissionRequest.xsl	D8-DecommissionResponse.xsl

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Business Object	Message XSL	Response XSL
Outbound Message for Itron connect command (D8-ItronRemoteConnectOutbndMsg)	D8-ReconnectRequest.xsl	D8-ReconnectResponse.xsl
Outbound Message for Itron connect result command (D8-ItronRemoteCntRsltOutMsg)	D8-GetReconnectRsRequest.xsl	D8-GetReconnectRsResponse.xsl
Outbound Message for Itron disconnect command (D8-ItronRemoteDisconnectOutMsg)	D8-DisconnectRequest.xsl	D8-DisconnectResponse.xsl
Outbound Message for Itron disconnect result command (D8-ItronRemoteDcntRsltOutMsg)	D8-GetDisconnectRsRequest.xsl	D8-GetDisconnectRsResponse.xsl
Outbound Message for Itron Contingency Read command (D8-ItronOnDemandReadOutbndMsg)	D8-ContRdByEndptsRequest.xsl	D8-ContRdByEndptsResponse.xsl
O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg)	D8-GetContRdByEndptsRsReq.xsl	D8-GetContRdByEndptsRsResp.xsl
Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg)	D8-InterrogateByGroupReq.xsl	D8-InterrogateByGroupResp.xsl
Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB)	D8-GetIntByGroupRsRequest.xsl	D8-GetIntByGroupRsResponse.xsl
Outbound Message for Itron Device status check (D8-ItronDvcStatusChkOutbndMsg)	D8-PingByEndpointsRequest.xsl	D8-PingByEndpointsResponse.xsl
O/B Msg for Itron DeviceStatusCheck Result (D8-ItronDvcStatusChkRstlOBMsg)	D8-GetPingByEndptsRsReq.xsl	D8-GetPingByEndptsRsResp.xsl
Outbound Message for Itron Multi device status check (D8-ItronDvcStChkMtDvcOutbndMsg)	D8-PingByEndpointsRequest.xsl	D8-PingByEndpointsResponse.xsl

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Business Object	Message XSL	Response XSL
O/B Message for Itron MultiDeviceStatusCheck Result (D8-ItronDvcStChkMtDvcRsltOBMsg)	D8-GetPingByEndptsRsReq.xsl	D8-GetPingByEndptsRsResp.xsl
Outbound Message for Itron Detect Load Side Voltage (D8-ItronDetectLoadSideVolOBMsg)	D8-DetLSVByMeterRequest.xsl	D8-DetLSVByMeterResponse.xsl
Outbound Message for LoadCheckResult Cmd (D8-ItronDetLSVoltMtrRsltOutMsg)	D8-GetDetLSVByMeterRsReq.xsl	D8-GetDetLSVByMeterRsResp.xsl
Outbound Message for Itron Read Disconnect State D8-ItronReadDisconStateOutMsg	D8-RdDiscStByMtrsRequest.xsl	D8-RdDiscStByMtrsResponse.xsl
Outbound Message for ReadDisconStateMeter Cmd D8-ItronRdDiscStByMtRsltOutMsg	D8-GetRdDiscStByMtrsRsReq.xsl	D8-GetRdDiscStByMtrsRsResp.xsl

Landis+Gyr XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Landis+Gyr adapter.

Outbound Message Business Object	Message XSL	Response XSL
LG - Add Meter To Inventory OB Message D3-AddMeterToInventoryOBMsg	D3-AddMtrToInvRequest.xsl	D3-AddMtrToInvResponse.xsl
LG - Meter Remove Notification OB Message D3-MeterRemoveNotifOBMsg	D3-MtrRemoveNotfRequest.xsl	D3-MtrRemoveNotfResponse.xsl
LG - Meter Retire Notification OB Message D3-MeterRetireNotifOBMsg	D3-MtrRetireNotfRequest.xsl	D3-MtrRetireNotfResponse.xsl
LG -Meter Exchange Notification OB Message D3-MeterExchNotificatnOBMsg	D3-MtrExcNotifRequest.xsl	D3-MtrExcNotifResponse.xsl
L+G Initiate Connect Disconnect OB Message D3-InitConnectDisconnectOBMsg	D3-InitiateConnDisconReq.xsl	D3-InitiateConnDisconResp.xsl

Copyright © 2026, Oracle and/or its affiliates.

Outbound Message Business Object	Message XSL	Response XSL
LG - Schedule Demand Reset OB Message D3-ScheduleDemandResetOBMsg	D3-ScheduleDemandRstReq.xsl	D3-ScheduleDemandRstResp.xsl
L+G MultiSpeak InitiateMeterReadByMeterNumber OB Message D3-InitMtrReadByMtrNumOBMsg	D3-InitMtrRdMtrNumRequest.xsl	D3-InitMtrRdMtrNumResponse.xsl
CIM AMIRequest GetMeterReads OB Message D3-CIMAMIRequestGetMtrRdOBMsg	D3-CIMGetMtrReadsRequest.xsl	D3-CIMGetMtrReadsResponse.xsl
L+G CIM 2.0 - GetMeterReads OB Message D3-CIMGetMtrRdOBMsg	D3-CIM2GetMtrReadsRequest.xsl	D3-CIM2GetMtrReadsResponse.xsl

Sensus XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Sensus adapter.

Outbound Message Business Object	Message XSL	Response XSL
Sensus - Meter Add Notification OB Message D6-MeterAddNotificationOBMsg	D6-MtrAddRemoveNotifReq.xsl	D6-MeterAddNotifResponse.xsl
Sensus -Meter Remove Notification OB Msg (D6-MeterRemoveNotificatnOBMsg)	D6-MtrAddRemoveNotifReq.xsl	D6-MeterAddNotifResponse.xsl
Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg)	D6-InitiateConnDisconReq.xsl	D6-InitiateConnDisconResp.xsl
Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg)	D6-InitiateConnDisconReq.xsl	D6-InitiateConnDisconResp.xsl
Sensus MultiSpeak InitiateMeterReadByMeterNumber OB Message (D6-InitMtrRdMtrIdOBMsg)	D6-InitMtrRdMtrIdRequest.xsl	D6-InitMtrRdMtrIdResponse.xsl
Sensus InitiateOutageDetection OB Message (D6-InitOutageDetectionOBMsg)	D6-InitOutDetectRequest.xsl	D6-InitOutDetectResponse.xsl

Copyright © 2026, Oracle and/or its affiliates.

Silver Spring Networks XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Silver Spring Networks adapter.

Outbound Message Business Object	Message XSL	Response XSL
SSN - Find Device Outbound Message D7-FindDeviceOutboundMsg	D7-FindDeviceRequest.xsl	D7-FindDeviceResponse.xsl
SSN - Get Job Status Outbound Message D7-GetJobStatusOutboundMsgType	D7-GetJobStatusRequest.xsl	D7-GetJobStatusResponse.xsl
SSN - Replace Location (D7-ReplaceLocationDR)	D7-ReplaceLocationRequest.xsl	D7-ReplaceLocationResponse.xsl
SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR)	D7-RepDvcAtLocRequest.xsl	D7-RepDvcAtLocResponse.xsl
SSN - Connect or Disconnect (D7-ConnectDisconnectDR)	D7-AddRemProvJobRequest.xsl	D7-AddRemProvJobResponse.xsl
SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR)	D7-GetRemProvRsByJobIDReq.xsl	D7-GetRemProvRsByJobIDResp.xsl
SSN - Add Meter Read Job Scalar OB Msg D7-AddMeterReadJobScalarOBMsg	D7-AddMtrRdJobScRequest.xsl	D7-AddMtrRdJobScResponse.xsl
SSN - GetMeterReadResults(Scalar) By JobId D7-GetSclrMtrRdRstByJobIdOBMsg	D7-GetMtrRdRsByJobIDReq.xsl	D7-GetMtrRdRsByJobIDResp.xsl
SSN - AddMeterReadJob Interval OB Msg D7-GetIntMtrRdRsltByJobIdOBMsg	D7-AddMtrRdJobIntRequest.xsl	D7-AddMtrRdJobIntResponse.xsl
SSN - GetMeterReadResults(Interval) By JobId D7-GetIMtrRdRsltJobIdOBMsgTyp	D7-GetMtrRdRsByJobIDReq.xsl	D7-GetMtrRdRsByJobIDResp.xsl
SSN - Add Ping Job (D7-AddPingJobDR)	D7-AddPingJobRequest.xsl	D7-AddPingJobResponse.xsl
SSN - Ping Results (D7-PingResultsDR)	D7-GetMtrRdRsByJobIDReq.xsl	D7-GetMtrRdRsByJobIDResp.xsl

Copyright © 2026, Oracle and/or its affiliates.

Silver Spring NetworksMultiSpeak XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Silver Spring Networks adapter for MultiSpeak commands.

Command	Outbound Message Business Object	Message XSL	Response XSL
Device Commissioning	SSN - MultiSpeak Meter Add Notification (D7-MSPMeterAddNotifyOBMsg)	D7-MSMtrAddRemoveNotifReq.xsl	D7-MSMeterAddNotifResponse.xsl
Device Decommissioning	SSN - MultiSpeak Meter Remove Notification (D7-MSPMeterRemoveNotifyOBMsg)	D7-MSMtrAddRemoveNotifReq.xsl	D7-MSMtrRemoveNotfResponse.xsl
Remote Connect	SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg)	D7-MSInitiateConnDisconReq.xsl	D7-MSInitiateConnDisconRes.xsl
Remote Disconnect	SSN - MSP Initiate Connect Disconnect D7-MSPInitConDisconOBMsg	D7-MSInitiateConnDisconReq.xsl	D7-MSInitiateConnDisconRes.xsl
On-Demand Read (MultiSpeak)	SSN - MSP Initiate Meter Read By Meter ID D7-MSPInitMtrRdMtrIdOBMsg	D7-MSInitMtrRdMtrIdRequest.xsl	D7-MSInitMtrRdMtrIdRequest.xsl
Device Status Check	SSN - MultiSpeak Initiate Outage Detection D7-MSPInitOutageDetectionOBMsg	D7-MSInitOutDetectRequest.xsl	D7-MSInitOutDetectResponse.xsl

Configuring Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record. The tables below list the details appropriate to configuring processing methods for each command.

This table includes the following:

- **Command**: The command you wish to configure. When a command is initiated for a device, the system creates a command activity.
- **Processing Role**: The Processing Role appropriate for the command to be configured.

Copyright © 2026, Oracle and/or its affiliates.

- **Business Object**: The outbound communication business object to be created by the command activity. Note that most commands have separate business objects for the initial command request and the result request.
- **Outbound Message Business Object** The business object that should be used when creating outbound message types for each command. Note that most commands use separate business objects for the initial outbound message type and the result outbound message type.

Itron OpenWay Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Itron OpenWay head end system.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	Itron - Add Meter Definition (D8-AddMeterDefinitionsDR)	Outbound Message for Itron Commission D8-ItronCommissionOutboundMsg
Device Decommissioning	Device Deregistration	Itron - Deregister Meter (D8-DeregisterMeterDR)	Outbound Message for Itron Decommission D8-ItronDecommissionOutbndMsg
Remote Connect	Remote Connect	Itron - Reconnect Meter (Remote Connect) D8-ReconnectMeterDR)	Outbound Message for Itron connect command D8-ItronRemoteConnectOutbndMsg
	Get Remote Connect Result	Itron - Reconnect Meter Result (D8-ReconnectMeterResult)	Outbound Message for Itron connect result command (D8-ItronRemoteCntRsltOutMsg)
Remote Disconnect	Remote Disconnect	Itron - Disconnect Meter (D8-DisconnectMeterDR)	Outbound Message for Itron disconnect command D8-ItronRemoteDisconnectOutMsg
	Get Remote Disconnect Result	Itron - Disconnect Meter Result (D8-DisconnectMeterResult)	Outbound Message for Itron disconnect result command (D8-ItronRemoteDcntRsltOutMsg)
On-Demand Read (Scalar)	On-Demand Read (Scalar)	Itron - Contingency Read (Scalar) (D8-ReadScalarDR)	Outbound Message for Itron Contingency Read command D8-ItronOnDemandReadOutbndMsg
	Get On-Demand Read Result	Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR)	O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg)
On-Demand Read (Interval)	On-Demand Read (Interval)	Itron - Contingency Read (Interval) (D8-ReadIntervalDR)	Outbound Message for Itron Contingency Read command (D8-ItronOnDemandReadOutbndMsg)
	Get On-Demand Read Result	Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR)	O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
Scheduled Read (Scalar)	Send Schedule Read Request	Itron - Interrogate by Group (Scalar) (D8-InterrogateByGroupDR)	Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg)
	Get Schedule Read Result	Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR)	Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB)
Scheduled Read (Interval)	Send Schedule Read Request	Itron - Interrogate by Group (Interval) (D8-InterrogateByGroupDR)	Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg)
	Get Schedule Read Result	Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR)	Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB)
Device Status Check	Device Status Check	Itron - Ping by Endpoints (Status Check) (D8-PingByEndpointsDR)	Outbound Message for Itron Device status check (D8-ItronDvcStatusChkOutbndMsg)
	Get Device Status Check Result	Itron - Ping By Endpoints Result (D8-PingByEndpointsResultDR)	O/B Msg for Itron DeviceStatusCheck Result (D8-ItronDvcStatusChkRstlOBMsg)
Multi-Device Status Check	Multi-Device Status Check	Itron - Ping By Endpoints Multi-Device (D8-PingByEndpointsMultiDvcDR)	Outbound Message for Itron Multi device status check (D8-ItronDvcStChkMtDvcOutbndMsg)
	Get Multi-Device Status Check Result	Itron – Ping By Endpoints Multi Dvc Result (D8-PingByEndpointsMDResultDR)	O/B Message for Itron MultiDeviceStatusCheck Result (D8-ItronDvcStChkMtDvcRsltOBMsg)
Load Check	Load Check	Itron - Detect Load Side Voltage (D8-DetectLoadSideVoltgeByMtrDR)	Outbound Message for Itron Detect Load Side Voltage (D8-ItronDetectLoadSideVolOBMsg)
	Get Load Check Results	Itron – Detect Load Side Voltage Result (D8-DetLoadSideVoltageMtrRsltDR)	Outbound Message for LoadCheckResult Cmd (D8-ItronDetLSVoltMtrRsltOutMsg)

Landis+Gyr Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Landis+Gyr head end system.

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	L+G Add Meter to Inventory(MultiSpeak) (D3-AddMeterToInventryMultiSpDR)	LG - Add Meter To Inventory OB Message D3-AddMeterToInventoryOBMsg
Device Decommissioning	Device Removal	L+G Meter Remove Notification (MultiSpeak) (D3-MtrRmvNotifMultiSpeakDR)	LG - Meter Remove Notification OB Message (D3-MeterRemoveNotifOBMsg)
Meter Retire	Device Deregistration	L+G Meter Retire Notification (D3-MeterRetireNotificationDR)	LG - Meter Retire Notification OB Message (D3-MeterRetireNotifOBMsg)
Meter Exchange	Device Installation	L+G Meter Exchange Notification (D3-MeterExNotificatnMultiSpDR)	LG -Meter Exchange Notification OB Message (D3-MeterExchNotificatnOBMsg)
Remote Connect	Remote Connect	L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR)	L+G Initiate Connect Disconnect OB Message (D3-InitConnectDisconnectOBMsg)
Remote Disconnect	Remote Disconnect	L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR)	L+G Initiate Connect Disconnect OB Message (D3-InitConnectDisconnectOBMsg)
Demand Reset	Demand Reset	L+G Schedule Demand Reset (Multispeak) (D3-ScheduleDemandResetDR)	LG - Schedule Demand Reset OB Message (D3-ScheduleDemandResetOBMsg)
On-Demand Read (MultiSpeak)	On-Demand Read (Scalar)	L+G MultiSpeak Initiate Meter Read By Meter Num Direct Route (D3-InitiateMRByMtrNbrDR)	L+G MultiSpeak InitiateMeterReadByMeterNumber OB Message (D3-InitMtrReadByMtrNumOBMsg)
On-Demand Read (Interval) [CIM]	On-Demand Read (Interval)	CIM Meter On Demand Read Direct Route (Interval) (D3-CIMGetLPDataDR)	CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg)
On-Demand Read (Scalar) [CIM]	On-Demand Read (Scalar)	CIM Meter On Demand Read Direct Route (Scalar) (D3-CIMMeterOnDemandReadDR)	CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg)
Device Status Check [CIM]	Device Status Check	CIM Ping Direct Route (D3-CIMPingDR)	CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg)
On-Demand Read (Interval) [CIM 2.0]	On-Demand Read (Interval)	L+G CIM 2.0 On-Demand Read DR (Interval) (D3-CIMGetMtrReadIntervalDR)	L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
On-Demand Read (Scalar) [CIM 2.0]	On-Demand Read (Scalar)	L+G CIM 2.0 - On-Demand Read DR (Scalar) (D3-CIMGetMtrReadScalarDR)	L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)
Device Status Check [CIM 2.0]	Device Status Check	L+G CIM 2.0 - Ping DR (D3-CIM2PingDR)	L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg)

Sensus Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Sensus head end system.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	Sensus -Meter Add Notification Direct Route (D6-MeterAddNotificationDR)	Sensus - Meter Add Notification OB Message D6-MeterAddNotificationOBMsg
Device Decommissioning	Device Removal	Sensus -Meter Remove Notification Direct Route (D6-MeterRemoveNotificationDR)	Sensus -Meter Remove Notification OB Msg (D6-MeterRemoveNotificatnOBMsg)
Remote Connect	Remote Connect	Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR)	Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg)
Remote Disconnect	Remote Disconnect	Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR)	Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg)
On-Demand Read (MultiSpeak)	On-Demand Read (Scalar)	Sensus - Initiate Meter Read By Meter ID (D6-InitiateMeterByMeterIdDR)	Sensus MultiSpeak InitiateMeterReadByMeterNumber OB Message (D6-InitMtrRdMtrIdOBMsg)
Device Status Check	Device Status Check	Sensus Initiate Outage Detection DR (D6-InitiateOutageDetectionDR)	Sensus InitiateOutageDetection OB Message (D6-InitOutageDetectionOBMsg)

Silver Spring Networks Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Silver Spring Networks head end system.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	SSN - Replace Location (D7-ReplaceLocationDR)	SSN - Replace Location (Comm) OB Message (D7-ReplaceLocationOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Decommissioning	Device Removal	SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR)	SSN - Replace Device At Loc O/B Message: (D7-ReplaceDeviceAtLocOBMsg)
Remote Connect	Remote Connect	SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP)	SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg)
	Get Remote Connect Result	SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR)	SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg)
Remote Disconnect	Remote Disconnect	SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP)	SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg)
	Get Remote Disconnect Result	SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR)	SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg)
On-Demand Read (Scalar)	On-Demand Read (Scalar)	SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP)	SSN - Add Meter Read Job Scalar OB Msg Type D7-AddMeterReadJobScalarOBMsg
	Get On-Demand Read Result (Scalar)	SSN - Meter Read Results (Scalar) (D7-MeterReadResultsScalar)	SSN - GetMeterReadResults(Scalar) By JobId (D7-GetSMtrRdRsltJobIdOBMsgTyp)
On-Demand Read (Interval)	On-Demand Read (Interval)	SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP)	SSN - AddMeterReadJob Interval OB Msg Type (D7-AddMtrReadJobIntervalOBMsg)
	Get On-Demand Read Result (Interval)	SSN - Meter Read Results (Interval) (D7-MeterReadResultsInterval)	SSN - GetMeterReadResults(Interval) By JobId (D7-GetIMtrRdRsltJobIdOBMsgTyp)
Device Status Check	Device Status Check	SSN - Add Ping Job - Polling (D7-AddPingJobDRP)	SSN - Add Ping Job OB Message (D7-AddPingJobOBMsg)
	Get Device Status Check Result	SSN - Ping Results (D7-PingResultsDR)	SSN - Get Ping Results O/B Message (D7-GetPingResultsOBMsg)
Get Job Status	Get Job Status	SSN - Get Job Status for Polling (D7-GetJobStatusDRP)	SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg)

Silver Spring Networks MultiSpeak Processing Methods

This table lists the details appropriate to configuring processing methods for each MultiSpeak command supported for the Silver Spring Networks head end system.

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	SSN - MultiSpeak Meter Add Notification DR (D7-MSPMeterAddNotificationDR)	SSN - MultiSpeak Meter Add Notification (D7-MSPMeterAddNotifyOBMsg)
Device Decommissioning	Device Removal	SSN - MSP Meter Remove Notification DR (D7-MSPMeterRemNotifyDR)	SSN - MultiSpeak Meter Remove Notification (D7-MSPMeterRemoveNotifyOBMsg)
Remote Connect	Remote Connect	SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR)	SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg)
Remote Disconnect	Remote Disconnect	SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR)	SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg)
On-Demand Read (MultiSpeak)	On-Demand Read (Scalar)	SSN - MSP Initiate Read By Meter ID DR (D7-MSPInitMtrByMtrIdDR)	SSN - MSP Initiate Meter Read By Meter ID (D7-MSPInitMtrRdMtrIdOBMsg)
Device Status Check	Device Status Check	SSN - MSP Initiate Outage Detection DR (D7-MSPInitOutageDetectDR)	SSN - MultiSpeak Initiate Outage Detection (D7-MSPInitOutageDetectionOBMsg)

Receiving Inbound Communications

Receiving inbound communications from a head end system involves configuring the following:

- **Inbound Web Services**: These services are used to receive inbound communications from external systems. See Inbound Web Services in the *Administrative User Guide* for more information.
- **Master Configurations**: Defines configuration settings for individual adapters.
- Inbound Web Services

Inbound Web Services

Copyright © 2026, Oracle and/or its affiliates.

Inbound web services are used to receive inbound communications from external systems. The tables below list the inbound web services provided with each Smart Grid Gateway adapter.

Itron OpenWay Inbound Web Services

The table below lists the inbound web service provided with the Itron OpenWay adapter.

Name	Description
D8-RequestStatusChangedService	Itron RequestStatusChangedService: Use to receive "status changed" messages from the Itron OpenWay head end system. This service creates instances of the "Itron – StatusChanged" (D8-StatusChanged) inbound communication business object for each message.
D8-DataSubscriberService	Itron DataSubscriberService: Used to receive usage and device event payloads from the Itron OpenWay head end system. This service invokes the "Itron – DataArrived Request" (D8_DATAARR) service script, which invokes the ItronDataArrivedStreamProcessor business component to process the usage or device event payload based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup.
D8-EventSubscriberService	Itron EventSubscriberService: Used to receive device event payloads from the Itron OpenWay head end system. This service invokes the "Itron – EventsArrived Request" (D8_EVNTSARR) service script, which invokes the ItronEventsArrivedStreamProcessor business component to process the device event payload based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup.
D8-ExceptionSubscriberService	Itron ExceptionSubscriberService: Used to receive exceptions from the Itron OpenWay head end system. This service invokes the "Itron – ExceptionsArrived Request" (D8_EXCPARR) service script, which invokes the ItronExceptionsArrivedStreamProcessor business component to process the exception based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup.

Landis+Gyr Inbound Web Services

The table below lists the inbound web services provided with the Landis+Gyr adapter.

Name	Schema Name	Request / Response XSL
CD State Changed Notification (Callback) (D3-CB_CD)	D3-ConnectDisconStateChgNtfDR	D3-ConDiscnStateChgNtfReq.xsl D3-ConDiscnStateChgNtfResp.xsl

Copyright © 2026, Oracle and/or its affiliates.

Name	Schema Name	Request / Response XSL
Reading Changed Notification (Callback) (D3-CB_MR)	D3-ReadingChgNotificationDR	D3-ReadingChgNotifRequest.xsl D3-ReadingChgNotifResponse.xsl
CIM Service Callback (D3-CIMCallback)	D3-CIMAsyncReplyMeterReadingDR	D3-CIMMeterReadingRequest.xsl D3-CIMMeterReadingResponse.xsl
Schedule Demand Response (Multispeak) (D3-ScheduleDemandResetResponse)	D3-ReadingChgNotificationDR	D3-ReadingChgNotifRequest.xsl D3-ReadingChgNotifResponse.xsl
L+G CIM 2.0 - Meter Read Response Callback (D3-CIM2ODRResponse)	D3CIM2MDIn	D3-CIMMtrReadingsRequest.xsl D3-CIMMtrReadingsRequest.xsl

Sensus Inbound Web Services

The table below lists the inbound web services provided with the Sensus adapter.

Name	Schema Name	Request / Response XSL
Reading Changed Notification (Callback) (D6-CB_MR)	D6-ReadingChgNotificationDR	D6-ReadingChgNotifRequest.xsl D6-ReadingChgNotifResponse.xsl
Sensus MultiSpeak CB_Server Interface (D6-CB_Server)	D6-ConnectDisconStateChgNtfDR	D6-ConDiscnStateChgNtfReq.xsl D6-ConDiscnStateChgNtfResp.xsl
	D6-ReadingChgNotificationDR	D6-ReadingChgNotifRequest.xsl D6-ReadingChgNotifResponse.xsl
Sensus MultiSpeak OA_Server Interface (D6-OA_Server)	D6-ODEventNotificationDR	D6-ODEventNotifRequest.xsl D6-ODEventNotifResponse.xsl
Unsolicited Event Response D6-UnsolicitedEventNotification	D6-UnsolicitedEvtNotification	

Silver Spring Networks MultiSpeak Inbound Web Services

The table below lists the inbound web services provided with the Silver Spring Networks adapter for use with MultiSpeak commands.

Copyright © 2026, Oracle and/or its affiliates.

Name	Operation	Schema Name	Request / Response XSL
SSN - MultiSpeak CB_Server Interface (D7-MS_CB_Server)	CDStatesChangedNotification	SSN - MultiSpeak CD State Change Notification Inbound Request (D7MSCDSCNIR)	D7-MSConDiscnStaChgNtfReq.xsl D7-MSConDiscnStaChgNtfRes.xsl
	ReadingChangedNotification	SSN - MultiSpeak Reading Change Notification Inbound Request (D7MSReChNtIR)	D7-MSReadingChgNotiRequest.xsl D7-MSReadChgNotiResponse.xsl
SSN - MultiSpeak OA_Server Interface (D7-MS_OA_Server)	ODEventNotification	SSN - MultiSpeak OD Event Notification Inbound Request (D7MSODEvNtIR)	D7-MSODEventNotifRequest.xsl D7-MSODEventNotifResponse.xsl

Adapter Development Kit - MultiSpeak Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process smart meter commands in the MultiSpeak format.

These commands can be used to support smart meter commands for meters for unsupported head end systems that use the MultiSpeak 4.1 protocol.

- Adapter Development Kit MultiSpeak Command Flows
- Adapter Development Kit MultiSpeak Command Extendable Lookups
- Sending Adapter Development Kit MultiSpeak Command Outbound Communications
- Receiving Adapter Development Kit MultiSpeak Command Inbound Communications

Adapter Development Kit MultiSpeak Command Flows

The table below lists the communication flows used with each Silver Spring Networks MultiSpeak command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Commissioning (MultiSpeak)	ADK - MultiSpeak Meter Add Notification DR (DG-MSPMeterAddNotificationDR)		Device Commissioning Completion Event
Device Decommissioning (MultiSpeak)	ADK - MSP Meter Remove Notification DR (DG-MSPMeterRemNotifyDR)		Device Decommissioning Completion Event
Remote Connect (MultiSpeak)	ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR)	ADK - MSP Conn/ Discon State Change DR (DG-MSPConDisconStaChgNtfDR)	Connect Device Completion Event
Remote Disconnect (MultiSpeak)	ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR)	ADK - MSP Conn/ Discon State Change DR (DG-MSPConDisconStaChgNtfDR)	Disconnect Device Completion Event
On-Demand Read (MultiSpeak)	ADK - MSP Initiate Read By Meter ID DR (DG-MSPInitMtrByMtrIdDR)	ADK - MSP Reading Changed Notification DR (DG-MSPReadingChgNotifyDR)	Create IMD Completion Event
Device Status Check (MultiSpeak)	ADK - MSP Initiate Outage Detection DR (DG-MSPInitOutageDetectDR)	ADK - MSP OD Event Notification DR (DG-MSPODEventNotificationDR)	

Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match (configured UOM/TOU/SQI combination are processed. All others are filtered o (ut.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the MSP Create IMD Completion Event DR (D1-MSCCEDR) algorithm on the Create Completion Events status of the MultiSpeak Reading Changed Notification DR (D1-MSPReadingChgNotifyDR) business object.

The MSP Check if Device has any Filtering errors DR (D1-MSVADVFDR) algorithm on the Validate status of the D1-MSPReadingChgNotifyDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D1-

Copyright © 2026, Oracle and/or its affiliates.

MSCCEDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Adapter Development Kit MultiSpeak Command Extendable Lookups

Processing of MultiSpeak commands requires configuration of values for a number of extendable lookups. This topic lists the extendable lookups used with processing MultiSpeak commands using the Adapter Development Kit.

- MultiSpeak Connect/Disconnect Reason Code (D1-MSPCDReasonCodeLookup)
- MultiSpeak Header Build Descriptor (D1-MSPBuildDescriptorLookup)
- MultiSpeak Header Coordinate System Units (D1-MSPCSUnitsLookup)
- MultiSpeak Header Message Context (D1-MSPMessageContextLookup)
- MultiSpeak Load Action Code (D1-MSPLoadActionCodeLookup)
- MultiSpeak Service Type (D1-MSPServiceTypeLookup)
- MultiSpeak Service Type MappingD1-MSPServiceTypeMappingLookup)
- MultiSpeak Time Units (D1-MSPTimeUnits)
- MultiSpeak Verb Action Code (D1-MSPActionCodeLookup)

Sending Adapter Development Kit MultiSpeak Command Outbound Communications

This topic describes the configuration required to process MultiSpeak outbound communications.

Adapter Development Kit MultiSpeak Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used.

This table lists the outbound message types used with Adapter Development Kit MultiSpeak commands, along with the business objects used with each.

Command	Outbound Message Type / Business Object
Device Commissioning	ADK - MultiSpeak Notify Meter Add (DG-MSPMANOMT) ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg)
Device Decommissioning	ADK - MultiSpeak Notify Meter Remove (DG-MSPMRNOMT) ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg)
Remote Connect	ADK - MultiSpeak Initiate Connect Disconnect by Meter Number (DG-MSPICBOMT) ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)
Remote Disconnect	ADK - MultiSpeak Initiate Connect Disconnect by Meter Number (DG-MSPICBOMT) ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Type / Business Object
On-Demand Read (MultiSpeak)	ADK- MultiSpeak Initiate Meter Read by Meter Id (DG-MSPMRBMIO) ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg)
Device Status Check	ADK - MultiSpeak Initiate Outage Detection (DG-MSPIODOMT) ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg)

Adapter Development Kit MultiSpeak Message Senders

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

Note: When creating message senders as part of the command communication process used with native implementations, the Message Class MUST be set to SOAPSNDR (Sender for real-time HTTP / SOAP messages).

See **Message Senders** in the *Administrative User Guide* for more information about creating message senders.

Adapter Development Kit MultiSpeak External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

Adapter Development Kit MultiSpeak XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Adapter Development Kit for MultiSpeak commands.

Command	Outbound Message Business Object	Message XSL	Response XSL
Device Commissioning	ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg)	DG-MSMtrAddRemoveNotifReq.xsl	DG-MSMeterAddNotifResponse.xsl
Device Decommissioning	ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg)	DG-MSMtrAddRemoveNotifReq.xsl	DG-MSMtrRemoveNotfResponse.xsl
Remote Connect	ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)	DG-MSInitiateConnDisconReq.xsl	DG-MSInitiateConDisconResp.xsl
Remote Disconnect	ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)	DG-MSInitiateConnDisconReq.xsl	DG-MSInitiateConDisconResp.xsl

Copyright © 2026, Oracle and/or its affiliates.

Command	Outbound Message Business Object	Message XSL	Response XSL
On-Demand Read (MultiSpeak)	ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg)	DG-MSInitMtrRdMtrIdRequest.xsl	DG-MSIniMtrRdMtrIdResponse.xsl
Device Status Check	ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg)	DG-MSInitOutDetectRequest.xsl	DG-MSInitOutDetectResponse.xsl

See **External Systems** in the *Administrative User Guide* for more information about creating exteral systems.

Adapter Development Kit MultiSpeak Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record.

This table lists the details appropriate to configuring processing methods for each MultiSpeak command used with the Adapter Development Kit.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Commissioning	Device Registration	ADK - MultiSpeak Meter Add Notification DR (DG-MSPMeterAddNotificationDR)	ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg)
Device Decommissioning	Device Removal	ADK - MSP Meter Remove Notification DR (DG-MSPMeterRemNotifyDR)	ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg)
Remote Connect	Remote Connect	ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR)	ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)
Remote Disconnect	Remote Disconnect	ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR)	ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg)
On-Demand Read (MultiSpeak)	On-Demand Read (Scalar)	ADK - MSP Initiate Read By Meter ID DR (DG-MSPInitMtrByMtrIdDR)	ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg)

Copyright © 2026, Oracle and/or its affiliates.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Status Check	Device Status Check	ADK - MSP Initiate Outage Detection DR (DG-MSPInitOutageDetectDR)	ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg)

See **Configuring Processing Methods**for more information.

Receiving Adapter Development Kit MultiSpeak Command Inbound Communications

Inbound web services are used to receive inbound communications from external systems.

The table below lists the inbound web services provided with the Adapter Development Kit for use with MultiSpeak commands.

Name	Operation	Schema Name	Request / Response XSL
ADK MultiSpeak CB_Server Interface (DG-CB_Server)	CDStatesChangedNotification	MultiSpeak Connect Disconnect State Change Notification Inbound Request (DGMSCDSCNIR)	DG-MSConDiscStateChgNtfReq.xsl DG-MSConDisStateChgNtfResp.xsl
	ReadingChangedNotification	MultiSpeak Reading Change Notification Inbound Request (DGMSReChNtIR)	DG-MSReadingChgNotRequest.xsl DG-MSReadingChgNotResponse.xsl
ADK MultiSpeak OA_Server Interface (DG-OA_Server)	ODEventNotification	MultiSpeak OD Event Notification Inbound Request (DGMSODEvNtIR)	DG-MSODEventNotifRequest.xsl DG-MSODEventNotifResponse.xsl

See **Inbound Web Services** for more information.

Adapter Development Kit - Common Information Model Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process smart meter commands in the Common Information Model (CIM) format.

- CIM Commands Introduction and Overview
- Adapter Development Kit CIM Command Flows
- CIM Command Extendable Lookups
- Sending CIM Command Outbound Communications

Copyright © 2026, Oracle and/or its affiliates.

- Receiving CIM Command Inbound Communications

CIM Commands Introduction and Overview

The Adapter Development Kit supports smart meter commands in the Common Information Model (CIM) format. The CIM formats and standards supported by the Adapter Development Kit are based on the International Standard (IEC 61968-9, Edition 2.0), available **here**. Reference to specific data elements from the CIM command messages used in this section are based on this document. Refer to this document for detailed information about the CIM commands.

Adapter Development Kit CIM Command Flows

The table below lists the communication flows used with Common Information Model (CIM) commands, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

Command	Outbound Communication	Inbound Communication	Completion Event
Device Status Check	ADK CIM Ping Direct Route (DG-CIMPingDR)	ADK CIM Ping Response Direct Route (DG-CIMPingResponseDR)	
On-Demand Read (Interval) [CIM]	ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR)	ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR)	Create IMD Completion Event
On-Demand Read (Scalar) [CIM]	ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR)	ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR)	Create IMD Completion Event

Configuring Command Types

The following communication business objects (from the table above) all have corresponding communication types that must be configured before communications based on those business objects can be created. The table below lists the communication type business objects used with each.

Communication Business Object	Communication Type Business Object
ADK CIM Ping Direct Route (DG-CIMPingDR)	ADK - CIM Ping DR Type (DG-CIMPingDRType)

Copyright © 2026, Oracle and/or its affiliates.

Communication Business Object	Communication Type Business Object
ADK CIM Ping Response Direct Route (DG-CIMPingResponseDR)	ADK - CIM Ping Async Response DR Type (DG-CIMPingResponseDRType)
ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR)	ADK - CIM On-Demand Read (Interval) DR Type (DG-CIMGetMtrReadIntervalType)
ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR)	ADK - CIM On-Demand Read (Scalar) DR Type (DG-CIMGetMtrReadScalarType)
ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR)	ADK - CIM AsyncReply MeterReads DR Type (DG-CIMMtrReadResponseDRType)

Refer to **Communication Types** for more information about communication types.

CIM Command Extendable Lookups

Processing of CIM commands requires configuration of values for a number of extendable lookups. This topic outlines the extendable lookups used with processing CIM commands.

Payload Processing Extendable Lookups

Processing of measurement data returned via on-demand read commands requires configuration of values for the following extendable lookups:

- **Unit of Measure Mapping**: Unit of measure mapping is based on the ReadingType string in the CIM payload and uses the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup.
- **Interval Size Mapping**: Interval size (expressed as seconds per interval or SPI) mapping is based on the measuringPeriod value extracted from the ReadingType string and uses the ADK CIM Measuring Period Lookup (DG-CIMMeasuringPeriodLookup) extendable lookup.
- **Condition Code Mapping**: Condition code mapping is based on the ReadingQualityType string in the CIM payload (if present) and uses the the ADK - Interval Sts Code to Cond Map (DG-IntStsCodeToCondMapLookup) extendable lookup.

See **CIM Payload Processing Data Mapping and Extendable Lookups** for more information about these extendable lookups.

ADK CIM Header Verb Lookup

The ADK CIM Header Verb Lookup (DG-CIMHeaderVerbLookup) extendable lookup defines values for message type verbs used in CIM messages. Refer to Annex A in the **International Standard document** (LINK) for detailed information about verbs. This extendable lookup is pre-populated with values from the International Standard document.

ADK CIM Reply Code Lookup

The ADK CIM Reply Code Lookup (DG-CIMReplyCodeLookup) extendable lookup contains the enumeration of the values in the Reply / Error / Code element in replies from the head end system. The values are defined in Annex B in the **International Standard document** (LINK). Values comprise two numbers separated by a dot. Example: ReplyCode = <ReplyCode Category> “.” <ReplyCode Index>

This extendable lookup is pre-populated with values from **Table B.10 – Common enumerations** in the International Standard document.

Copyright © 2026, Oracle and/or its affiliates.

ADK CIM Reply Result Lookup

The ADK CIM Reply Result Lookup (DG-CIMResponseReplyRsltLookup) extendable lookup contains the enumeration of the values for the Reply.Result value returned from the head end system. This extendable lookup is pre-populated with values from the International Standard document (Refer to Annex B).

ADK CIM Data Source Extendable Lookup

The ADK CIM Data Source Extendable Lookup (DG-DataSourceLookup) extendable lookup is used to map descriptions to data sources defined in the head end system. Each value defined for the CIM Data Source extendable lookup should include the following:

- **Data Source**: The CIM code for the data source
- **Description**: A description of the data source
- **Status**: The status of the lookup value (can be Active or Inactive)

This extendable lookup is pre-populated with two values : AMI and Database.

Sending CIM Command Outbound Communications

This topic describes the configuration required to process (CIM) outbound communications.

Adapter Development Kit CIM Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used.

This table lists the outbound message types used with Adapter Development Kit CIM commands, along with the business objects used with each.

Command	Outbound Message Type / Business Object
Device Status Check	ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg)
On-Demand Read (Interval)	ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK - CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg)
On-Demand Read (Scalar)	ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK - CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg)

See **Outbound Messages** in the *Administrative User Guide* for more information about creating message senders.

Adapter Development Kit CIM Message Senders

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

Note: When creating message senders as part of the command communication process used with native implementations, the Message Class MUST be set to SOAPSNDR (Sender for real-time HTTP / SOAP messages).

Copyright © 2026, Oracle and/or its affiliates.

See **Message Senders** in the *Administrative User Guide* for more information about creating message senders.

Adapter Development Kit CIM External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

Adapter Development Kit CIM XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided for use with Adapter Development Kit CIM commands.

Outbound Message Type	Outbound Message Business Object	Message XSL	Response XSL
ADK - CIM Get MeterReadings (DG-CIMGMR)	ADK - CIM Get Meter Reads(DG-CIMGetMtrRdOBMsg)	DG-CIMGetMtrReadsRequest.xsl	DG-CIMGetMtrReadsResponse.xsl

Adapter Development Kit CIM Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record.

This table lists the details appropriate to configuring processing methods for each command used with the Adapter Development Kit.

Command	Processing Role	Business Object	Outbound Message Business Object
Device Status Check	Device Status Check	ADK CIM Ping Direct Route (DG-CIMPingDR)	ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg)
On-Demand Read (Scalar)	On-Demand Read (Scalar)	ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR)	ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg)
On-Demand Read (Interval)	On-Demand Read (Interval)	ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR)	ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg)

See **Configuring Processing Methods** for more information.

Receiving CIM Command Inbound Communications

Inbound web services are used to receive inbound communications from external systems.

The table below lists the inbound web service provided for use with Adapter Development Kit CIM commands.

Copyright © 2026, Oracle and/or its affiliates.

Name	Description
DG-CIMPingResponse	This Inbound Web Service receives incoming asynchronous callback requests for Device Status Check. This web service accepts requests for the Response method defined in {http://iec.ch/TC57/2011/abstract}Generic.wsdl.
DG-CIMODRResponse	This Inbound Web Service receives incoming asynchronous callback requests for On-Demand Reads. This web service accepts requests for the Response method defined in {http://iec.ch/TC57/2011/abstract}Generic.wsdl.

See **Inbound Web Services** for more information.

Adapter Development Kit - Creating Custom Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to support custom smart meter commands, including those from non-supported head end systems.

This includes:

- An overview of the data and objects used by smart meter commands
- A description and example of synchronous commands
- A description and example of asynchronous commands

- Overview
- Synchronous Commands
- Asynchronous Commands

Overview

This section provides an overview of how smart meter commands work and the types of objects used in executing commands when implementing Smart Grid Gateway adapters using native implementations and with Oracle Utilities cloud services.

As noted inDevice Communication Overview, command processing works slightly differently than it does in middleware implementations. That section provides an introduction to command processing in Smart Grid Gateway adapters, and the **Native Implementation Command Processing** section provides a step-by-step description of the various data and objects created as part of command processing.

The remainder of this section provides additional details regarding the three primary types of data and objects used with smart meter command processing:

- Command Activities
- Outbound Communications
- Inbound Communications

- Command Activities
- Outbound Communications
- Inbound Communications

Copyright © 2026, Oracle and/or its affiliates.

Command Activities

Smart meter commands are managed by activities. Activities orchestrate one or more communications and execute any completion events associated with the command.

Key Algorithms

This section outlines some of the important algorithms executed as part of the command activity's lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.

Lifecycle State	Algorithms
Validate	The Validate state is an opportunity to check whether the command should execute. • Check Device Command Eligibility: Checks whether the command request is eligible for the device by calling the Determine Eligible Commands Business Service (D1-DetermineEligibleCommands). The command request is considered eligible if it exists in the eligible commands list returned by the business service. • Validate Head End's Capability to Perform Activity: This algorithm checks to ensure that the head end system has the capability to support the command requested. It accomplishes this by invoking the Determine Service Providers and Methods Business Service. • Check for Existing Active Request: This algorithm looks for non-Final Activities with the same Device and of the same type. This prevents duplicate requests from being sent to the head end system.
Validation Error	If the Activity fails validation, it transitions to the Validation Error state. From this state the activity can be retried and create To Do entriesto notify users of the error. • Retry BO In Error: Reprocesses the BO by transitioning it to its original state. • Create To Do Entry for BO in Error: This algorithm attempts to create a To Do entry using the To Do type and role specified by the algorithm parameters. • Generic To Do Completion for BOs: This algorithm completes all To Do entries with Drill Keys = Current Business Object's prime keys except for To Do Types with Characteristic Type and Value as specified by algorithm parameters. This prevents the existence of multiple To Do entries for the same command and device.
Waiting For Effective Date	Prior to sending the request to the head end system, the activity enters this state. • Wait Time Out - Transition to Rejection: This algorithm rejects the business object if the expiration date time has been reached. • Wait for Effective Date: When the required date is reached, this algorithm transitions the business object to the next configured state. • Send Response to External Requester: This algorithm sends an acknowledging received response outbound message to the requester if the business object is configured to do so.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Connection Ready	The Connection Ready state shows that the activity can be executed and creates the proper outbound communication business object. This state can be re-entered when there are multiple outbound communications that are required by the head end system's API. • Create Outbound Communication: This algorithm creates an outbound communication instance if the value of the completion flag specified by an algorithm parameter is false. • Create Results Outbound based on Completion Flag: This algorithm triggers a secondary communication to the head end system based on the value of an algorithm parameter. Typically, this indicates the successful completion of an initial outbound communication.
Communication In Progress	This lifecycle state indicates that an outbound communication is in-progress. • Wait Time Out - Transition to Exception: If the business object has been in the In Progress state too long, this algorithm transitions the business object to an exception state. • Check Children Communications: If all of the outbound communication business object instances related to this activity have completed, the algorithm transitions the business object to the next state.
Communication Error	An error-handling lifecycle state to which the business object transitions when the communication encounters a problem. The algorithms are similar to those in the Validation Error state. • Wait Time Out - Transition to Rejection: If the business object has been in the In Progress state too long, this algorithm transitions the business object to a rejection state. • Retry BO In Error: This algorithm reprocesses the business object by transitioning it to its original state. • Create To Do Entry for BO in Error: This algorithm attempts to create a To Do entry using the To Do type and role specified by the algorithm parameters. • Generic To Do Completion for BOs: This algorithm completes all To Do entries with Drill Keys = Current Business Object's prime keys except for To Do Types with Characteristic Type and Value as specified by algorithm parameters. This prevents the existence of multiple To Do entries for the same command and device
Retry	The lifecycle state responsible for discarding existing outbound communications and transitioning the activity back to Connection Ready.
Execute Completion Events	This lifecycle state collates the information acquired by the communications and creates some result. For example, in a DeviceStatusCheck, the overall results are set. • Execute Completion Events: Fires the completion events associated with the activity.
Completion Event Error	This is another error-handling state similar to the Validation Error and the Communication Error states. The algorithms are the same as in those states.
Completed	This is the final state representing a successful activity. Finishing tasks are performed and external requesters are notified.
Discarded	Similarly to the Completed state, the Discarded state finishes the activity and indicates an overall failure. External requesters are notified of the failure and other finishing tasks are performed.

Copyright © 2026, Oracle and/or its affiliates.

Outbound Communications represent the business logic of sending a request and receiving a reply. Outbound communication business objects are based on the D1-COMMOUT maintenance object. The outbound communication is not the actual message, but creates the outbound message. Outbound communication business objects used with native implementations are suffixed with “-DR” for “direct route” to indicate the request go straight to the head end system.

Schema

This section provides details concerning outbound communication business object schemas.

Outbound communication schemas typically have a “sendDetail” element corresponding to the data being sent to the head end system, or more precisely, the XSL handler for the outbound message. The synchronous response is transformed by the XSL handler and placed into the “responseDetail” element.

Key Algorithms

This section outlines some of the important algorithms executed as part of the outbound communication’s lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.

Lifecycle State	Algorithms
Validate	Similar to the sctivity, the Validate state houses algorithms that will execute prior to sending the message. • Validate Communication Type: This algorithm ensures the communicationType field is populated on the message.
Validation Error	If the Validate state encounters an error, the business object transitions to this state. The algorithms are similar to those in the activity business object.
Awaiting Response	The Awaiting Response state sends an outbound message and awaits the response. • Populate Send Detail: Each outbound communication has a specific algorithm to populate all the schema elements that are required to create the outbound message. • Create Outbound Message: This algorithm creates a specific outbound message based on the Processing Role. The response is stored in the responseDetails element. • Timeout: This algorithm detects whether a communication has been waiting for an asynchronous response too long.
Retry	The lifecycle state responsible for transitioning the outbound communication back to Awaiting Response.
Response Error	The business object enters the Response Error state when an error has occurred in sending the request. It contains algorithms for creating To Do entries and for retrying similar to the activity’s Communication Error state.
Completed	This is the final state representing a successful outbound communication. Finishing tasks such as setting flags for use by subsequent communications and notifying the parent activity of completion are performed.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Discarded	Includes an algorithm that transfers the parent activity to Failed.

Inbound Communications

Inbound communication business objects are based on the D1-COMMIN maintenance objects. Inbound Communication objects have a much less defined schema since they are driven by the needs of the head end system's API and those of the business process. As with outbound communications, the schema does not represent the exact message, but is the post-processed XSL result.

Key Algorithms

This section outlines some of the important algorithms executed as part of the inbound communication's lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.

Lifecycle State	Algorithms
Validate	Similar to the activity, the Validate state houses algorithms that will execute as the message is created. • Validate Communication Type: This algorithm ensures the communicationType field is populated on the message. • Find Parent Outbound Communication for Async Inbound: This algorithm is responsible for linking the incoming message with the original outbound communication using, for example, a transaction ID. This algorithm is what causes an inbound communication to be linked with a larger process.
Validation Error	If the Validate state encounters an error, the business object transitions to this state. The algorithms are similar to those in the activity business object.
Evaluate Response	The Evaluate Response state processes the inbound communication in the context of an "asynchronous response." The algorithms should perform any required update tasks such as setting flags on the parent outbound communication.
Completed	This successful end state for inbound communications should transition the parent outbound communication to the next state. This typically completes the outbound communication and may trigger the activity to create a subsequent outbound communication.
Discarded	The parent outbound communication should transition to a failure state. This will also place the activity into the Communication Error state where it will await corrections or a Retry attempt.

Synchronous Commands

Synchronous commands, also known as "request-reply" commands, are single HTTP transactions with an initial request followed by a response. The request will contain whatever information is necessary to complete the command, such as meter identifiers or dates. The reply will contain any results generated by the command. It could be a simple

Copyright © 2026, Oracle and/or its affiliates.

acknowledgement or a complex data structure. In general, it will return quickly as HTTP transactions will fail if the response does not arrive quickly enough.

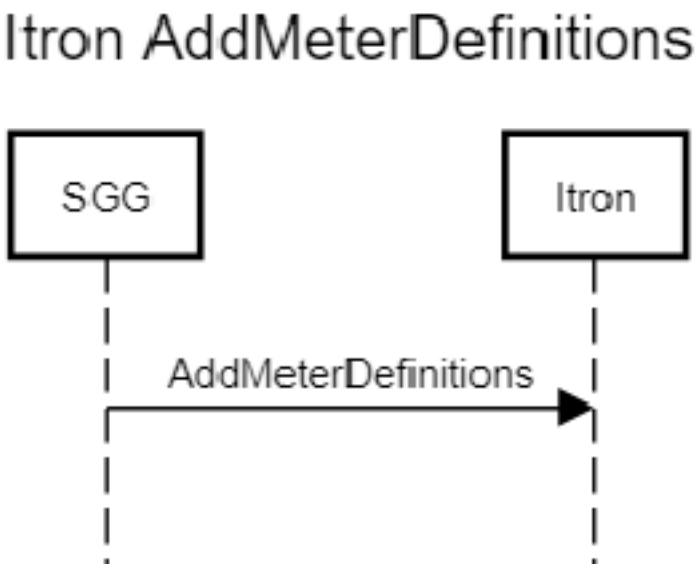
- Example: Itron Commissioning Process
- Itron Commissioning XSL Transformations

Example: Itron Commissioning Process

This section provides an example of a synchronous command, the commission device command supported by the Itron OpenWay adapter.

The Itron OpenWay 3.9 command for registering a meter (commissioning a device) in the system is AddMeterDefinitions in the `www.itron.com.ami.2008.10.provisioning.wsdl` definition. The Smart Grid Gateway Adapter for Itron OpenWay typically sends a meter serial number and some other configuration data. The response indicates whether or not an error was encountered.

The image below illustrates the process flow used by this command.



Below is a summary of the important features of the Smart Grid Gateway implementation of this command. Many of the concepts can be applied when creating a custom synchronous command.

Command Activity — Device Commissioning

Business Object: D1-DeviceCommissioning

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Device Commissioning command activity.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Validate	The important features here are not the specific algorithms, but rather the ability to perform validation logic at a process level. • D1-VALDVCNAC – Validate Device Not Already Commissioned: This algorithm checks whether the device is already commissioned • D1-VHCPCOMMS – Validate Head-End's Capability to Commission Device: This algorithm checks to ensure that the head-end system has the capability to support the command requested. Some commands may not be supported by every head end system. • D1-CACTCOMM – Check for Existing Active Commissioning Command Request: This algorithm prevents multiple, simultaneous commands from being issued by SGG. • D1-CHKFDCOMM – Check for Concurrent Decommissioning Command Request: This algorithm prevents a commissioning request from executing when there is a current decommissioning request.
Commission Ready	Several algorithms in this state are designed to create outbound communications to the head end system based on various conditions. • D1-AMIOBCOMM – AMI Device Identifier Outbound Communication Creation: This algorithm creates an outbound message, a communication to the head end system.
Communication In Progress	• D1-CHKCHILD – Check Children Communications: This algorithm checks all of the outbound communications related to the activity and, if complete, will transition the activity.
Execute Completion Events	• D1-EXCMPEVTS – Execute Completion Events: This algorithm provides a mechanism for running completion events upon successful completion of the commissioning operation.

Outbound Communication — Itron - Add Meter Definition (Commission)

Business Object: D8-AddMeterDefinitionsDR

Schema

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-AddMeterDefinitionsDRDA data area) and the `responseDetail` element (constructed using the D8-AddMeterDefRespDRDA data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Add Meter Definition outbound communication.

Lifecycle State	Algorithms
Validate	• D1-VALCOMTP – Validate Communication Type: This algorithm ensures the communication type is populated on the communication.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Awaiting Response	• D8-PAMDSDDR – Itron - Populate Add Meter Definitions Send Detail - Direct Route: This algorithm constructs the `sendDetails` element used in the outbound request. • D8-COUTMCMDR – Create Outbound Message for Commissioning / Decommissioning Direct Route: This algorithm determines the correct outbound message to create based upon the input processing role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure).
Response Error	The business object will transition to this state if the reply indicates a failure. • D8-RAMBOE – Itron - Retry Add Meter BO in Error: This algorithm provides a means to retry the communication in the event of an error. • D8-CTDEBOEO – Itron - Create To Do Entry for BO in Error for Outbound: This algorithm creates a To Do entry in order to inform a user to take corrective action. • D1-ALLOWTPA – Set Allow Transition Parent Activity Flag To True For Synchronous Outbound Communication: This algorithm conditionally allows the parent activity to transition to complete.
Create Completion Event	• D8-CRCDCOMPE – Itron - Create Commission Device Completion Event: This algorithm creates a completion event to show the device as commissioned.
Completed	• D8-UCCFTPA – Itron - Update Commission Completion Flag: This algorithm notifies the parent command activity that the communication has completed.

Message Sender XSLs

Outbound messages created by Add Meter Definition outbound communication use the following XSL files:

- **Request XSL**: D8-CommissionRequest.xsl
- **Response XSL**: D8-CommissionResponse.xsl

Refer to Creating Message Senders for more information about creating message senders.

Itron Commissioning XSL Transformations

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports multiple versions of the Itron OpenWay API. To achieve this support, several of the request XSLs check a version number embedded in the source schema. Similarly, the responses XSLs are designed to handle different source XML and map the values to the common SGG format.

The Itron commissioning process uses the following XSL transformations.

D8-CommissionRequest.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:serArr="http://schemas.microsoft.com/2003/10/Serialization/
Arrays"
```


Copyright © 2026, Oracle and/or its affiliates.

```
        xmlns:prov390="http://www.itron.com/ami/2009/08/provisioning"
        xmlns:prov370="http://www.itron.com/ami/2008/10/provisioning">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
        <xsl:apply-templates select="./requestEnvelope/AddMeterDefinitions"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.70']">
        <xsl:apply-templates select="." mode="Itr370"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.90']">
        <xsl:apply-templates select="." mode="Itr390"/>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr370">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr370"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr390">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2009/08/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr390"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr390">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr370">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

D8-CommissionResponse.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:serArr="http://schemas.microsoft.com/2003/10/Serialization/
Arrays"
```


Copyright © 2026, Oracle and/or its affiliates.

```
        xmlns:prov390="http://www.itron.com/ami/2009/08/provisioning"
        xmlns:prov370="http://www.itron.com/ami/2008/10/provisioning">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
        <xsl:apply-templates select="./requestEnvelope/AddMeterDefinitions"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.70']">
        <xsl:apply-templates select="." mode="Itr370"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.90']">
        <xsl:apply-templates select="." mode="Itr390"/>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr370">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr370"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr390">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2009/08/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr390"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr390">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr370">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

Asynchronous Commands

An asynchronous command is actually a collection of two or more synchronous commands. This type of command is useful when more than a few seconds will elapse before results can be collected and returned. A typical use case will begin with a Smart Grid Gateway request to start a job on the head end system. The initial request will contain enough data to initiate the process, such as a meter identifier or a date range. The response usually contains an identifier

Copyright © 2026, Oracle and/or its affiliates.

for the process started on the head end system. At some later time, a request will be initiated by the head end system. The request will contain the same process identifier so it can be linked to the original request. Some head end systems use this secondary request to indicate the status of the process. Other head end systems send the results of the process in the secondary request. Typically, the response message for either type of secondary request is a functional acknowledgement.

- Example: Itron On-Demand Read Process
- Itron On-Demand Read XSL Transformations

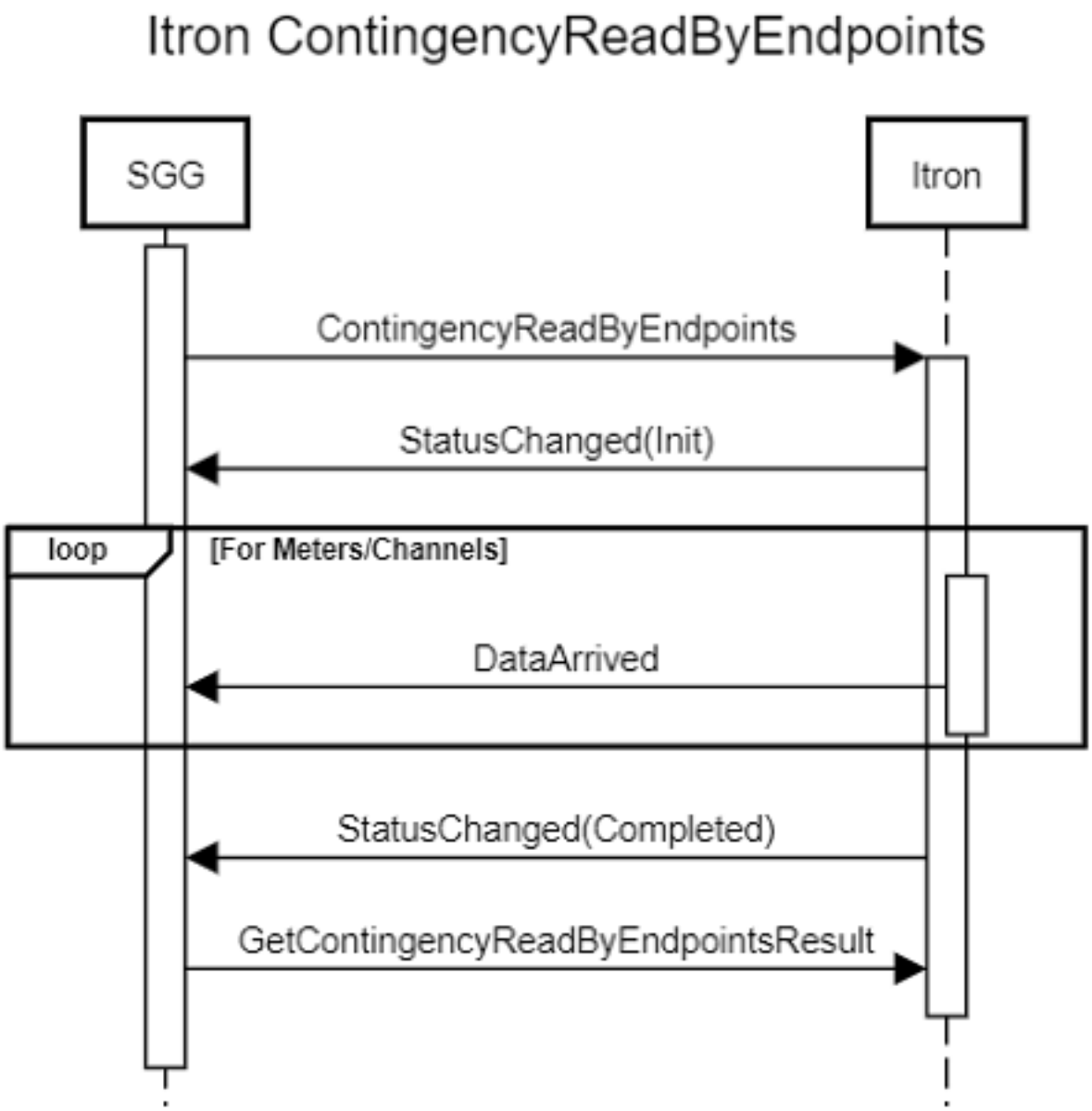
Example: Itron On-Demand Read Process

This section provides an example of an asynchronous command, the On-Demand Read command supported by the Itron OpenWay adapter.

The Itron `www.itron.com.ami.2008.10.data.wsdl` definition contains a series of commands for an On-Demand Read process. The Smart Grid Gateway adapter for Itron OpenWay initiates the process by invoking the ContingencyReadByEndpoints method. The reply contains a “request token ID” that identifies the entire flow and will be shared by subsequent communications. On the Itron side, a job for collecting usage and events is configured. To inform the caller (SGG) that the job is starting, Itron sends a StatusChanged request (defined in `www.itron.com.ami.2008.10.common.wsdl`) containing “State = Initialized”. Itron sends another StatusChanged request to notify SGG that the process of gathering the usage is underway. As the usage is collected, Itron submits one or more DataArrived requests to SGG. When Itron has finished making internal requests for usage, it sends a final StatusChanged request to SGG. That completion notification informs SGG that it should submit GetContingencyReadByEndpointsResult, the reply to which contains information about errors encountered during the process.

The image below illustrates the process flow used by this command.

Copyright © 2026, Oracle and/or its affiliates.



Below is a summary of the important features of the Smart Grid Gateway implementation of this command. Many of the concepts can be applied when creating a custom synchronous command.

Command Activity — On-Demand Read (Interval)

Business Object: D1-OnDemandReadInterval

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the On-Demand Read (Interval) command activity.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Validate	The important features here are not the specific algorithms, but rather the ability to perform validation logic at a process level. • D1-VALDEVMC – Validate that Device has appropriate Measuring Component: This algorithm ensures the device has at least one measuring component whose Measuring Component Type is registered as interval type • D1-VALIDCMD – Validate Headend's capability to perform On Demand Read (Interval): This algorithm checks whether the head end system has the capability to support an on-demand read request. If the command is supported, an outbound communication business object is returned. • D1-CHKMST – Check for existing Measurements: This algorithm checks whether complete measurements for the specified time period exist in the database already.
Connection Ready	• D1-CODRIOB – Create Outbound communication for OnDemandRead Interval: This algorithm creates an outbound communication business object based on the Processing Role and algorithm parameters. The algorithm will look up the outbound communication business object defined for the Processing Role parameter and create an instance of that business object. In the case of this business object, the parameter is D1IN and it should create an instance of the D8-ReadIntervalDR business object. Once the Outbound Communication record is created, the Activity business object moves from the Connection Ready state to the Communication In Progress state. When the outbound communication is complete, it moves the Activity to next default state, which is Connection Ready (the activity moves from Communication In Progress *back* to Connection Ready). The outbound communication will also set the “isOnDemandReadComplete” flag value in the activity business object, in this case, indicating the outbound communication for a particular processing role was created. If the condition is true, the processing moves on to next algorithm. • D1-CODRSRTOB – Create On Demand Read Results Outbound based on Completion Flag: This algorithm is the second algorithm in this state responsible for sending outbound communications. If the completion flag passed as a parameter is true, it will create the second outbound communication. In this example, it should create an instance of D8-ReadOnDemandReadResultDR to send a second request to Itron for the results of the Contingency Read operation.
Waiting for Measurement	• D1-RIINTM – Retrieve Interval Initial Measurements: This algorithm retrieves the most recent Initial Measurement Data for each interval measuring component for the period of the measurement requested.

Outbound Communication — Itron - Contingency Read (Interval)

Business Object: D8-ReadIntervalDR

Schema

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-ContingencyReadRequestDR data area) and the `responseDetail` element (constructed using the D8-ContingencyReadResponDtlDA data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Contingency Read outbound communication.

Lifecycle State	Algorithms
Validate	• D1-VALCOMTP – Validate Communication Type: This algorithm ensures the communication type is populated on the communication.
Awaiting Response	• D8-PSDMRSCDR – Itron - Populate Send Detail for ContingencyReadByEndpoints Direct Route: This algorithm populates the `sendDetail` element with the needed items for the web service input. • D8-COUTMINDR – Itron - Create Outbound Message Contingency Read (Interval): This algorithm determines the correct outbound message to create based upon the input processing role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure).
Response Error	• D8-RBORINTER – Itron - Retry Read Interval BO in Error: This algorithm provides a means to retry the communication in the event of an error.
Completed	• D1-UCFLAGTPA – Update On-Demand Read Completion Flag And Transition Parent Activity: If requirements are met, this algorithm sets the completion flag on the parent activity to “true” and transitions it to the next state.

Inbound Communication — Itron - StatusChanged

Business Object: D8-StatusChanged

This business object handles the inbound StatusUpdate requests. It contains the request token ID created by the initial request. When it receives schema values of “State = Completed” and Result of either Canceled or Success, it triggers the next command in the Activity. For On-DemandRead, the next command is GetContingencyReadByEndpointsResult.

Schema

Unlike other communications containing `sendDetail` or `responseDetail` elements, this business object contains an input element with child elements that map to the those within the StatusChanged XSD definition.

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Status Changed Read inbound communication.

Lifecycle State	Algorithms
Validate	• D8-IDNPOCAA – Find Parent Outbound Communication For Async Inbound: This algorithm finds the initial outbound communication associated with the current inbound communication using the request token id.

Copyright © 2026, Oracle and/or its affiliates.

Lifecycle State	Algorithms
Evaluate Response	• D8-EVALSTCHG – Itron - Evaluate Response for StatusChanged: This algorithm performs a task based on the values of the arriving StatusChanged request. If the Status is Completed and the Result is Success or Failure, the parent outbound communication is completed. This alerts the activity to initiate the next outbound communication.

Outbound Communication — Itron - On Demand Read Result

Business Object: D8-ReadOnDemandReadResultDR

This is the second of two outbound requests within the Itron Contingency Read API. The outbound request submits the request token ID. The response contains error information collected by the Itron server during the execution of the Contingency Read process.

Schema

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-ReadODRResultRequestDR data area) and the `responseDetail` element (constructed using the D8-ReadODRResultResponseDR data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the On-Demand Read Result outbound communication.

Lifecycle State	Algorithms
Awaiting Response	• D8-PSDODRRDR – Itron - Populate OnDemand Read Result Send Details: This algorithm populate the `sendDetail` element for the outbound message. Primarily, this consists of the request token ID. • D8-CODRROBDR – Create Outbound Message for On Demand Read Result: This algorithm determines the correct outbound message to create based upon the input Processing Role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure).
Evaluate Response	• D8-EVODRRSLT – Create Outbound Message for On Demand Read Result: This algorithm evaluates the response from Itron and determines whether the command was successful or was a failure. • D8-VDVCMCDR – Validate if Device has appropriate Measuring Component: This algorithm performs a check to ensure that the reading has a proper measuring component.

Service Script: Itron - DataArrived Request

This Groovy—based service script handles the incoming DataArrived requests. It saves the payload to file storage locations (including Cloud Object Storage buckets in the case of cloud implementations) based on configuration.

Schema

Copyright © 2026, Oracle and/or its affiliates.

The schema maps closely to the Itron DataArrived message. It includes the D8-DataArrived data area.

Steps

Step 10 Edit Data – Invoke Main Groovy Method

This step shows how to trigger a Groovy script from other OUAF objects. It is similar to a main method in other programming languages. In this case, it is called by an Inbound Web Service method.

Step 20 – Groovy Imports

This step imports other classes needed by the script.

Step 30 – Groovy Members

1. Look up the Master Configuration object
2. Use the Master Configuration to find the File Storage details
    a. Find the File Adapter type
    b. Compute a filename using a configured generation script
    c. Compute a fill URL using the bucket, filename, and other details
3. Use the FileStorageAdapter business service to save the file to the targeted location.

Message Sender XSLs

Outbound messages created by on-demand read outbound communications use the following XSL files:

ContingencyReadByEndpoints

- **Request XSL**: D8-ContinegencyReadByEndpointsRequest.xsl
- **Response XSL**: D8-ContingencyReadByEndpointsResponse.xsl

GetContingencyReadByEndpointsResult

- **Request XSL**: D8-GetContingencyReadByEndpointsResultRequest.xsl
- **Response XSL**: D8-GetContingencyReadByEndpointsResultResponse.xsl

Refer to Creating Message Senders for more information about creating message senders.

Inbound Web Services

The Itron on-demand read process uses the following inbound web services.

D8-RequestStatusChangedService

- **Operation Name**: StatusChanged
- **Schema Name**: Itron — Status Changed BO
- **Request XSL**: D8-RequestStatusChangedServiceRequest.xsl
- **Response XSL**: D8-RequestStatusChangedServiceResponse.xsl

D8-DataSubscriberService

- **Operation Name**: DataArrived
- **Schema Name**: Itron — DataArrived Request Script
- **Request XSL**: D8-DataSubscriberServiceRequest.xsl

Copyright © 2026, Oracle and/or its affiliates.

- **Response XSL**: D8-DataSubscriberServiceResponse.xsl

Itron On-Demand Read XSL Transformations

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports multiple versions of the Itron OpenWay API. To achieve this support, several of the request XSLs check a version number embedded in the source schema. Similarly, the responses XSLs are designed to handle different source XML and map the values to the common SGG format.

The Itron on-demand read process uses the following XSL transformations.

D8-ContRdByEndptsRequest.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:ns2="http://www.itron.com/ami/2008/10/common"
        xmlns:ns3="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
       <ContingencyReadByEndpoints xmlns="http://www.itron.com/ami/2008/10/
data">
          <xsl:apply-templates select="./requestEnvelope/
ContingencyReadByEndpoints/*"/>
       </ContingencyReadByEndpoints>
    </xsl:template>


    <xsl:template match="ElectronicSerialNumbers">
        <xsl:element name="ns2:ElectronicSerialNumbers">
            <xsl:for-each select="./string">
                <xsl:element name="ns3:string">
                    <xsl:value-of select="."/>
                </xsl:element>
            </xsl:for-each>
        </xsl:element>
    </xsl:template>


    <xsl:template match="StatusChangedService">
        <xsl:element name="ns2:StatusChangedService">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>


    <xsl:template match="Parameters">
        <ns1:Parameters>
            <ns1:ReadingStartTime><xsl:value-of select="./
ReadingStartTime"/></ns1:ReadingStartTime>
            <ns1:ReadingEndTime><xsl:value-of select="./ReadingEndTime"/></
ns1:ReadingEndTime>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <xsl:if test="string-length(./PerformDemandReset) > 0">
                <ns1:PerformDemandReset><xsl:value-of select="./
PerformDemandReset"/></ns1:PerformDemandReset>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveHomeNetworkData) > 0">
                <ns1:RetrieveHomeNetworkData><xsl:value-of select="./
RetrieveHomeNetworkData"/></ns1:RetrieveHomeNetworkData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveInstantaneousData) > 0">
                <ns1:RetrieveInstantaneousData><xsl:value-of select="./
RetrieveInstantaneousData"/></ns1:RetrieveInstantaneousData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLastDemandReset) > 0">
                <ns1:RetrieveLastDemandReset><xsl:value-of select="./
RetrieveLastDemandReset"/></ns1:RetrieveLastDemandReset>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLoadProfileData) > 0">
                <ns1:RetrieveLoadProfileData><xsl:value-of select="./
RetrieveLoadProfileData"/></ns1:RetrieveLoadProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLogEvents) > 0">
                <ns1:RetrieveLogEvents><xsl:value-of select="./
RetrieveLogEvents"/></ns1:RetrieveLogEvents>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveNetworkStatistics) > 0">
                <ns1:RetrieveNetworkStatistics><xsl:value-of select="./
RetrieveNetworkStatistics"/></ns1:RetrieveNetworkStatistics>
            </xsl:if>
            <xsl:if test="string-length(./RetrievePriorSelfRead) > 0">
                <ns1:RetrievePriorSelfRead><xsl:value-of select="./
RetrievePriorSelfRead"/></ns1:RetrievePriorSelfRead>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveRecentRegisters) > 0">
                <ns1:RetrieveRecentRegisters><xsl:value-of select="./
RetrieveRecentRegisters"/></ns1:RetrieveRecentRegisters>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveVoltageMonitorData) > 0">
                <ns1:RetrieveVoltageMonitorData><xsl:value-of select="./
RetrieveVoltageMonitorData"/></ns1:RetrieveVoltageMonitorData>
            </xsl:if>
            <xsl:if test="string-length(./ReportSelfReadOption) > 0">
                <ns1:ReportSelfReadOption><xsl:value-of select="./
ReportSelfReadOption"/></ns1:ReportSelfReadOption>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveExtendedCurrentData) > 0">
                <ns1:RetrieveExtendedCurrentData><xsl:value-of select="./
RetrieveExtendedCurrentData"/></ns1:RetrieveExtendedCurrentData>
            </xsl:if>
            <xsl:if test="string-length(./RetrievePriorExtendedSelfReadData)
> 0">
                <ns1:RetrievePriorExtendedSelfReadData><xsl:value-of
select="./RetrievePriorExtendedSelfReadData"/></
ns1:RetrievePriorExtendedSelfReadData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveInstrumentationProfileData)
> 0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                <ns1:RetrieveInstrumentationProfileData><xsl:value-of
select="./RetrieveInstrumentationProfileData"/></
ns1:RetrieveInstrumentationProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveExtendedLoadProfileData) >
0">
                <ns1:RetrieveExtendedLoadProfileData><xsl:value-of select="./
RetrieveExtendedLoadProfileData"/></ns1:RetrieveExtendedLoadProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveToolboxData) > 0">
                <ns1:RetrieveToolboxData><xsl:value-of select="./
RetrieveToolboxData"/></ns1:RetrieveToolboxData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveTemperatureMonitoringData)
> 0">
                <ns1:RetrieveTemperatureMonitoringData><xsl:value-of
select="./RetrieveTemperatureMonitoringData"/></
ns1:RetrieveTemperatureMonitoringData>
            </xsl:if>
        </ns1:Parameters>
    </xsl:template>


    <xsl:template match="*">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/data">
            <xsl:apply-templates select="*|text()"/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

D8-ContRdByEndptsResponse.xsl

```
<xsl:stylesheet
        version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="env ns1 com">

<xsl:output method="xml" encoding="utf-8" indent="yes"/>

<xsl:template match="env:Envelope">
   <xsl:apply-templates select="./env:Body/*"/>
</xsl:template>


<xsl:template match="ns1:ContingencyReadByEndpointsResponse">
    <responseDetail>
       <responseEnvelope>
           <ContingencyReadByEndpointsResponse>
               <xsl:apply-templates select="./*"/>
           </ContingencyReadByEndpointsResponse>
```


Copyright © 2026, Oracle and/or its affiliates.

```
       </responseEnvelope>
    </responseDetail>
</xsl:template>


<xsl:template match="env:Fault[./detail/*[contains('|OperationFault|
NoValidTargetsFoundFault|InvalidUriFormatFault|InvalidDateTimeRangeFault|
ArgumentFault|TooManyConcurrentRequestsFault|SecurityFault|',
concat('|',local-name(),'|'))]]">
    <fault>
        <xsl:apply-templates select="./detail/*"/>
    </fault>
</xsl:template>


<xsl:template match="env:Fault">
    <fault>
        <xsl:apply-templates select="./*"/>
    </fault>
</xsl:template>


<xsl:template match="detail">
    <detail><xsl:apply-templates select="./*" mode="serialize"/></detail>
</xsl:template>


<xsl:template match="*">
    <xsl:variable name="name">
       <xsl:value-of select="local-name()"/>
   </xsl:variable>
   <xsl:element name="{$name}" namespace="">
       <xsl:apply-templates select="*|text()"/>
   </xsl:element>
</xsl:template>


<xsl:template match="*" mode="serialize">
    <xsl:text><</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:for-each select="@*">
        <xsl:value-of select="concat(' ', local-name())"/>
        <xsl:text>="</xsl:text>
        <xsl:value-of select="."/>
        <xsl:text>"</xsl:text>
    </xsl:for-each>
    <xsl:text>></xsl:text>
    <xsl:apply-templates select="*" mode="serialize"/>
    <xsl:value-of select="."/>
    <xsl:text></</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:text>></xsl:text>
</xsl:template>

</xsl:stylesheet>
```


Copyright © 2026, Oracle and/or its affiliates.

D8-GetContRdByEndptsRsReq.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:ns2="http://www.itron.com/ami/2008/10/common">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
      <GetContingencyReadByEndpointsResult xmlns="http://www.itron.com/ami/
2008/10/data">
         <xsl:apply-templates select="./requestEnvelope/
GetContingencyReadByEndpointsResult/*" />
      </GetContingencyReadByEndpointsResult>
    </xsl:template>

    <xsl:template match="Id">
        <xsl:element name="ns2:Id">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/data">
            <xsl:apply-templates select="*|text()"/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

D8-GetContRdByEndptsRsResp.xsl

```
<xsl:stylesheet
        version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="env ns1 com">

<xsl:output method="xml" encoding="utf-8" indent="yes"/>

<xsl:template match="env:Envelope">
    <xsl:apply-templates select="./env:Body/*"/>
</xsl:template>

<xsl:template match="ns1:GetContingencyReadByEndpointsResultResponse">
    <responseDetail>
       <responseEnvelope>
           <GetContingencyReadByEndpointsResultResponse>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                <xsl:apply-templates select="./*"/>
            </GetContingencyReadByEndpointsResultResponse>
        </responseEnvelope>
    </responseDetail>
</xsl:template>

<xsl:template match="*">
    <xsl:variable name="name">
        <xsl:value-of select="local-name()"/>
    </xsl:variable>
    <xsl:element name="{$name}" namespace="">
        <xsl:apply-templates select="*|text()"/>
    </xsl:element>
</xsl:template>

<xsl:template match="env:Fault[./detail/*[contains('|OperationFault|
RequestNotFoundFault|RequestNotFinishedFault|ArgumentFault|
TooManyConcurrentRequestsFault|SecurityFault|', concat('|',local-
name(),'|'))]]">
    <fault>
        <xsl:apply-templates select="./detail/*"/>
    </fault>
</xsl:template>


<xsl:template match="env:Fault">
    <fault>
        <xsl:apply-templates select="./*"/>
    </fault>
</xsl:template>


<xsl:template match="detail">
    <detail><xsl:apply-templates select="./*" mode="serialize"/></detail>
</xsl:template>

<xsl:template match="*" mode="serialize">
    <xsl:text><</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:for-each select="@*">
        <xsl:value-of select="concat(' ', local-name())"/>
        <xsl:text>="</xsl:text>
        <xsl:value-of select="."/>
        <xsl:text>"</xsl:text>
    </xsl:for-each>
    <xsl:text>></xsl:text>
    <xsl:apply-templates select="*" mode="serialize"/>
    <xsl:value-of select="."/>
    <xsl:text></</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:text>></xsl:text>
</xsl:template>

</xsl:stylesheet>
```


Copyright © 2026, Oracle and/or its affiliates.

D8-DataSubSvcRequest.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
        xmlns:data="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        xmlns:ev="http://www.itron.com/ami/2008/10/events"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        exclude-result-prefixes="data com ev soap11 soap12">

    <xsl:template match="soap11:Envelope">
        <xsl:apply-templates select="//data:DataArrived"/>
    </xsl:template>


    <xsl:template match="soap11:Body">
        <xsl:apply-templates select="//data:DataArrived"/>
    </xsl:template>


    <xsl:template match="data:DataArrived">
        <DataArrived>
            <input>
                <RequestToken>
                    <Id><xsl:value-of select="./data:input/data:RequestToken/
com:Id"/></Id>
                </RequestToken>
                <ReadDataCollection>
                    <xsl:for-each select="./data:input/
data:ReadDataCollection/data:ReadData">
                        <ReadData>
                            <Identifier><xsl:value-of select="./
data:Identifier"/></Identifier>
                            <DeviceClass><xsl:value-of select="./
data:DeviceClass"/></DeviceClass>
                            <ScalingType><xsl:value-of select="./
data:ScalingType"/></ScalingType>
                            <xsl:call-template name="EventLog"/>
                            <xsl:call-template name="LoadProfileChannels"/>
                            <xsl:call-template name="NetworkStatistics"/>
                            <xsl:call-template name="RegisterValues"/>
                            <xsl:call-template name="BlockInfo"/>
                            <xsl:if test="./data:DeviceSerialNumber">
                                <DeviceSerialNumber><xsl:value-of select="./
data:DeviceSerialNumber"/></DeviceSerialNumber>
                            </xsl:if>
                        </ReadData>
                    </xsl:for-each>
                </ReadDataCollection>
                <xsl:if test="count(./data:input/data:RequestMetadata/*) > 0">
                    <RequestMetadata>
                        <xsl:copy-of select="./data:input/
data:RequestMetadata/*"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    </RequestMetadata>
                </xsl:if>
            </input>
        </DataArrived>
    </xsl:template>


    <xsl:template name="EventLog">
        <xsl:if test="count(./data:EventLog/ev:MeterEvents/*) > 0 or count(./
data:EventLog/ev:UnreportedEventCount) > 0">
            <EventLog>
                <xsl:if test="count(./data:EventLog/ev:MeterEvents/*) > 0">
                    <MeterEvents>
                        <xsl:for-each select="./data:EventLog/ev:MeterEvents/
ev:MeterEvent">
                            <MeterEvent>
                                <Timestamp><xsl:value-of select="./
ev:Timestamp"/></Timestamp>
                                <Category><xsl:value-of select="./
ev:Category"/></Category>
                                <LogType><xsl:value-of select="./
ev:LogType"/></LogType>
                                <Name><xsl:value-of select="./ev:Name"/></
Name>
                                <xsl:if test="count(./ev:Arguments/*) > 0">
                                    <Arguments>
                                        <xsl:for-each select="./ev:Arguments/
ev:Argument">
                                            <Argument>
                                                <Name><xsl:value-of select="./
ev:Name"/></Name>
                                                <Value><xsl:value-of
select="./ev:Value"/></Value>
                                            </Argument>
                                        </xsl:for-each>
                                    </Arguments>
                                </xsl:if>
                                <xsl:if test="./ev:ID">
                                    <ID><xsl:value-of select="./ev:ID"/></ID>
                                </xsl:if>
                                <xsl:if test="./ev:Source">
                                    <Source><xsl:value-of select="./
ev:Source"/></Source>
                                </xsl:if>
                                <xsl:if test="./ev:SequenceNumber">
                                    <SequenceNumber><xsl:value-of select="./
ev:SequenceNumber"/></SequenceNumber>
                                </xsl:if>
                            </MeterEvent>
                        </xsl:for-each>
                    </MeterEvents>
                </xsl:if>
                <xsl:if test="./data:EventLog/ev:UnreportedEventCount">
                    <UnreportedEventCount><xsl:value-of select="./
data:EventLog/ev:UnreportedEventCount"/></UnreportedEventCount>
                </xsl:if>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </EventLog>
        </xsl:if>
    </xsl:template>


    <xsl:template name="LoadProfileChannels">
        <xsl:if test="count(./data:LoadProfileChannels/
data:LoadProfileChannel/data:IntervalLength) > 0">
            <LoadProfileChannels>
                <xsl:for-each select="./data:LoadProfileChannels/
data:LoadProfileChannel">
                    <LoadProfileChannel>
                        <IntervalLength><xsl:value-of select="./
data:IntervalLength"/></IntervalLength>
                        <PulseMultiplier><xsl:value-of select="./
data:PulseMultiplier"/></PulseMultiplier>
                        <Quantity><xsl:value-of select="./data:Quantity"/></
Quantity>
                        <TimeDataEnd><xsl:value-of select="./
data:TimeDataEnd"/></TimeDataEnd>
                        <IntervalValues>
                            <xsl:for-each select="./data:IntervalValues/
data:IntervalValue">
                                <IntervalValue>
                                    <ChannelValue><xsl:value-of select="./
data:ChannelValue"/></ChannelValue>
                                    <xsl:for-each select="./
data:ProfileStatuses/data:ProfileStatus">
                                        <ProfileStatuses>
                                            <ProfileStatus><xsl:value-of
select="."/></ProfileStatus>
                                        </ProfileStatuses>
                                    </xsl:for-each>
                                </IntervalValue>
                            </xsl:for-each>
                        </IntervalValues>
                    </LoadProfileChannel>
                </xsl:for-each>
            </LoadProfileChannels>
        </xsl:if>
    </xsl:template>


    <xsl:template name="NetworkStatistics">
        <xsl:if test="count(./data:NetworkStatistics/data:NetworkStatistic/
data:Name) > 0">
            <NetworkStatistics>
                <xsl:for-each select="./data:NetworkStatistics/
data:NetworkStatistic">
                    <NetworkStatistic>
                        <Name><xsl:value-of select="./data:Name"/></Name>
                        <NetworkKind><xsl:value-of select="./
data:NetworkKind"/></NetworkKind>
                        <Value><xsl:value-of select="./data:Value"/></Value>
                    </NetworkStatistic>
                </xsl:for-each>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </NetworkStatistics>
        </xsl:if>
    </xsl:template>


    <xsl:template name="RegisterValues">
        <xsl:if test="count(./data:RegisterValues/data:RegisterValue/
data:Quantity) > 0">
            <RegisterValues>
                <xsl:for-each select="./data:RegisterValues/
data:RegisterValue">
                    <RegisterValue>
                        <Quantity><xsl:value-of select="./data:Quantity"/></
Quantity>
                        <Source><xsl:value-of select="./data:Source"/></
Source>
                        <Timestamp><xsl:value-of select="substring(./
data:Timestamp, 1, 19)"/></Timestamp>
                        <Value><xsl:value-of select="./data:Value"/></Value>
                        <xsl:if test="count(./data:Status/data:string) > 0">
                            <Status>
                                <xsl:for-each select="./data:Status/
data:string">
                                    <string><xsl:value-of select="."/></
string>
                                </xsl:for-each>
                            </Status>
                        </xsl:if>
                    </RegisterValue>
                </xsl:for-each>
            </RegisterValues>
        </xsl:if>
    </xsl:template>


    <xsl:template name="BlockInfo">
        <xsl:if test="count(./data:BlockInfo/data:Block) > 0">
            <BlockInfo>
                <Block><xsl:value-of select="./data:BlockInfo/data:Block"/></
Block>
                <TotalBlocks><xsl:value-of select="./data:BlockInfo/
data:TotalBlocks"/></TotalBlocks>
            </BlockInfo>
        </xsl:if>
    </xsl:template>


    <!-- When the SOAP Message has already been transformed by the SOAP Proxy
-->
    <xsl:template match="DataArrived[namespace-uri(.) != 'http://
www.itron.com/ami/2008/10/data']">
        <xsl:copy-of select="."/>
    </xsl:template>

</xsl:stylesheet>
```


Copyright © 2026, Oracle and/or its affiliates.

D8-DataSubSvcResponse.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:tns="http://www.itron.com/ami/2008/10/data">

    <xsl:template match="/">
        <soap11:Envelope>
            <soap11:Header/>
            <soap11:Body>
                <tns:DataArrivedResponse/>
            </soap11:Body>
        </soap11:Envelope>
    </xsl:template>

</xsl:stylesheet>
```

D8-ReqStsChgSvcRequest.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="soap11 soap12 com">

    <xsl:template match="soap11:Envelope">
        <xsl:apply-templates select="//com:StatusChanged"/>
    </xsl:template>


    <xsl:template match="soap11:Body">
        <xsl:apply-templates select="//com:StatusChanged"/>
    </xsl:template>


    <xsl:template match="com:StatusChanged">
        <StatusChanged>
            <bo/>
            <input>
                <RequestStatus>
                    <RequestToken>
                        <Id><xsl:value-of select="./com:input/
com:RequestStatus/com:RequestToken/com:Id"/></Id>
                    </RequestToken>
                    <State><xsl:value-of select="./com:input/
com:RequestStatus/com:State"/></State>
                    <EndpointsScheduled><xsl:value-of select="./com:input/
com:RequestStatus/com:EndpointsScheduled"/></EndpointsScheduled>
                    <Result><xsl:value-of select="./com:input/
com:RequestStatus/com:Result"/></Result>
                    <SuccessfulEndpoints><xsl:value-of select="./com:input/
com:RequestStatus/com:SuccessfulEndpoints"/></SuccessfulEndpoints>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <FailedEndpoints><xsl:value-of select="./com:input/
com:RequestStatus/com:FailedEndpoints"/></FailedEndpoints>
                    <xsl:if test="string-length(./com:input/com:RequestStatus/
com:CreatedWhen) > 0">
                        <CreatedWhen><xsl:value-of select="substring(./
com:input/com:RequestStatus/com:CreatedWhen, 1, 23)"/></CreatedWhen>
                    </xsl:if>
                    <xsl:if test="string-length(./com:input/com:RequestStatus/
com:FinishedWhen) > 0">
                        <FinishedWhen><xsl:value-of select="substring(./
com:input/com:RequestStatus/com:FinishedWhen, 1, 23)"/></FinishedWhen>
                    </xsl:if>
                    <xsl:if test="count(./com:input/com:RequestStatus/
com:Metadata/*) > 0">
                        <Metadata>
                            <xsl:copy-of select="./com:input/
com:RequestStatus/com:Metadata/*"/>
                        </Metadata>
                    </xsl:if>
                </RequestStatus>
            </input>
        </StatusChanged>
    </xsl:template>


    <!-- When the SOAP Message has already been transformed by the SOAP Proxy
-->
    <xsl:template match="StatusChanged[namespace-uri(.) != 'http://
www.itron.com/ami/2008/10/common']">
        <xsl:copy-of select="."/>
    </xsl:template>

</xsl:stylesheet>
```

D8-RequestStatusChangedServiceResponse.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:tns="http://www.itron.com/ami/2008/10/common">

    <xsl:template match="/">
        <soap11:Envelope>
            <soap11:Header/>
            <soap11:Body>
                <tns:StatusChangedResponse/>
            </soap11:Body>
        </soap11:Envelope>
    </xsl:template>

</xsl:stylesheet>
```

Cloud Service Test Harness

Copyright © 2026, Oracle and/or its affiliates.

The Smart Grid Gateway Cloud Service Test Harness is used to simulate a head end system for testing smart meter commands in a cloud implementation.

The test harness is "stateless", meaning it does not store meters or their current status. The harness is designed to return a successful result to all command requests by default. Devices must be created in a specific manner to return an error.

How the Test Harness Works

The test harness includes SOAP web service implementations for each command supported by the various Smart Grid Gateway adapters. The Test Harness's container is managed by Oracle cloud services and internal routing definitions allow web service requests to locate the proper container without requiring users to update ports or other configurations. There is no user-accessible configuration required for the test harness itself.

The test harness uses a specific system-owned user (D1SGG) to send asynchronous communications back to Smart Grid Gateway.

Creating Devices for use with the Test Harness

The test harness works with any sort of device capable to sending smart meter commands. By default, all commands are successful.

Devices must be created in a specific manner in order for errors to be returned from the test harness. Specifically, commands for devices whose identifiers (badge number, serial number, internal meter number, pallet number, external ID, MDM external ID, and asset ID) end in "ERR" will return an error. This means that when setting up devices for testing via the test harness, you must create separate devices for successful and unsuccessful (error) results.

The simplest approach when creating "error" devices is to use the same identifier value for all identifiers, but at a minimum, "ERR" MUST be the final part of the identifier value for the **AMI Device ID Type** and **AMI Pre-Commissioning Device ID Type** configured on the head end system.

Defining Head End Systems, Processing Methods, Outbound Message Types, and Message Senders for use with the Test Harness

This section provides guidelines for creating data and records used with smart meter commands when using the test harness.

Head End System

When configuring a head end system for use with the test harness, note the identifier types specified in the **AMI Device ID Type** and **AMI Pre-Commissioning Device ID Type** fields. When creating devices intended to return an error, the identifier values for the identifier type(s) specified for these fields MUST end in "ERR" to return an error from the test harness.

Outbound Message Types

When creating outbound message types for use with the test harness, use the appropriate outbound business object based on the specific command. See **Smart Meter Command Flows** for details.

Message Senders

When creating message senders for use with the test harness, use the following settings:

Main tab:

- **Invocation Type**: Real-Time
- **Message Class**: SOAPSNDR

Copyright © 2026, Oracle and/or its affiliates.

- **Active**: <checked>
- **Msg Encoding**: UTF-8 message encoding

Context tab:

- **HTTP Header**: soapAction: <soapaction>

  where <soapaction> is the URI (Soap Action) for the command as shown under **Adapter-Specific Test Harness Commands** below.

  Note the format used here is:

  SoapAction: {URI}

  That's using a **colon, followed by a space** between the SoapAction and the value. You **cannot** use an equals (=) sign. For example:

  ```
  SoapAction: http://www.multispeak.org/Version_4.1_Release/
  MeterAddNotification
  ```

- **HTTP Login User**: A user which has been assigned to the SGGHarnessUser role in Oracle Identity Cloud Service (IDCS). See **Setting Up a User with Access to the Smart Grid Gateway Test Harness** in the *Oracle Utilities Cloud Services Administration Guide* for more information.
- **HTTP Login Password**: The password for the user above.
- **HTTP Method**: POST
- **HTTP URL 1**: <endpoint_URL>

  where <endpoint_URL> is the HTTP URL or the command as shown under **Adapter-Specific Test Harness Commands** below, formatted as follows:

  ```
  @SGG_TEST@<adapter_name>/<webservice>
  ```

  where:

  – @SGG_TEST@ is a token that combines the protocol, host, port, tenant, domain, and application

  – <adapter_name> is one “lg”, “itron”, “ssn”, or “sensus”, as appropriate

  – <webservice> is a name specific to each adapter (e.g. OD_Server, MR_CB, DeviceManager, and so on. See Adapter-Specific Test Harness Commands)

> **Note**
>
> When creating message senders for use with the test harness, do **NOT** specify a **Sender Security Type** on the **Context** tab.

External Systems

When creating outbound message types on external systems and for use with the test harness, be sure to specify the appropriate Message XSL and Response XSL files based on the outbound message business object. See Creating an External Systemfor lists of these files.

Processing Methods

Command processing methods for a head end system used with the test harness should reference an Outbound Message Type, either the **Default Outbound Message Type** or an **Outbound Message Type** for a specific **Device Type**.

Copyright © 2026, Oracle and/or its affiliates.

- Adapter-Specific Test Harness Commands

Adapter-Specific Test Harness Commands

This section provides specific details for invoking commands for specific adapters.

Each adapter comprises multiple web services that enable commands. The commands available for each web service for each adapter are provided below.

HTTP Header Format

The format used for the HTTP Header is as follows:

SoapAction: {URI}

That's using a **colon, followed by a space** between the SoapAction and the value. You **cannot** use an equals (=) sign. For example:

```
SoapAction: http://www.multispeak.org/Version_4.1_Release/
MeterAddNotification
```

Itron OpenWay Commands

Control Service

The table below lists details for commands available via the Itron OpenWay Control Service web service.

Command	Web Method	HTTP Header	HTTP URL
Connect	ReconnectMeter	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/ReconnectMeter	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control
Connect (Result)	GetReconnectMeterResult	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetReconnectMeterResult	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control
Disconnect	DisconnectMeter	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/DisconnectMeter	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control
Disconnect (Result)	GetDisconnectMeterResult	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetDisconnectMeterResult	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control
Load Check	DetectLoadSideVoltageByMeter	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/DetectLoadSideVoltageByMeter	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP Header	HTTP URL
Load Check (Result)	GetDetectLoadSideVoltageByMeterResult	SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetDetectLoadSideVoltageByMeterResult	@SGG_TEST@/itron/www.itron.com.ami.2008.10.control

Control Diagnostic

The table below lists details for commands available via the Itron OpenWay Control Diagnostic web service.

Command	Web Method	HTTP Header	HTTP URL
Device Status Check	PingByEndpoints	SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/PingByEndpoints	@SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic
Device Status Check (Result)	GetPingByEndpointsResult	SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/GetPingByEndpointsResult	@SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic
Read Disconnect State	ReadDisconnectStateByMeters	SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/ReadDisconnectStateByMeters	@SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic
Read Disconnect State (Result)	GetReadDisconnectStateByMetersResult	SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/GetReadDisconnectStateByMetersResult	@SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic

Data Service

The table below lists details for commands available via the Itron OpenWay Data Service web service.

Command	Web Method	HTTP Header	HTTP URL
On-Demand Read	ContingencyReadByEndpoints	SoapAction: http://www.itron.com/ami/2008/10/data/DataService/ContingencyReadByEndpoints	@SGG_TEST@/itron//www.itron.com.ami.2008.10.data

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP Header	HTTP URL
On-Demand Read (Result)	GetContingencyReadByEndpointsResult	SoapAction: http://www.itron.com/ami/2008/10/data/DataService/GetContingencyReadByEndpointsResult	@SGG_TEST@/itron//www.itron.com.ami.2008.10.data
Scheduled Read	InterrogateByGroup	SoapAction: http://www.itron.com/ami/2008/10/data/DataService/InterrogateByGroup	@SGG_TEST@/itron//www.itron.com.ami.2008.10.data
Scheduled Read (Result)	GetInterrogateByGroupResult	SoapAction: http://www.itron.com/ami/2008/10/data/DataService/GetInterrogateByGroupResult	@SGG_TEST@/itron//www.itron.com.ami.2008.10.data

Provisioning Service

The table below lists details for commands available via the Itron OpenWay Provisioning Service web service.

Command	Web Method	HTTP Header	HTTP URL
Commission	AddMeterDefinitions	SoapAction: http://www.itron.com/ami/2009/08/provisioning/ProvisioningService/AddMeterDefinitions	@SGG_TEST@/itron/www.itron.com.ami.2009.08.provisioning
Decommission	DeregisterMeters	SoapAction: http://www.itron.com/ami/2009/08/provisioning/ProvisioningService/DeregisterMeters	@SGG_TEST@/itron/www.itron.com.ami.2009.08.provisioning

Landis+Gyr Commands

MR_CB

The table below lists details for commands available via the Landis+Gyr MR_CB web service.

Command	Web Method	HTTP Header	HTTP URL
Commission	MeterAddNotification	SoapAction: http://www.multispeak.org/Version_3.0/MeterRemoveNotification	@SGG_TEST@/lg/MR_CB
Decommission	MeterRemoveNotification	SoapAction: http://www.multispeak.org/Version_3.0/MeterAddNotification	@SGG_TEST@/lg/MR_CB

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP Header	HTTP URL
Meter Retire	MeterRetireNotification	SoapAction: http://www.multispeak.org/Version_3.0/MeterRetireNotification	@SGG_TEST@/lg/MR_CB
On-Demand Read	InitiateMeterReadByMeterNumber	SoapAction: http://www.multispeak.org/Version_3.0/InitiateMeterReadByMeterNumber	@SGG_TEST@/lg/MR_CB
Scheduled Demand Reset	ScheduleDemandReset	SoapAction: http://www.multispeak.org/Version_3.0/ScheduleDemandReset	@SGG_TEST@/lg/MR_CB

CD_MR

The table below lists details for commands available via the Landis+Gyr CD_MR web service.

Command	Web Method	HTTP Header	HTTP URL
Meter Exchange	MeterExchangeNotification	SoapAction: http://www.multispeak.org/Version_3.0/MeterExchangeNotification	@SGG_TEST@/lg/CB_MR

CD_CB

The table below lists details for commands available via the Landis+Gyr CD_CB web service.

Command	Web Method	HTTP Header	HTTP URL
Connect/Disconnect	InitiateConnectDisconnect	SoapAction: http://www.multispeak.org/Version_3.0/InitiateConnectDisconnect	@SGG_TEST@/lg/CD_CB

Metering

The table below lists details for commands available via the Landis+Gyr Metering web service.

Command	Web Method	HTTP Header	HTTP URL
Add Meter to Inventory	AddMeterToInventory	SoapAction: turtletech.com/TS2/AddMeterToInventory	@SGG_TEST@/lg/Metering

CIM

The table below lists details for commands available via the Landis+Gyr CIM web service.

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP Header	HTTP URL
On Demand Read (Interval & Scalar) Device Status Check	AMIRequest	SoapAction: urn:LandisGyr.GSIS.CimService.CIMService.AMIRequest	@SGG_TEST@/lg/CIMService

Note: The Landis+Gyr ScheduleDemandReset API does not offer a way to pass a callback URL in. In actual use, L+G allows the callback to be configured in their system. To handle the scenario in the Test Harness, add an additional HTTP header named **SGG-Callback-Endpoint** to the callback URL in the Message Sender. The value of the header should be the address of the D3-CB_MR Inbound Web Service, for example: SGG-Callback-Endpoint: “https://144.25.14.142:8086/iddev2/prod/ccs/soap/api/iws/D3-CB_MR”. The URL value is located near the bottom of the WSDL on the Inbound Web Service record.

Sensus Commands

MR_Server

The table below lists details for commands available via the Sensus MR_Server web service.

Command	Web Method	HTTP Header	HTTP URL
Commission	MeterAddNotification	SoapAction: http://www.multispeak.org/Version_4.1_Release/MeterAddNotification	@SGG_TEST@/sensus/MR_Server
Decommission	MeterRemoveNotification	SoapAction: http://www.multispeak.org/Version_4.1_Release/MeterRemoveNotification	@SGG_TEST@/sensus/MR_Server
On-Demand Read	InitiateMeterReadingsByMeterNumber	SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateMeterReadingsByMeterID	@SGG_TEST@/sensus/MR_Server

CD_Server

The table below lists details for commands available via the Sensus CD_Server web service.

Command	Web Method	HTTP Header	HTTP URL
Connect/Disconnect	InitiateConnectDisconnect	SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateConnectDisconnect	@SGG_TEST@/sensus/CD_Server

OD_Server

The table below lists details for commands available via the Sensus OD_Server web service.

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP Header	HTTP URL
Device Status Check	InitiateOutageDetectionEventRequest	SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateOutageDetectionEventRequest	@SGG_TEST@/sensus/OD_Server

Silver Spring Networks Commands

Note: The Silver Spring Networks adapter works differently from the other adapters. This adapter does not use SoapAction identifiers on the operations, so the HTTP Header can be left off of the Message Sender configuration.

DataAggregation

The table below lists details for commands available via the Silver Spring Networks DataAggregation web service.

Command	Web Method	HTTP URL
	GetMeterFieldStatus	@SGG_TEST@/ssn/DataAggregation

DeviceManager

The table below lists details for commands available via the Silver Spring Networks DeviceManager web service.

Command	Web Method	HTTP URL
All	FindDevice	@SGG_TEST@/ssn/DeviceManager
Commission	ReplaceLocation	@SGG_TEST@/ssn/DeviceManager
Decommission	ReplaceDeviceAtLocation	@SGG_TEST@/ssn/DeviceManager

JobManager

The table below lists details for commands available via the Silver Spring Networks JobManager web service.

Command	Web Method	HTTP URL
Connect/Disconnect	addRemoteProvisioningJob	@SGG_TEST@/ssn/JobManager
On-Demand Read	addMeterReadJob	@SGG_TEST@/ssn/JobManager
Device Status Check	addPingJob	@SGG_TEST@/ssn/JobManager
On-Demand Read, Device Status Check	getJobStatus	@SGG_TEST@/ssn/JobManager
Connect/Disconnect	getJobStatusForDevice	@SGG_TEST@/ssn/JobManager

DeviceResults

The table below lists details for commands available via the Silver Spring Networks DeviceResults web service.

Copyright © 2026, Oracle and/or its affiliates.

Command	Web Method	HTTP URL
Connect/Disconnect	getRemoteProvisioningResults ByJobID	@SGG_TEST@/ssn/ DeviceResults
On-Demand Read	getMeterReadResultsByJobID	@SGG_TEST@/ssn/ DeviceResults
Device Status Check	getPingresultsByJobID	@SGG_TEST@/ssn/ DeviceResults

Web Service Authentication

This section describes configuration settings for web service authentication for each Smart Grid Gateway Adapter. Each adapter section below includes guidelines for defining credentials to outbound web service requests and instructions for delivering credentials when calling back into the application via inbound web services.

For the purposes of this document, “Inbound” refers to requests originating from outside Smart Grid Gateway and are targeted at Smart Grid Gateway web services. “Outbound” requests originate within Smart Grid Gateway and are targeted at web services hosted on other servers.

- Itron OpenWay Web Service Authentication
- Landis+Gyr Web Service Authentication
- Sensus Web Service Authentication
- Silver Spring Networks Web Service Authentication
- Adapter Development Kit Web Service Authentication

Itron OpenWay Web Service Authentication

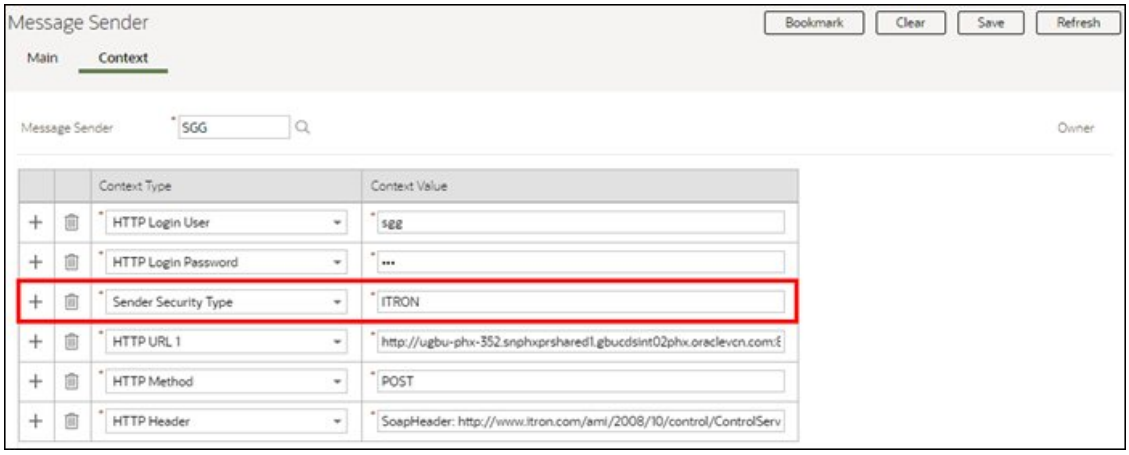
Inbound Requests

Incoming requests should utilize standard basic HTTPS authentication.

Outbound Requests

The Itron system can be configured to expect a WSSE security element in the SOAP header for authentication. That element will contain certain features such as a nonce, in addition to the credentials. This can be configured in Smart Grid Gateway by setting the Message Sender’s **Sender Security Type** value to "ITRON" (as shown in the following screen capture).

Copyright © 2026, Oracle and/or its affiliates.



Landis+Gyr Web Service Authentication

The L+G Adapter communicates using the MultiSpeak 3.0 standard as well as the Common Information Model (CIM) standard. The MultiSpeak standard includes a SOAP Header element, MultiSpeakMsgHeader with "UserId" and "Pwd" security attributes. The Message Sender configuration is used to insert credentials into the header element. Incoming requests will be scanned for header credentials when an instruction to do so is found in the URL.

Similarly, the CIM standard defines credential locations within the SOAP Header and Body, depending on the version.

MultiSpeak 3.0

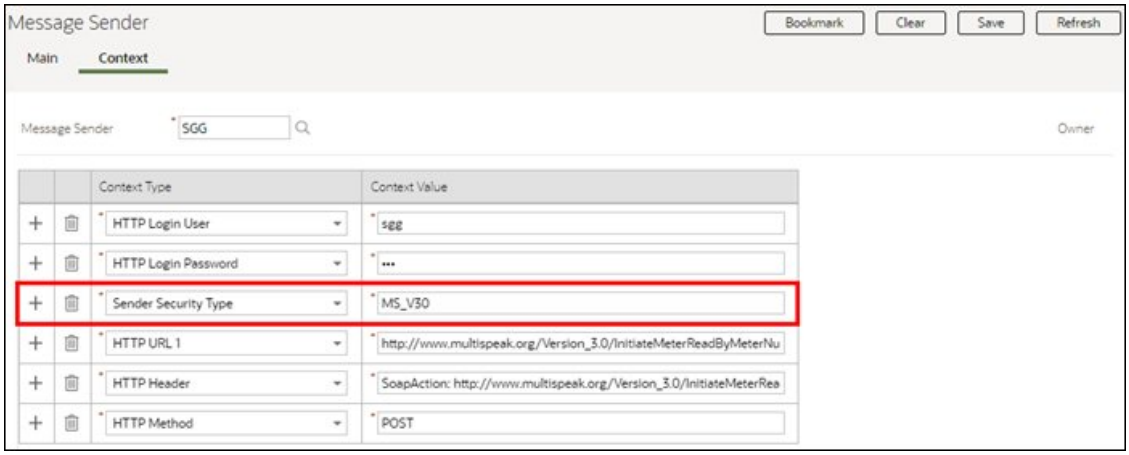
Inbound Requests

Include “/multispeak30” prior to the inbound web service name in the URL to signal the application that it should locate credentials in the MultiSpeakMsgHeader SOAP Header element.

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/multispeak30**/D3-CB_MR

Outbound Requests

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “MS_V30” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added as attributes of MultiSpeakMsgHeader in the outgoing SOAP header.



Copyright © 2026, Oracle and/or its affiliates.

CIM 1.2

Inbound Requests

Include “/cim12” prior to the inbound web service name in the URL to signal the application that it should include SOAP Body credentials ({Specific Message Type}/Header/User/UserId and {Specific Message Type}//Header/Property/Name[text() = 'Password']).

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/cim12**/D3-CIMCallback

Outbound Requests

CIM 1.2 outbound requests should utilize either standard basic HTTPS authentication or send credentials in the SOAP Header element as in CIM 2.0.

CIM 2.0

Inbound Requests

Include “/cim20” prior to the inbound web service name in the URL to signal the application that it should include credentials in the SOAP Header in the form of elements:

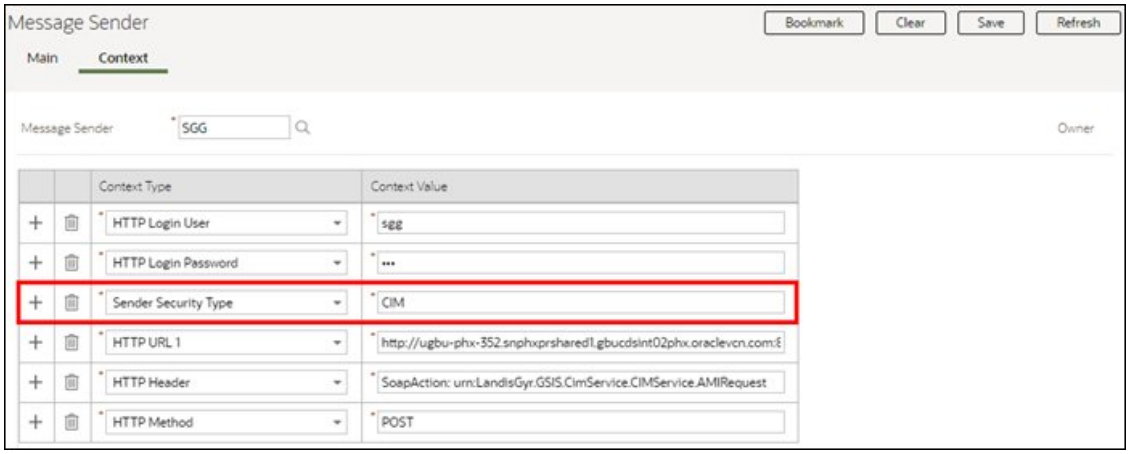
- {http://www.landisgyr.com/iec61968/2010/03}UserName
- {http://www.landisgyr.com/iec61968/2010/03}Password

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/cim20**/D3-CIMCallback

Outbound Requests

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “CIM” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added to the outgoing SOAP Header at as elements:

- {http://www.landisgyr.com/iec61968/2010/03}UserName
- {http://www.landisgyr.com/iec61968/2010/03}Password



Sensus Web Service Authentication

MultiSpeak 4.1 Release

Inbound Requests

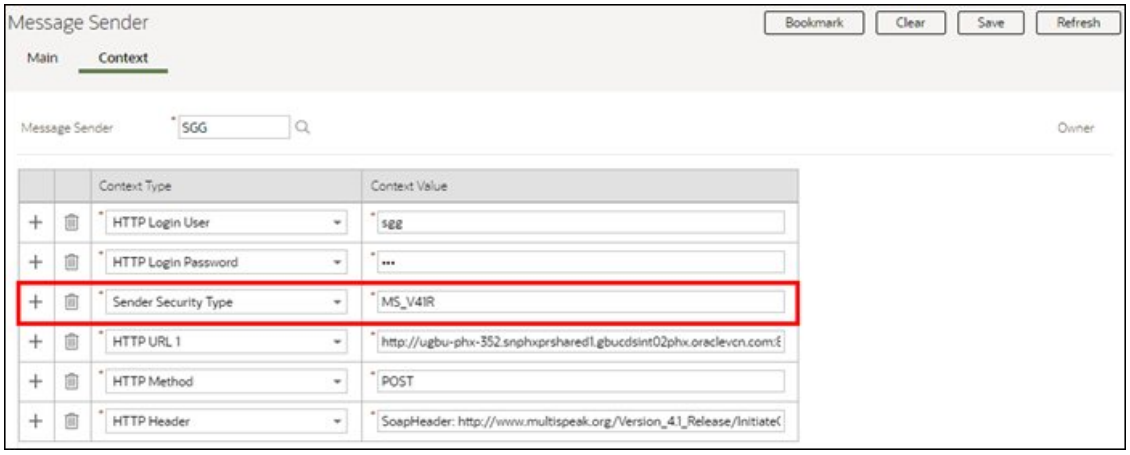
Copyright © 2026, Oracle and/or its affiliates.

Include “/multispeak41” prior to the inbound web service name in the URL to signal the application that it should locate credentials in the MultiSpeakMsgHeader SOAP Header element.

Ex: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/multispeak41**/D6-CB_Server

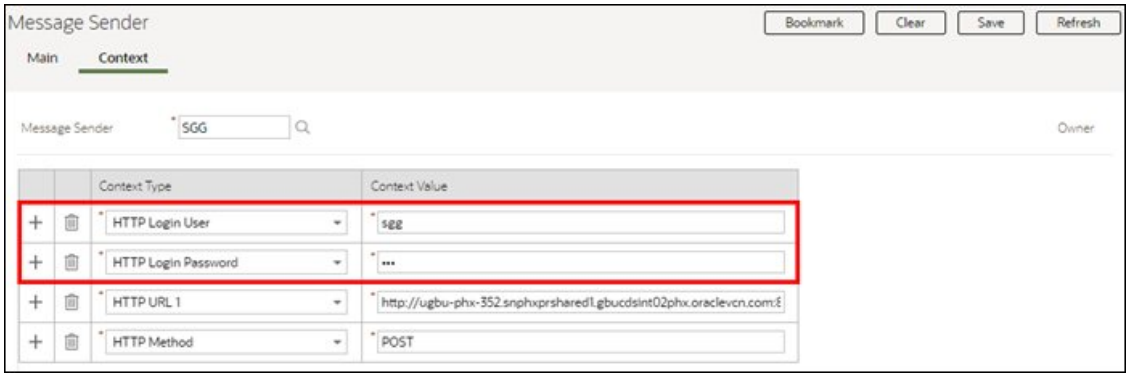
Outbound Requests

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “MS_V41R” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added as attributes of MultiSpeakMsgHeader in the outgoing SOAP header.



Silver Spring Networks Web Service Authentication

The Silver Spring Neworks AMM API does not support Inbound Requests (from the SilverSprings server to Smart Grid Gateway). Outbound requests should utilize standard basic HTTPS authentication options via the **HTTP Login User** and **HTTP Login Password** parameters on the **Context** tab of the **Message Sender** portal.



Adapter Development Kit Web Service Authentication

Inbound Requests

The Adapter Development Kit utilizes standard basic HTTPS authentication for inbound requests.

Copyright © 2026, Oracle and/or its affiliates.

Outbound Requests

Outbound requests should utilize standard basic HTTPS authentication options the **HTTP Login User** and **HTTP Login Password** parameters on the **Context** tab of the **Message Sender** portal.



Master Configurations

Master configurations define configuration settings for individual adapters. The sections below provide details for the master configuration provided for each Smart Grid Gateway adapter.

Itron OpenWay Master Configuration

The “Itron Configuration” master configuration defines details used by the Itron OpenWay commands and payload processing, including the following:

- **Itron Version**: Used to specify the version of the Itron OpenWay head end system being used.
- **Command Status Changed URL Configuration**: Used to define URL configuration for asynchronous callback requests.
  - **Status Changed Service URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Status Changed Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Commands**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations**: Used to define how payloads and exceptions are routed and processed by the Itron DataSubscriberService, Itron EventSubscriberService, and Itron ExceptionSubscriberService inbound web services.
  - **DataArrived Output Configuration**: Defines how payloads are routed and processed by the Itron DataSubscriberService inbound web service.

Copyright © 2026, Oracle and/or its affiliates.

        - **Save to Staging**: A flag that indicates if payloads routed and processed by the Itron DataSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where payloads are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where payloads are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming payloads to the staging area. By default, the D8-DEFFNGEN script is used.
    - **EventsArrived Output Configuration**: Defines how event payloads are routed and processed by the Itron EventSubscriberService inbound web service.
        - **Save to Staging**: A flag that indicates if event payloads routed and processed by the Itron EventSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming event payloads to the staging area. By default, the D8-DEFFNGEN script is used.
    - **ExceptionsArrived Output Configuration**: Defines how exceptions are routed and processed by the Itron ExceptionSubscriberService inbound web service
        - **Save to Staging**: A flag that indicates if exceptions routed and processed by the Itron ExceptionSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where exceptions are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where exceptions are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving exceptions to the staging area. By default, the D8-DEFFNGEN script is used.

Itron Field Collection System Master Configuration

The “FCS Configuration” master configuration defines details used by the Itron FCS payload processing, including the following:

- **File Header Information**: XML tags used in the file header for export files, including XML Declaration, Import, and Workset tags.
- **FCS Mapping Information**: Details regarding mapping of data from FCS to Smart Grid Gateway

Copyright © 2026, Oracle and/or its affiliates.

- **General Configuration**: Characteristic types used to define values for things such as Special Codes, Transport Modes, Tamper Counts, and so on, as well as the Residential Service Point Type and Rate Code used with FCS customers.
- **Walk-by Read Method**: Read methods used when collecting FCS meter reads, such as "Manual" or "Mobile Meter Read".
- **UOM Mapping to Standard UOM Mapping**: Mappings between FCS Text Prompt ("TextPrompt") and combinations of standard units of measure (UOM) and time of use periods (TOU).
- **Read Method Mapping Information**: Mappings between FCS read methods and standard read methods defined in the system.

Landis+Gyr Master Configuration

The “L+G – Master Config” master configuration defines details used by the Landis+Gyr commands and payload processing, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations**
  - **CIM 1.0 Events Configuration**: Used to define settings for the CIM 1.0 AMIEvent interface
    - **Save to Staging Area**: Flag that indicate whether to save incoming data to a staging area, such as Oracle Object Storage, or process it directly
    - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
    - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
    - **Filename Generation Script**: A service script used to compose the filename when saving the incoming payload to the staging area. By default, the D8-DEFFNGEN script is used.
  - **CIM 2.0 Events Configuration**: Used to define settings for the CIM 2.0 AMIEvent interface
    - **Save to Staging Area**: Flag that indicate whether to save incoming data to a staging area, such as Oracle Object Storage, or process it directly
    - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.

Copyright © 2026, Oracle and/or its affiliates.

        * **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
        * **Filename Generation Script**: A service script used to compose the filename when saving the incoming payload to the staging area. By default, the D3-DEFFNGEN script is used.
        * **XSL Transformer Template**: The XSL used when writing data to a file system. By default, the L+G CIM 2.0 - EndDeviceEvents Output (XSL D3-CIMEndDvcEvtsOutput.xsl) xsl is used.
        * **File Extension**: The extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

Sensus Master Configuration

The “Sensus Configuration” master configuration defines details used by the Sensus commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

Silver Spring Networks Version Master Configuration

The “SSN – UIQ Version Configuration” master configuration specifies the version of the Silver Spring Networks head end system with which the system is communicating:

- **SSN Version**: Used to define the version of the Silver Spring Networks head end system. Options include:
  - SSN v4.10 (WSDL 2.1)
  - SSN v4.14 (WSDL 2.5.1)
  - SSN v4.4 (WSDL v1.4)
  - SSN v4.7 (WSDL v1.6)
  - SSN v5.0-5.1 (WSDL v2.7)
  - SSN v5.1 (WSDL v2.8)
  - SSN v5.2 (WSDL v2.9)
  - SSN v6.1 (WSDL v2.11)

Copyright © 2026, Oracle and/or its affiliates.

- **SSN - Get Job Status Configuration**: Used to specify the **Retry Limit** and **Wait Duration (in Seconds)** for the Get Job Status requests sent by outbound communications.

Silver Spring Networks MultiSpeak Version Master Configuration

The “SSN – MultiSpeak Master Configuration” master configuration defines details used by the Silver Spring Networks MultiSpeak commands and payload processing, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

Adapter Development Kit Master Configuration

The “ADK Configuration” master configuration defines details used by Adapter Development Kit commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations - Save Locations for Events**: Defines how event payloads are routed and processed.

Copyright © 2026, Oracle and/or its affiliates.

- **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
- **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
- **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming event payloads to the staging area. By default, the D1DEFFNGEN script is used.
- **XSL Transformer Template**: A Managed Content record (of type XSLTC) used to transform event data
- **File Extension**: The extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

Adapter Development Kit MultiSpeak Master Configuration

The “ADK MultiSpeak Master Configuration” master configuration defines details used by Adapter Development Kit MultiSpeak commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

Native Implementation Configuration Checklists

This section provides an overview of the steps involved in configuring Native implementations of Oracle Utilities Smart Grid Gateway adapters, including Payload Processing and Smart Meter Command Processing.

Payload Processing Checklist

1. Create Keys and Key Rings (Cloud Only). See Creating Key Rings and Pairs for more information.
2. Configure Object Storage Locations (Cloud Only). See Creating Object Storage Locations for more information.
3. Configure File Storage Extendable Lookup Values. See Creating File Storage Extendable Lookup Values for more information.

Copyright © 2026, Oracle and/or its affiliates.

4. Configure a Head End System to represent the Head End System. See Creating Head End Systems for more information.
5. Configure Processing Methods for the Head End System appropriate for payload processing, including.
    - Initial Measurement Creation
    - Device Event Mapping
    - UOM Translation

    > **Note**
    >
    > These processing methods rely on extendable lookups. See Configuring Adapter Extendable Lookups for more information.
6. Configure SGG Payload Processing Extendable Lookup Values. See Creating SGG Payload Processing Extendable Lookup Values for more information.
7. Configure the adapter's Master Configuration (if appropriate). See Master Configurations for more information.
8. Configure Other Extendable Lookup Values. See Configuring Adapter Extendable Lookups for more information.
9. Configure Batch Controls. See Creating Payload Processing Batch Controls for more information.

Smart Meter Command Processing Checklist

1. Configure Activity Types for each command you wish to support in your implementation. Smart meter commands are activities in the system. For each command you plan to support, confirm that you have a corresponding Activity Type configured.
2. Configure Communication Types appropriate for each command you wish to support in your implementation. Each communication sent to the head end system as part of a command request requires a communication type. See the following sections for more information about the specific communications used with each command:
    - Itron OpenWay Command Flows
    - Landis+Gyr Command Flows
    - Sensus Command Flows
    - Silver Spring Networks Command Flows

    > **Note**
    >
    > Note: Most Adapters use outbound communications only.
3. Configure Message Senders for each command you wish to support in your implementation. See Sending Outbound Communications and Creating Message Senders for more information.
4. Configure Outbound Message Types each command you wish to support in your implementation. See Sending Outbound Communications and Creating Outbound Message Types for more information.
5. Configure an External System to represent the Head End System.

Copyright © 2026, Oracle and/or its affiliates.

a. Configure Message Senders/Outbound Message Types for each command you wish to support in your implementation See Sending Outbound Communications and Creating an External System for more information.

6. Configure a Head End System to represent the Head End System. Head End Systems reference the corresponding External System. See Creating Head End Systems for more information.
7. Configure Processing Methods for the Head End System appropriate for each command you wish to support in your implementation. See Configuring Processing Methods for more information.
8. Configure Inbound Web Services as appropriate for the Head End System See Receiving Inbound Communications and Inbound Web Services for more information.
9. Configure the adapter's Master Configuration (if appropriate) See Master Configurations for more information.
10. Configure Extendable Lookups See Configuring Adapter Extendable Lookups for more information.

Configuration Details Reference

Refer to Knowledge Base article KB523753 on **My Oracle Support** for a spreadsheet containing configuration details for the following Adapters:

- Itron OpenWay
- Silver Spring Networks
- Landis+Gyr
- MV90 (Payload Processing Only)

This spreadsheet contains lists of configuration objects used by each adapter, including:

- Activity Types
- Communication Types
- Message Senders
- Head End System
- Outbound Message Type
- External System
- Master Configurations (as applicable)
- Extendable Lookups
- Batch Controls

Defining Settlement Options

- Settlement Configuration Overview
- Settlement Subscription Types
- Settlement Transaction Exception Types
- Settlement Units

Copyright © 2026, Oracle and/or its affiliates.

- Settlement Calculations
- Settlement Item Types
- Settlement Item Quantity Types
- Market Contract Types
- Market Products
- Settlement Module Configuration

Settlement Configuration Overview

This section provides information about set up and configuration of administration data used by calculations and processes provided with Oracle Utilities Market Settlements Management.

Settlement Subscription Types

- Understanding Settlement Subscription Types
- Configuring Settlement Subscription Types

Understanding Settlement Subscription Types

Settlement subscription types are collections of properties defining a class of settlement subscriptions. Settlement subscription types also control valid values for various attributes of settlement subscriptions.

Settlement subscription types are defined by:

- **Subscription Business Object**: The business object used for settlement subscriptions based on settlement subscriptions of this type
- **Transaction Business Object**: The business object used for settlement transactions based on settlement subscriptions of this type
- **Default Recipient**: The recipient that will be defaulted for settlement subscriptions of this type. A settlement subscription's recipient dictates the system to which results of settlement transactions are sent.
- **Subscription Category**: Specifies the category of settlement subscriptions based on settlement subscriptions of this type, such as "Initial", "Final", and so on. Valid categories are defined in the SUBSCRIPTION_CAT_FLG lookup
- **Transaction Retention Mode**: Defines how settlement transactions based on settlement subscriptions of this type are retained. "Keep All" indicates that all settlement transactions are retained when a replacement transaction is created (older transactions are sent to a status of "Superseded"). "Keep One Record per Period" indicates that only a single settlement transaction is retained for a time period (older transactions are deleted when the new one is completed).
- **Processing Timetable Type**: A processing timetable type used when creating settlement transactions based on settlement subscriptions of this type via batch processing.
- **Valid Settlement Calculation Groups**: A list of valid settlement calculation groups for settlement subscriptions of this type
- **Valid Recipients**: A list of valid recipients for settlement subscriptions of this type

Copyright © 2026, Oracle and/or its affiliates.

- **Communication**: The outbound communication type used to create of outbound communications when sending the results of settlement transactions based on settlement subscriptions of this type

Refer to Processing Timetable Types for more information about setting up processing timetable types.

Refer to Configuring Settlement Units for more information about setting up settlement units and their respective dimension combinations, factors, and factor values.

For a deeper functional understanding, refer to the About Settlement Subscriptions or About Settlement Calculation sections .

Configuring Settlement Subscription Types

Refer to Understanding Settlement Subscription Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Subscription Type List**: This zone lists all usage subscription type records. Broadcast a record to display the details of the selected record.
- **Settlement Subscription Type**: This zone displays details for the selected usage subscription type.

The settlement subscription type defines a number of lists for valid objects that can be used in conjunction with the overall usage calculation process:

- Valid Settlement Calculation Groups
- Valid Recipients

Settlement subscription types also define the outbound communication type used when sending the results of settlement transactions based on settlement subscriptions of this type.

Settlement Transaction Exception Types

- Understanding Settlement Transaction Exception Types
- Configuring Settlement Transaction Exception Types

Understanding Settlement Transaction Exception Types

Settlement Transaction Exception Types define the groupings of exceptions for a settlement transaction based on their functional similarity. This provides a way to define Settlement Transaction Exceptions in a distinct enough way to understand the root issue that was generated from the usage calculation rule.

For a deeper functional understanding of Settlement Calculation, refer to the About Settlement Calculations section.

Configuring Settlement Transaction Exception Types

Copyright © 2026, Oracle and/or its affiliates.

This portal is used to display and maintain a Settlement Transaction Exception Type.

Refer to Understanding Settlement Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Transaction Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Transaction Exception Type List**: displays all of the Settlement Transaction Exception Types so you can choose the one you want to display in more detail
- **Settlement Transaction Exception Type**: shows the specific configuration for the selected Settlement Transaction Exception Type

Settlement Units

- Understanding Settlement Units
- Configuring Settlement Units

Understanding Settlement Units

Settlement units define the “lowest common denominator set” of aggregation dimensions. For example, if aggregations are performed based on Rate Class, Strata, Procurement Group, and Supplier, the following dimensions would form the “lowest common denominator set” of these dimensions (since these are shared by different suppliers):

- Rate Class
- Strata
- Procurement Group

Settlement units define factor values based on unique combinations of dimension values. Different factor values can be defined for each unique combination of a settlement unit’s dimensions. For example, a Rate Class, Strata, and Procurement Group settlement unit could have different factor values for each unique combination of those dimensions. These multi-variable factor values can be retrieved by value derivation algorithms when performing aggregation, estimation, and forecasting calculations based on the same dimensions as defined for the settlement unit.

For example, an aggregation measuring component that aggregates data based on a specific combination of Rate Class, Strata, and Procurement Group could retrieve a “loss” factor value from a Rate Class, Strata, and Procurement Group settlement unit based on the same combination of dimensions. See Applying Losses for an example of how settlement units can be referenced in aggregation calculations.

Aggregation processing also uses settlement units to generate counts of accounts for forecasting based on combinations of these dimensions.

Settlement units are defined by:

- **Dimensions**: The attributes by which aggregated data should be analyzed
- **Dimension Combinations**: Unique combinations of the settlement unit’s dimensions (defined in the **Settlement Unit Dimension Combination** zone)

Copyright © 2026, Oracle and/or its affiliates.

- **Valid Factors**: Valid multi-variable factors (defined in the **Settlement Unit Valid Factors** zone)

Refer to Configuring Settlement Units for more information about setting up settlement units and their respective dimension combinations, factors, and factor values.

Configuring Settlement Units

Refer to Understanding Settlement Units for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Unit**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Unit**: This zone displays details for the settlement unit, including configured dimensions.
- **Settlement Unit Dimension Combinations**: This zone displays combinations of the settlement unit's dimensions, based on the dimensions configured in the **Settlement Unit** zone. Click **Add Dimension Combination** in the zone's title bar to add a new combination.
- **Settlement Unit Valid Factors**: This zone displays valid multi-variable factors for the settlement unit. Click **Add Factor** in the zone's title bar to add a new factor. Values for these factors are defined on the **Multi-Variable Factor** tab.

The following zones appear as part of the portal's **Dimension Combinations** tab page:

- **Settlement Unit Dimension Combinations**: This zone displays a list of dimension combinations for the settlement unit. Broadcast a record to display the details of the selected record. Click **Add Dimension Combination** in the zone's title bar to add a new combination.
- **Dimension Combination Values**: This zone displays factor values for each multi-variable factor defined for the selected dimension combination. You can use this zone to add and/or edit factor values.

The following zones appear as part of the portal's **Multi-Variable Factor** tab page:

- **Multi-Variable Factor List**: This zone displays a list of multi-variable factors for the settlement unit. Broadcast a record to display the details of the selected record.
- **Multi-Variable Factor Value List**: This zone displays factor values for each dimension combination for the selected multi-variable factor. You can use this zone to add and/or edit factor values.

Settlement Calculations

- Settlement Calculation Groups
- Settlement Calculation Rules

Settlement Calculation Groups

- Understanding Settlement Calculation Groups

Copyright © 2026, Oracle and/or its affiliates.

- Understanding Bulk Settlement Calculation Groups
- Configuring Settlement Calculation Groups
- Configuring Bulk Settlement Calculation Groups

Understanding Settlement Calculation Groups

Settlement calculation groups are collections of settlement calculation rules that are used to perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). During the settlement transaction process, the system executes the settlement calculation rules defined in the settlement calculation group referenced on the settlement subscription. The rules within a settlement calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Settlement calculation groups are associated with specific settlement subscriptions and settlement subscriptions types (or both). When assigned to settlement subscriptions, settlement calculation groups contain the settlement calculation rules used to perform the appropriate calculations. Settlement calculation groups associated with settlement subscription types are those groups considered valid for settlement subscriptions of that type.

Settlement calculation groups can also be referenced by the Execute Usage Calculation Group calculation rule.

Understanding Bulk Settlement Calculation Groups

Bulk settlement calculation groups are calculation groups intended for use with high volume calculations, such as UFE allocation.

Some specific differences between bulk settlement calculation groups and standard settlement calculation groups include:

- Bulk settlement calculation groups reference only a single settlement calculation rule.
- Execution of bulk settlement calculation groups is performed by batch execution only, typically within a set of calculations performed during settlement transaction processing.
- Bulk settlement calculation groups use the “Settlement Bulk Calculation Group” business object (D1–SettlementBulkCalcGroup).
- Bulk settlement calculation groups have a **Calculation Group Class** of “Settlement Bulk Calculation”.

Configuring Settlement Calculation Groups

This portal is used to display and maintain a settlement calculation group.

Refer to Understanding Settlement Calculation Groups for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Calculation Group**: This zone displays basic information about a settlement calculation group.

Copyright © 2026, Oracle and/or its affiliates.

- **Settlement Calculation Rules List**: This zone lists the settlement calculation rules belonging to the group. Not applicable to bulk calculation groups.
- **Referencing Settlement Calculation Rules List**: This zone lists the settlement calculation rules that reference the group. Not applicable to bulk calculation groups.
- **Referencing Settlement Subscription Type List**: This zone lists the settlement subscription types that reference the group.
- **Referencing Settlement Subscription List**: This zone lists the settlement subscriptions that reference the group.

Configuring Bulk Settlement Calculation Groups

The **Settlement Calculation Group** portal is used to display and maintain bulk settlement calculation groups.

Refer to Understanding Bulk Settlement Calculation Groups for more information about bulk settlement calculation groups and Configuring Settlement Calculation Groups for more information about creating and maintaining bulk settlement calculation groups.

The following apply when creating new bulk settlement calculation groups:

- Select “Settlement Bulk Calculation Group” from the **Settlement Calculation Group Business Object** drop-down list.
- Use a naming convention of some sort to identify bulk settlement calculation groups (the **Settlement Calculation Group** portal does not provide a way to search for groups based on **Calculation Group Class**). For instance, consider including “bulk” or “blk" in the **Description** of bulk settlement calculation groups.

Settlement Calculation Rules

- Understanding Settlement Calculation Rules
- Configuring Settlement Calculation Rules

Understanding Settlement Calculation Rules

Settlement calculation rules are standard and custom rules that perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). Settlement calculation rules are created for a specific settlement calculation group. For example, if you were configuring two settlement calculation groups and both included a specific usage calculation rule, you would need to create two instances of the settlement calculation rule, one for each group.

On almost every settlement calculation rule, the failure of the rule results in a settlement transaction exception and the Usage Transaction Exception Type for the failure can be configured on the rule. These usage transaction exception types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the settlement transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.

Copyright © 2026, Oracle and/or its affiliates.

- **Issues**: Used to report a problem that will prevent the settlement transaction from being sent. Multiple "issue exceptions" can be created during settlement transaction processing. If at least one issue exists after all rules have been applied, the settlement transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the settlement calculation process to stop and the settlement transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop for a settlement transaction). This should be used for cases where manual override / approval isn't accurate.

For a deeper functional understanding of settlement calculation, refer to the About Settlement Calculation section.

Configuring Settlement Calculation Rules

This portal is used to display and maintain a settlement calculation rule.

Refer to Understanding Settlement Calculation Rules, Understanding Settlement Calculation Groups, and Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Calculation Rule**: This zone displays details for the selected settlement calculation rule, including parameters used when executing the rule
- **Eligibility Criteria List**: Lists the eligibility criteria defined for the rule, if applicable.

Calculation Settlement Calculation Rules

The following is a list of the calculation settlement calculation rules provided as part of the base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Settlement Calculation Rule Name	Applicable Data Type(s)	Purpose
Apply Math (Interval Data)	Interval	This rule is used to perform calculations on interval data and stores the results in the settlement transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions). This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a settlement subscription. This rule can also multiply total usage by a factor using a custom formula.

Copyright © 2026, Oracle and/or its affiliates.

Settlement Calculation Rule Name	Applicable Data Type(s)	Purpose
Array Math	Interval, arrays, and calculated quantities	This rule is used to perform settlement calculations on arrays and interval data and it stores the results in the settlement transaction's service quantities.
Market Award Allocation	Interval	This rule is used to perform settlement calculations to allocate load based on market awards, based on appropriate methods for dividing load out to wholesale contracts.
Measuring Component Set Calculation	Interval and calculated quantities	This rule is used to perform UFE calculations.
Vector and Service Quantity Math	Interval	This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations. Typical uses: Perform math using interval data. For example, take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc. Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values). Support math functions: sin, cos, square root, etc. Store derived curves in memory that can be used in subsequent calculations Please note, this rule is not as efficient as other rules.

Decision-Making Settlement Calculation Rules

There are settlement calculation rules delivered as part of the base product that help with decision-making when executing the settlement calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.

Copyright © 2026, Oracle and/or its affiliates.

Settlement Calculation Rule Name	Purpose
Execute Usage Calculation Group	This rule performs a call to execute a separate settlement calculation group which includes execution of all settlement calculation rules within that group.
Exception Handler	This rule is used to terminate processing if exception count criteria specified in the rule is met.

Advanced Aside: Using Factors For Variables

A situation common in some implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this type can differ based on many different types of criteria, such as the location of the service point or other characteristics. This type of calculation can be implemented as a settlement calculation rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of "Number," and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).

Copyright © 2026, Oracle and/or its affiliates.

- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.

Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value. At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value. Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute**: you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and a third value if the amount conserved is greater than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.
- Calculation Rules

Calculation Rules

- Array Math
- Market Award Allocation
- Measuring Component Set Calculation

Array Math

This rule is used to perform settlement calculations on arrays and interval data and it stores the results in the settlement transaction's service quantities. Arrays can be derived from either measuring component sets or calculated quantities. Interval data vectors can be derived from measuring component sets, individual measuring components, or calculated quantities. A variety of options are available on this rule that include defining variables to use for arrays and interval data vectors, the specific calculations to perform (leveraging math functions and expressions), and how to save the results.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ARRAYMATH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-ArrayMath

Copyright © 2026, Oracle and/or its affiliates.

Market Award Allocation

This rule is used to perform settlement calculations to allocate load based on market awards, based on appropriate methods for dividing load out to wholesale contracts. This rule takes in a vector from the settlement transaction and divides the load to be allocated to various market awards for a configured market product set. The allocated amounts should be tied to the contract connected to the market award and then saved to the settlement transaction as a new array.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-AWARDALOC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-MarketAwardAllocation

Measuring Component Set Calculation

This rule is used to perform settlement UFE calculations.

This rule executes a batch program (D1-MCSCA, defined as a parameter on the Execute MC Set Calculation-Apply Rule algorithm), which in turn executes the Perform MC Set Calculation algorithm to perform the calculation. The rule configuration includes definition of variables and the equation used for UFE calculations. Variables can be based on specified measuring component sets, aggregation measuring components, or calculated quantities.

This rule should be used ONLY with Bulk Settlement Calculation Groups.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-EXEMCSCAL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

Business Object

D2-MCSetCalculation

Settlement Item Types

- Understanding Settlement Item Types
- Configuring Settlement Item Types

Copyright © 2026, Oracle and/or its affiliates.

Understanding Settlement Item Types

Settlement Item Types define specific types of end-customer settlement accounts.

Settlement item types define the following attributes:

- **Settlement Item Business Object**: The business object to use create settlement items of this type.
- **Settlement Billed Usage Business Object**: The business object to use to capture billed usage for a settlement account.
- **Valid Service Point Types**: Valid service point types for the settlement item type.

> **Note**
>
> Settlement item types are similar to usage subscription types used in Meter Data Management, and are based on the Usage Subscription Type maintenance object (D1-USTYPE).

Configuring Settlement Item Types

Refer to Understanding Settlement Item Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Item Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Type List**: This zone lists all settlement item types. Broadcast a record to display the details of the selected record.
- **Settlement Item Type**: This zone displays details for the selected settlement item type.

Settlement Item Quantity Types

- Understanding Settlement Item Quantity Types
- Configuring Settlement Item Quantity Types

Understanding Settlement Item Quantity Types

Settlement item quantity types define types of quantities that can be stored for a settlement item. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class.

Settlement item quantity types use the following parameters:

- **Service Type**: The service type (electric, gas, water, etc.) for settlement item quantities of this type.
- **Settlement Item Quantity Business Object**: The business object used for settlement item quantities of this type.

Copyright © 2026, Oracle and/or its affiliates.

- **Settlement Item Quantity Identifiers**: Value identifiers related to the current settlement item quantity type (used to provide shorthand descriptions of the various types of values measured by settlement item quantities of this type).

Configuring Settlement Item Quantity Types

You use the **Settlement Item Quantity Type** portal to display and maintain settlement item quantity types.

Refer to Understanding Settlement Item Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Item Quantity Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Quantity Type List**: This zone lists all settlement item quantity type records. Broadcast a record to display the details of the selected record.
- **Settlement Item Quantity Type**: This zone displays details for the selected settlement item quantity type.

Market Contract Types

- Understanding Market Contract Types
- Configuring Market Contract Types

Understanding Market Contract Types

Market Contact Types define specific types of market contracts used in settlement processing.

Market contract types define the following:

- The market associated with market contracts of this type
- The pricing category for market contracts of this type
- Whether market contracts of this type have Unique Contract Buyers. When set to "Yes", market contracts of this type are validated to ensure that each buyer only has one valid contract at a point in time for the market contract type.

Configuring Market Contract Types

Refer to Understanding Market Contract Types for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Contract Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Contract Type List**: This zone lists all market contract type records. Broadcast a record to display the details of the selected record.
- **Market Contract Type**: This zone displays details for the selected market contract type.

Copyright © 2026, Oracle and/or its affiliates.

Market Products

- Market Product Sets
- Market Product Types

Market Product Sets

- Understanding Market Product Sets
- Configuring Market Product Sets

Understanding Market Product Sets

Market Product Sets define the highest level grouping for a set of products that will be processed together. Market Product Sets are used by the Market Award Allocation settlement calculation rule as a way to gather a set of products and their related awards for division of load into an array of interval curves / vectors.

Market product sets define the following:

- The market associated with the product set
- The service type (electric, gas, etc.) of the product set

Configuring Market Product Sets

Refer to Understanding Market Product Sets for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Product Set**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product Set**: This zone displays details for the selected market product set.

Market Product Types

- Understanding Market Product Types
- Configuring Market Product Types

Understanding Market Product Types

Market Product Types define types of products used in settlement processing.

There are two primary types of market product types, each based on a specific business object:

Copyright © 2026, Oracle and/or its affiliates.

- **Market Product Type**: Used for market product types whose award allocation is based on the configuration of the Market Award Allocation settlement calculation rule.
- **Formula-Based Market Type**: Used for market product types with more complex award allocation. This business object provides the ability to configure complex conditional calculations that take in both interval and scalar variables.

Market product types are defined by the following:

- **Division**: The division associated with the market product type.
- **Number of Decimals**: The number of decimal values used for market award quantities based on market products of this type.
- **Product Business Object**: The market product business object used for market products of this type.
- **Award Business Object**: The market award business object used for market awards based on market products of this type.
- **Calculation Details**: Applicable to Formula-Based Product Types only. Details used in performing formula-based calculations for market products of this type. Details include:
  - **Calculation Pre-Processing**: One or more algorithms to be executed prior to performing calculations for products of this type. These can include validation algorithms used to ensure the data being used by the calculations is valid and algorithms that retrieve, calculate, and store values used later in the calculation.
  - **Calculation Inputs**: Input variables (including Fixed Variables, Award Adhoc Characteristics, Factor Numbers, and Calculated Quantities) and Input Vectors (Fixed Vector Variables, Calculated Quantities, and Factor Profiles) that will be used in the calculations performed for products of this type.
  - **Calculations**: One or more calculations (based on the Calculation Inputs) to be performed for products of this type. Calculations can be based on Simple or Conditional formulas.
  - **Calculation Post-Processing**: Details of how the results of the calculations are saved and used in later processing. Options include:
    * **Save Results to Calculated Quantity**: Defines how calculation results are saved as service quantities on the Settlement Transaction that initiated the market product calculation and, if applicable, assigned to market contracts.
    * **Save Results to Characteristics**: Defines how calculation results are saved as characteristic values on the Settlement Transaction that initiated the market product calculation.

Configuring Market Product Types

Refer to Understanding Market Product Types for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Product Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product Type List**: This zone lists all market product type records. Broadcast a record to display the details of the selected record.
- **Market Product Type**: This zone displays details for the selected market product type.

Copyright © 2026, Oracle and/or its affiliates.

Settlement Module Configuration

Oracle Utilities Market Settlements Management shares its application environment with other Oracle Utilities meter solution products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. The default installation of any of these products provides access to all menu items and functionality provided with these products.

To configure the system such that only menu items and functionality used by Oracle Utilities Market Settlements Management is available, application users should be assigned to the following set of user groups:

- Meter Stack Common Services
- Meter Stack Common Batch Services*
- Settlement Services
- Settlement Batch Services*

*These user groups should be restricted to only those users who will be executing batch processing.

Refer to Defining Users in the *Oracle Utilities Application Framework Administrative User's Guide* for more information about setting up users.

> **Note**
>
> There is a Feature Configuration option that disables menu items and functionality used by Oracle Utilities Market Settlements Management even after assigning users to the above user groups. To check if this option is enabled, navigate to the **Feature Configuration** portal by clicking the **Menu** icon, then selecting **Admin Menu**, then **General**, and then **Feature Configuration**. In the **Feature Configuration Search** zone, select "Module Configuration" from the **Feature Type** dropdown list and click **Search**. Click the "Module Configuration" **Description** link and check to see if there is a "Turned Off" option defined in the **Options** zone with a value of "SE". If so, delete that option. Note you may have to flush the server cache (see **Caching Overview** in the *Framework Administrative User Guide*) after after deleting this option before the menu items and functionality are available.

Copyright © 2026, Oracle and/or its affiliates.

Part V
Oracle Utilities Cloud Service Foundation

Welcome to the Oracle Utilities Cloud Service Foundation Administrative User Guide. This document describes how to administer features associated with Oracle Utilities Cloud Service Foundation.

> **Note**
>
> This document does not apply to all implementations. Contact your system administrator to confirm if your implementation includes Oracle Utilities Cloud Service Foundation.

- Process Automation Tool
- Data Conversion Support for Cloud Implementations
- Batch Scheduler for Cloud Implementations
- Batch Queues Overview
- Batch Alerts
- ILM Dashboard for Cloud Implementations
- Oracle Cloud Infrastructure Identity and Access Management Integration
- Oracle Cloud Object Storage
- Maintaining Customer Code

Process Automation Tool

The Process Automation Tool allows customers and implementers to orchestrate and automate a set of infrastructure related multi-step processes. The tool provides the ability to monitor the progress of such process and each of the steps. The tool also supports user actions at the process and step level.

The process automation tool is made out of the following system entities:

- Infrastructure Process - a Business Object of the Service Task MO.
- Infrastructure Process Step - a Business Object of the Service Task MO.
- Infrastructure Process Type - a Business Object of the Service Task Type MO.
- Infrastructure Process Step Type - a Business Object of the Service Task Type MO.

An Infrastructure Process references a type that includes the steps that are a part of that process. The steps are executed one by one according to their sequence as long as the previous step ended successfully. When a step in a process fails, the process will stop and wait for a user to take further actions.

Copyright © 2026, Oracle and/or its affiliates.

An Infrastructure Process can span up to two product environments. This means that it can invoke and monitor steps in two systems. This, for example, allows it to orchestrate a configuration migration (using the Content Migration Assistant, described in your cloud service's *Administrative User Guide*) from one product environment to another. Such a migration can include steps to create the export data on the **source** environment, move the export file to the **target** environment and execute all the import tasks on the target environment, all as a single Infrastructure Process. In this case the Infrastructure Process will be created on the **target** environment and will pull the configuration from the **source** environment.

> **Note**
>
> This is done in order to ensure that only application users with sufficient security privileges in any system environment can create Infrastructure Processes that can update that environment.
>
> The rule is that when an Infrastructure Process is created to run processes on two systems, the process will always be created on the **environment with the higher security requirements**. For example, if you need to run processes on the Development and Test environments, the infrastructure process will be created and executed from the Test environment (assuming that the Test environment has higher security requirements than the Development environment).

The steps that make an Infrastructure Process reference a step type that governs their behavior. A comprehensive set of process and step types is provided with the product. This set can be extended by implementers as needed.

- Creating an Infrastructure Process
- Executing an Infrastructure Process
- Monitoring Infrastructure Process Progress
- Process Automation Tool Setup
- Configuring New Infrastructure Processes
- CMA Migration Infrastructure Process Types

Creating an Infrastructure Process

To create a new process, navigate using **Admin > Implementation Tools > Infrastructure Process**. Use the Infrastructure Process Query to search for an existing process. One an Infrastructure Process is selected, you are brought to the maintenance portal to view and maintain the selected record.

The following points provide information about creating a new Infrastructure Process.

Infrastructure Process references an **Infrastructure Process Type**. When creating a new process you need to select the type that the process will be based on.

The **Name** of the process is used for tracking purposes. The name doesn't have to be unique in the system.

The **Process Steps** list is read-only and represents the steps that were configured on the Infrastructure Process Type.

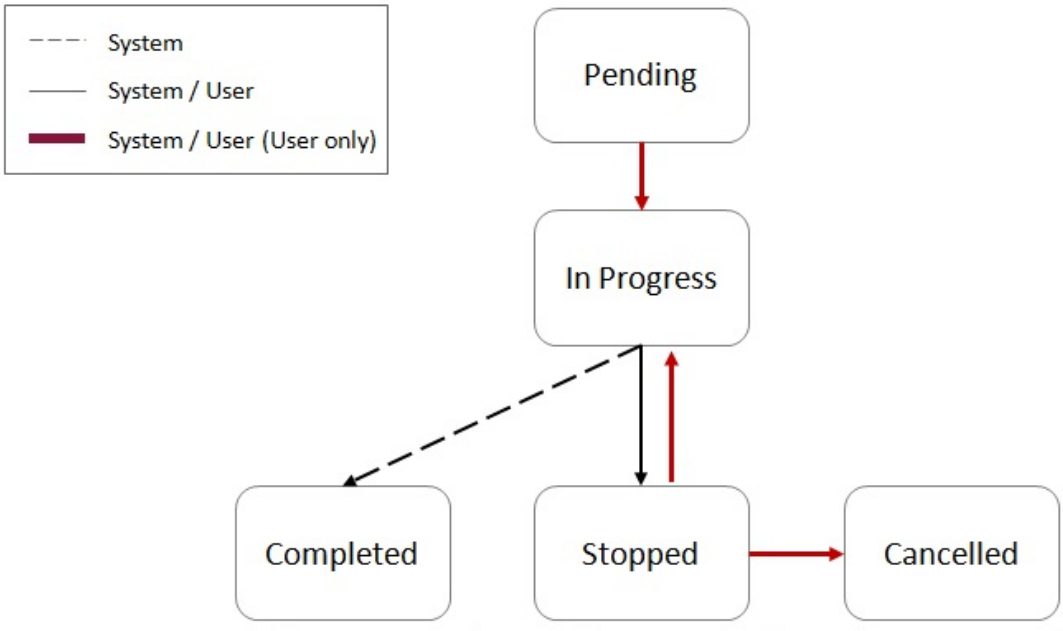
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Changing information on the Infrastructure Process Type, including the steps, will not affect existing Infrastructure Process records.

The **Process Details** section represents additional information that is required according to the corresponding process type definition.

Executing an Infrastructure Process

Infrastructure Process Lifecycle



New processes are created in a Pending status. While in this status the Infrastructure Process record can be modified, deleted or duplicated.

> **Note**
>
> Once an Infrastructure Process record is created, its type cannot be changed. To change the type, you can delete the process that was created with the wrong type (if it is still in Pending status) and create a new process with the correct type.

Use the **Start** action, on the maintenance portal to begin the process execution. The process status will change to **In Progress**.

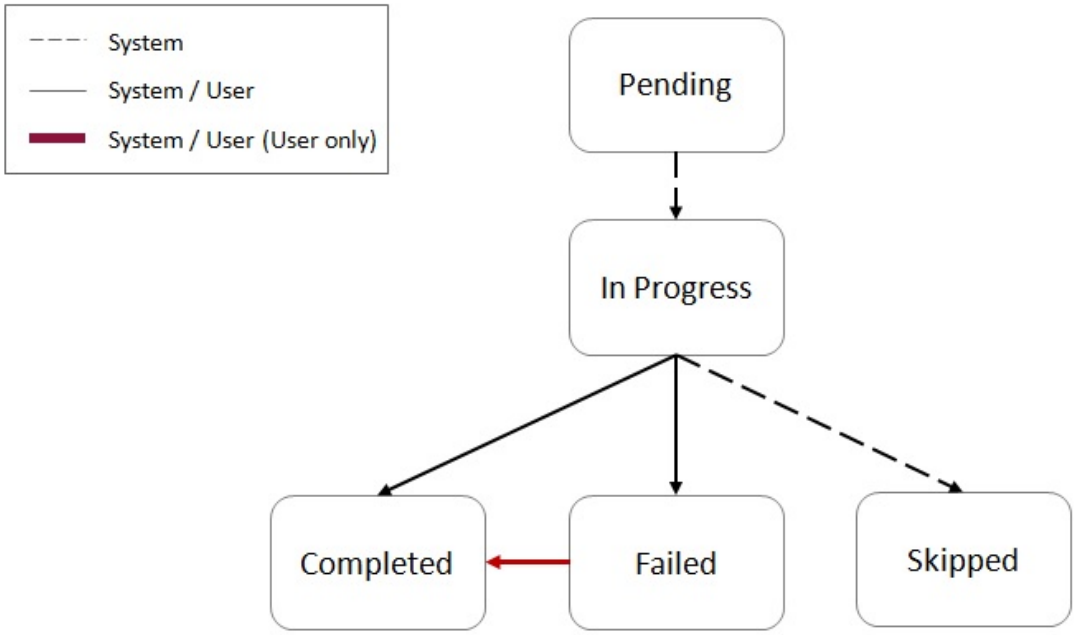
Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> While the Infrastructure Process is in **Pending** status, the steps will not have a status or any available actions. When the process transitions to **In Progress** status the process step records are created. From that point on they have a status and actions according to the step that and status.

While the process is **In Progress** the steps of the process will execute one by one according to their sequence. The process can be stopped using the **Stop** action. In this case the current step that is in progress will continue but the next step will not be executed. If any of the steps fail, the process will also transition into **Stopped** status and will not execute the next step.

When the process is in **Stopped** status it can be resumed by selecting the **Resume** action, or it can be cancelled by selecting the **Cancel** action.

Infrastructure Process Step Lifecycle



A process steps is created in **Pending** status and is transitioned to **In Progress** by the process logic when its turn to execute arrives.

When a step is **In Progress**, the following transitions are possible:

- Transition the step into Completed status upon successful completion of the step's work or by selecting the **Complete** action.
- Transition the step into **Failed** status when an error was detected during the step's work or by selecting the **Fail** action.
- Steps may include logic that will transition the step into **Skipped** status if that logic determines that the step should not be executed.

When the step is In Progress, selecting the **Job Status** action will display information about the job running status (if it is a batch job type of step).

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Manually changing the step status to **Completed** or **Failed** while the step is In Progress should be carefully considered. This should only be done when step stays **In Progress** status for a long time and upon investigation it is clear that some manual intervention is required.

When a step is in **Failed** status it is still possible to manually transition the step into Completed status.

> **Note**
>
> Transitioning the step from **Failed** to **Completed** status should only be done if it is clear that the step was able to successfully accomplish its task. This will likely be established after some investigation.

Monitoring Infrastructure Process Progress

- Infrastructure Process Control
- Monitoring a Running Infrastructure Process
- Dealing with Exceptions

Infrastructure Process Control

The Infrastructure Process Control portal provides an overview of Infrastructure Processes that are active (i.e. in **Pending** or **In Progress** status) or processes that did not complete (i.e. in **Stopped** status) within the last X days.

Navigate to Infrastructure Process Control portal, using **Admin > Implementation Tools > Infrastructure Process Control**.

Available actions on this portal are:

- **New Process** - creates a new Infrastructure Process
- **Environment Setup** - directs the user to the master configuration related to the process automation tool

Select any of the processes from the portal lists for more information.

Monitoring a Running Infrastructure Process

When a process is running (in **In Progress** status), you can monitor its progress by refreshing the page. To configure automatic page refresh, refer to the User Portal Preferences for this portal.

The Infrastructure Process maintenance portal provides information about the status of the overall process as well as each of the steps.

Copyright © 2026, Oracle and/or its affiliates.

You can also view the details of each individual step including their log for more information or to investigate exceptions.

Dealing with Exceptions

Infrastructure Process steps can be divided into two main categories:

- Steps that start a batch process (job)
- Steps that are not batch process related

In addition, steps can be local or remote. Local steps execute on the same product environment that the process is executing while remote steps are executed on a different product environment.

- Batch Job Step Exceptions
- Non-Batch Step Exceptions

Batch Job Step Exceptions

Batch job steps start a batch job either locally or on a different product environment.

Batch Job Step in Failed Status as a Result of a Batch Job Error

When a batch process (local or remote) ends with an error, the process step is automatically transitioned to **Failed** status.

In this case, you should:

1. Ascertain the cause of the job failure (in the local or remote environment).
2. If the issue is temporary or can be fixed, fix it and then rerun the batch job manually using the Batch Job Submission page (in the local or remote environment). If the resubmitted job ended successfully, you can transition the process step to the **Completed** status and use the **Resume** action at the process level to continue the process execution.

If the issue cannot be resolved, further investigation is required. In this case the process should remain in **Stopped** status.

Batch Job Step in In Progress Status While Batch Job Ended With an Error

When an Infrastructure Process Step that is associated with a batch job stays in **In Progress** status while the batch job ended with an error, further investigation is required. One possible cause for that situation could be an error that causes the job to fail and at the same time also prevents the job from communicating its status back to the originating process step.

In this case you should manually transition the step into **Failed** status and continue your investigation.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If you decide to stop the Infrastructure Process altogether (as a result of the exception), it is a good idea to mark the batch job that failed "Do Not Attempt Restart" in the Batch Run Control for that batch run. This will ensure that the next time the Infrastructure Processes runs, when it gets to that batch process step it will not attempt to resume the execution that ended with an exception, possibly causing a repeat of the exception.

Batch Job Step in In Progress Status While Batch Job Ended Successfully

When an Infrastructure Process Step that is associated with a batch job stays in **In Progress** status while the batch job ended successfully, further investigation is required. In this case, once you ascertained that the job did actually end successfully, you can manually transition the process step into **Completed** status and use the **Resume** action at the process level to continue its execution.

Non-Batch Step Exceptions

Infrastructure Process Steps that are not associated with batch jobs can experience errors as well and therefore can fail. As a reminder, the logic for these steps can experience local or remote errors. When such a step fails, information about the failure can be found in the process step log.

Process Automation Tool Setup

The process automation tool requires the setup of the following components:

- Master Configuration
- External Systems
- Message Senders

In addition, a security setup is required.

The setup itself is done automatically as part of the environment provisioning process. The information in this section is provided for reference and in case changes in existing setup are required.

- Security Setup
- Master Configuration Setup
- External Systems and Message Senders

Security Setup

Before the process automation tool can be used, the security credentials of the Message Senders, used by the process automation tool, must be updated.

In order to execute the security setup, navigate using **Admin > Implementation Tools > Process Automation Security Setup**.

The setup script will ask you to provide a User ID, a Password and to indicate whether or not to override the existing security setup.

Copyright © 2026, Oracle and/or its affiliates.

Please contact your system or security administrator for more information about what User ID should be used for this setup.

You should select to override existing security setup when you need to update the User ID or the Password on the existing Messages Senders, used by the process automation tool. A common use case will be mandatory periodic password rotation for the user account that was selected for the setup.

> **Note**
>
> At the end of the process, you will be asked to flush all the system cache. This is required in order for the information update to take effect.

Master Configuration Setup

The process automation tool relies on master configuration settings in order to identify all the product environments that are available to it. The Master Configuration is automatically setup as part of the system provisioning process.

This section describes the setup options in case manual changes are needed.

To access the configuration for the process automation tool, navigate using **Admin** > **General** > **Master Configuration** and look for Process Automation Configuration.

The Main section includes the definition of the **Current Environment** code and the **default file extension** for CMA export files.

The Current Environment code is a **unique** code that is given to each product environment.

The Environment List section describes all the existing product environments that the process automation tool need to interact with, for example, in order to run a process that moves integration configuration from one environment to another.

The environment in the list includes the following details:

- Environment: the environment code.
- **Product**: defined in the Process Automation Product extendable lookup.
- **Domain**: defined in the Process Automation Domain extendable lookup.
- **Migration Source**: select "Yes" if you can migrate configuration data from this environment.

  > **Note**
  >
  > You should not allow automated migration of configuration from your development environment directly to your production environment but you would allow migrating from the development to test environments.

- **External System**: reference to the External System definition that will support communication with the row"s environment.

External Systems and Message Senders

Copyright © 2026, Oracle and/or its affiliates.

The process automation tool communicates with other product environments through Outbound Messages.

The automated setup for the process automation tool creates a set of Message Senders and External Systems to support communication with other product environments.

Each product environment which is defined on the Process Automation Configuration has a corresponding unique External System. Each External System contains the details of all the message types that it supports, the message sender for each message type and additional data if required.

Configuring New Infrastructure Processes

The process automation tool configuration is essentially a collection of process types and step types that support the automation of various multi-step processes. A set of Infrastructure Process Types and Infrastructure Process Step Types is provided with the base product. Customers and implementers can create new process and step types.

This section describes the actions required to configure new Infrastructure Process Type and Infrastructure Process Step Type.

- Configuring a New Infrastructure Process Step Type
- Configuring a New Infrastructure Process Type

Configuring a New Infrastructure Process Step Type

In order to configure a new step type, navigate using **Admin > Implementation Tools > Infrastructure Process Step Type**. Use the Infrastructure Process Step Type Query to view existing step types.

> ⚠ **Caution**
>
> Base product step type names start with "K1" and should not be change. Changing these step types can break the process and step types that are provided with the product.

The following points provide information about creating a new Infrastructure Process Step Type.

When creating a new step type, a business object has to be selected first. The Infrastructure Process Step Type BO contains the information necessary for the Infrastructure Process Step that references that step type. For example, a batch process step type BO contains information related to the batch code that should be submitted and the parameters that will be passed to it. You can use existing Infrastructure Process Step Type BOs or create new BOs according to your business requirements.

The **Related Transaction BO** references the Infrastructure Process Step BO. The step BO contains the business logic associated with the step. For example, a batch process step BO is responsible to submit a batch job with to the information from the step type. You can use existing Infrastructure Process Step BOs or create new BOs according to your business requirements.

The **Process Step Type Details** section represents additional information that is required for this specific step type BO.

Copyright © 2026, Oracle and/or its affiliates.

Configuring a New Infrastructure Process Type

In order to configure a new process type, navigate using **Admin > Implementation Tools > Infrastructure Process Type**. Use the Infrastructure Process Step all-in-one portal to view existing process types.

> ⚠ **Caution**
>
> Base product process type names start with "K1" and should not be change. Changing these process types can break the base product process types that are provided with the product.

The following points provide information about creating a new Infrastructure Process Type.

When creating a new process type, a business object has to be selected first. The Infrastructure Process Type BO contains information necessary for the Infrastructure Process that references that process type. You can use existing Infrastructure Process Type BOs or create new BOs according to your business requirements.

The **Related Transaction BO** references the Infrastructure Process BO. The process BO contains the business logic associated with the process. This includes the orchestration of the process steps. The process BO also contains information that is shared by step business logic during the process. For example, a process BO that handles migration of data between product environments can have a buffer that will hold the information pulled from the source environment using one step so that another step can use it to push that to the target environment. You can use existing Infrastructure Process BOs or create new BOs according to your business requirements.

The **Process Type Steps** section includes a collection of step types. The sequence of steps has to be unique and this will be sequence of execution.

> ⚠ **Caution**
>
> When selecting step types for a process type you need to consider the fact that not all step types can be used on all process types. For example, some steps need to get or save information in the process itself so it can be used by other steps of that process. These types of steps can only work with a certain type of process. Please refer to the detailed description in each of the base product provided Infrastructure Process Step Types to see what process types they can be referenced on.

The **Infrastructure Process Type Details** section represents additional information that is required for the specific process type BO.

CMA Migration Infrastructure Process Types

The following Infrastructure Process Types are provided to support the following configuration migration use cases:

- CMA Accelerator Load - import an exported file created by the Content Migration Assistant export process.

Copyright © 2026, Oracle and/or its affiliates.

- CMA Migration - a wholesale or piecemeal configuration migration (using the Content Migration Assistant), exporting configuration from a source environment and importing it to a target environment.
- CMA Accelerator Load (K1-CMA-LOAD)
- CMA Migration (K1-CMA-MIGRATE)

CMA Accelerator Load (K1-CMA-LOAD)

The K1-CMA-LOAD Infrastructure Process Type supports importing of an exported CMA file.

K1-CMA-LOAD is based on K1-CMAMigrationProcessType Infrastructure Process Type BO and uses the K1-CMAMigration as the Infrastructure Process BO.

The CMA Accelerator Load process type includes load an exported CMA and to run all the batch processes that are a part of the CMA import.

> **Note**
>
> This process type does not include a step for a manual user review. It is assumed that when loading an accelerator, there shouldn't be anything to review and the new configuration content is expected to be new for the most part or OK to override existing configuration.

Using the CMA Accelerator Load Infrastructure Process Type

In order to import a CMA export file (e.g. an accelerator), you need to:

- Upload the file to the system designated storage location for CMA uploads. The location is configured on the CMA Master Configuration entry and is automatically defaulted to an Oracle Cloud Object Storage location during the system installation process.
- Create a new infrastructure process with the K1-CMA-LOAD Infrastructure Process Type and provide the following additional details:
  - **Name**.
  - The **Process Steps** list is for reference only.
  - **Export filename**: specify the CMA export file name that was uploaded to the system. You should specify the full name including the file extension name.
  - **User Review Needed**: select "Yes" only if you are expecting the CMA file to include data that already exists in the system and need to be reviewed before changes are applied.
  - **Default Status For Add**, **Default Status For Update** are relevant in case you selected to have a user review. In this case define the default status of migration data items when imported into the target environment.
- Once the process is saved, you can start it by selecting the Start action.

CMA Migration (K1-CMA-MIGRATE)

Copyright © 2026, Oracle and/or its affiliates.

The K1-CMA-MIGRATE Infrastructure Process Type supports a full or partial migration of configuration data, from a remote environment (the source) to the environment that the process is running on (the target).

> **Note**
>
> The configuration migration process is done as a PULL (running on the target environment and pulling data from the source environment) to make sure that users without security privileges to make changes in a product environment "A" won"t be able to run a process from a remote environment "B" that will force changes into "A".

K1-CMA-MIGRATE is based on K1-CMAMigrationProcessType Infrastructure Process Type BO and uses the K1-CMAMigration as the Infrastructure Process BO.

This configuration migration process type includes all the necessary steps to create an export CMA file from a migration request on the source environment, store the file in a shared file location (accessible by the source and target environment processes) and to run all the import batch processes on the target environment to process the file. This process type also includes a step for a manual user review of the imported data if needed.

> **Note**
>
> The shared file location is configured on the Migration Assistant Configuration master configuration entity and is defaulted to an Oracle Cloud Object Storage location during the system installation process.

Using the CMA Migration Infrastructure Process Type

In order to migrate configuration data from a remote environment (the source environment) to the current environment (the target environment) you need to:

- Create a new infrastructure process with the K1-CMA-MIGRATE Infrastructure Process Type and provide the following additional details:
  - **Name**.
  - **From Environment**: select the source environment from which the data will be migrated to the target environment (the current environment). The list of valid environments to choose from is taken from the Process Automation Master Configuration. The target environment is always the current environment - the one that this process is created on.
  - The **Process Steps** list is for reference only.
  - **Migration Request**: specify the CMA Migration Request ID that will be used for the migration process. The migration request should exists on the source environment and will be validated against that environment.
  - **Export filename**: specify the CMA export file name that will be used for the export and import process. You should specify the full name including the file extension name. If you leave this field empty, the system will default the file name according to the migration request ID and the default file extension name.
  - **User Review Needed**: select "Yes" only if you are expecting the CMA file to include data that already exists in the system and need to be reviewed before changes are applied.

Copyright © 2026, Oracle and/or its affiliates.

  - **Default Status For Add**, **Default Status For Update** are relevant in case you selected to have a user review. In this case define the default status of migration data items when imported into the target environment.
- Once the process is saved, you can start it by selecting the Start action.

> **Note**
>
> Before using this Infrastructure Process Type, make sure that the import and export directories for the Migration Assistant Configuration master configuration entry are pointing to the SAME shared location. This is required for a successful completion of this process type.

Data Conversion Support for Cloud Implementations

This chapter provides information and requirements on data conversion support for Oracle Utilities Cloud implementations.

- Data Conversion Overview
- Configuring Conversion Data Upload
- Input Data Files
- Conversion Orchestration
- Security Setup for Conversion
- Conversion Reconciliation Reports

Data Conversion Overview

The ability to convert customer data from legacy systems to OUAF based products is critical for every implementation. The following chapter describes the overall process, tools and considerations that are specific to OUAF based product implementations on Oracle Cloud.

- Conversion Steps
- SQL Loader
- Conversion Artifacts

Conversion Steps

In general data conversion effort in the cloud is divided into the following steps:

- Prepare for conversion
- Create conversion data files
- Upload conversion data files to the cloud
- Process conversion data files
- Process converted data in cloud application

Copyright © 2026, Oracle and/or its affiliates.

- Approve converted data for production mode

> **Note**
>
> It is important to note that the conversion process is a highly iterative process. All the steps above are likely to be done (completely or partially) multiple times during the conversion process.
>
> For example, customer typically start with a small set of data to convert, they fix issues discovered with that data and then re-create new data files and proceed to convert them. This can happen many times, each time with either a different data sample or bigger and bigger data conversion sets until all the required data has gone through the conversion process.

- Prepare for Conversion
- Create Conversion Data Files
- Upload Data Conversion Files to the Cloud
- Process Conversion Data Files
- Process Converted Data in Cloud Application
- Approve Converted Data for Production Mode

Prepare for Conversion

Preparing for conversion will typically include the following actions:

- Identify the OUAF product entities (Maintenance Objects) for which data will be converted. Refer the specific OUAF product documentation for information about converted entities for that product. Typically, each of the product converted entities will have a set of conversion staging tables that will receive the data from your legacy system before they get processed into the system.

  > **Note**
  >
  > Conversion staging schema design may vary between products, please refer to specific product's documentation for more details.

- Evaluate your project's requirements for the Information Lifecycle Management (ILM) and Data Archiving. If your implementation uses ILM, the ILM Date must be provided as part of the legacy data extract for each eligible table.

  > **Note**
  >
  > ILM support may vary between products, please refer to specific product's documentation for more details.

- Identify tables and maintenance objects whose data requires special treatment. That may include high volume tables, extremely large CLOB data, tables with multiple CLOB fields or any other special considerations related to the legacy data being converted.

Copyright © 2026, Oracle and/or its affiliates.

Address the special circumstances by customizing the default configurations:

- Enable conversion for tables that are not marked for conversion in the metadata use the Master Configuration
- If your table(s) contain multiple CLOBs, but you don't plan to supply the legacy data for all of them, create Conversion Task Type for this table(s) and populate **Exclude CLOB Field** list and then add a corresponding entry to Master Configuration's Special Instructions
- If you dealing with high volumes of data, you may want to adjust the load parameters and options for the SQL Loader. In this case, create a **custom Control File Header** (Managed Content), configure a dedicated Conversion Task Type referencing this custom Control File Header and then add a corresponding entry to Master Configuration's Special Instructions. Note that creating the custom Control File Header will result in a control file composed from the custom fragment and the generated fields list.
- If the generated field definitions list in the control file doesn't satisfy your requirements, you may completely override the control file. In this case, create a **custom Control File** (Managed Content, same as above), configure a dedicated Conversion Task Type referencing this custom Override Control File and then add a corresponding entry to Master Configuration's Special Instructions. Make sure your custom Control File contains all the run-time substitution variables that appear in the base Control File Header.

See Conversion Task Types and Conversion Master Configuration chapters for more details.

- Generate the conversion artifacts for the product entities you plan to convert data for. This will include, for example, a set of data specification files that describe the format of data required for each file.
- Explore conversion tasks generated for tables and/or maintenance objects, examine the input data specifications and file lists and determine the expected data formats and input file names.
- Identify your various sources of data and extraction methods.

Create Conversion Data Files

In this step you create the set of data conversion files that were identified for the conversion process. You can choose to generate data files for each staging table or create a smaller set for each converted Maintenance Object.

The data conversion files typically do not include all the data to begin with but start small and grow with every iteration of the conversion process until they reach their maximum size at the end.

> **Note**
>
> When performing data conversion in the cloud, customers don’t have access to the database staging tables. Data scrubbing and transformation activities has to take place before the data files are generated (in each conversion iteration) so the result of these activities is reflected on the data files that are moved to the cloud to be loaded into the staging tables.

Copyright © 2026, Oracle and/or its affiliates.

Upload Data Conversion Files to the Cloud

Since data conversion files are created locally they have to be uploaded to the cloud environment in order to be processed and loaded into the conversion staging tables.

In this step you should setup a dedicated input location on the object storage and upload the input data files there.

For more details refer to File Locations .

Process Conversion Data Files

After uploading the files to the cloud, you should run a set of batch processes that will process the data files using Oracle SQL Loader and insert the data to the product conversion staging tables.

The staging table load process also produces output files that provide feedback on the load process.

You can download these output files for review.

> **Note**
>
> The process that imports the data in the files to the staging tables, using Oracle SQL Loader, is using the artifacts that were generated in the "Prepare for Conversion" step.

For more details refer to Understanding Load Data Process .

Process Converted Data in Cloud Application

When the data intended for conversion (for a conversion iteration) is loaded into the appropriate conversion staging tables, the OUAF application in the cloud can process that data according to the product's data model, technical and business rules.

Refer to the OUAF product specific documentation for further guidelines and information on the specific processes and guidelines related to data conversion.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This is one step of the overall iterative process but it is also includes many internal steps, for example:
>
> 1. Validation of the data in the conversion staging tables.
> 2. Generation of the system assigned keys for the converted entities.
> 3. Migration of the converted data from the staging tables into the actual system tables.
>
> When the processing of the converted data in this iteration is completed, information from the system is used to identity changes necessary to source data, the extraction process or other activities involved in producing the data to be converted.
>
> The process can then start over by loading a new set of data into data conversion files, load them to the staging tables in the cloud and process them once more to get more feedback.

Approve Converted Data for Production Mode

When the quality and scope of data converted has reached its target, the process can stop. Once the overall data has been approved for production mode the conversion process comes to an end. The process of approving the data includes many activities, for example:

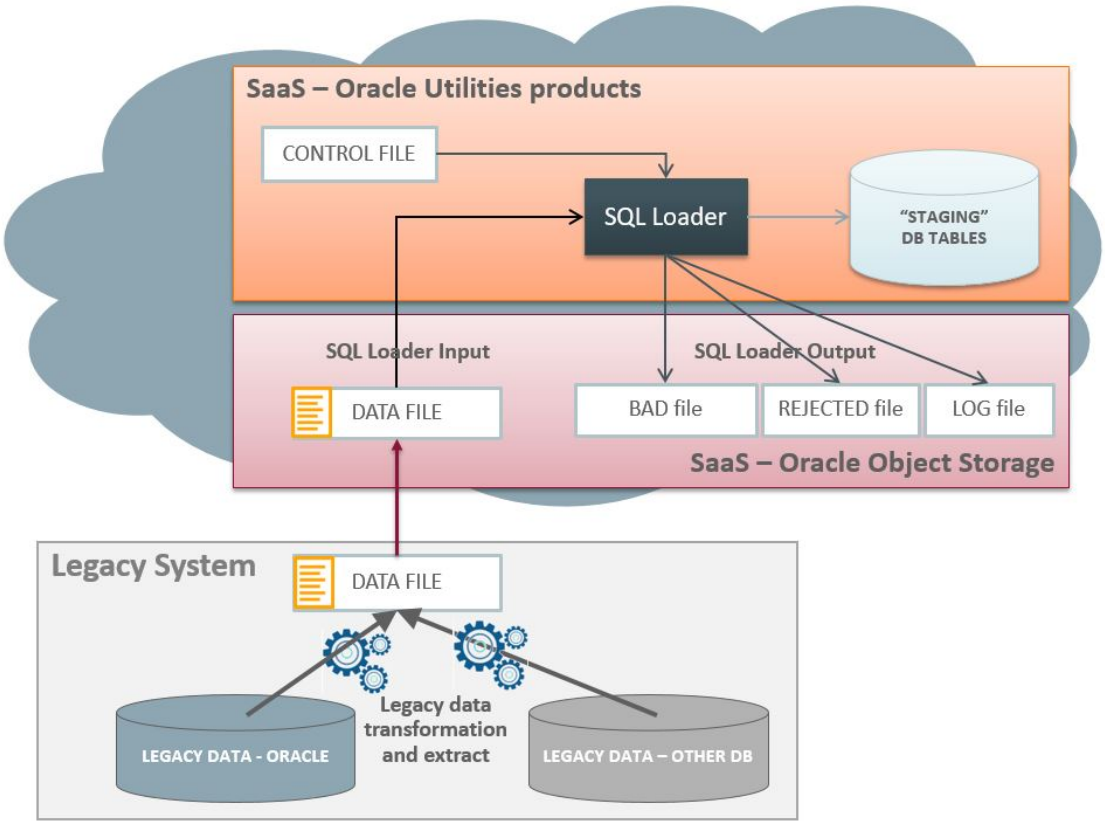
- Running various reports to compare converted data with legacy data to verify the quantity and quality of the data.
- Running reports and tests in the new application to verify the data and the system's behavior using that data.
- Manual review of random and key data in the system.

SQL Loader

Loading the data in the data conversion files into the product conversion stating tables is done using Oracle SQL Loader.

Oracle SQL Loader is an Oracle Database Tool that enables mass upload of data from various sources into database tables in a fast and safe way.

Copyright © 2026, Oracle and/or its affiliates.



SQL Loader processing is determined by Control Files. The control files for the data conversion process are provided for each of the product conversion staging tables as part of the artifact generation which is a part of the preparations activities before starting the conversion process.

The Control Files typically define:

- The location of input files to load.
- The character set of the input data.
- The target table(s) for the load and how to insert the data (e.g. append to or override).
- How to map the data in the input file to tables and fields.
- The location of the output files (e.g. log, bad or rejected records).
- Options for parallel processing and load balancing.

The control files that are generated for the product include all the parameters necessary to load the product conversion staging tables. There are some customization options for these files that are discussed later.

After loading the control files, SQL Loader reads the records in the input data file(s), converts the input into database tables and columns that are inserted into the database.

SQL Loader has 2 modes of operation:

- **Conventional Path Load** â€“ data is inserted to the database by using database SQL INSERT statements.
- **Direct Path Load** â€“ data is inserted into the database using data blocks that are written directly to the database. This method is much faster and therefore is the method selected for the data conversion process.

Copyright © 2026, Oracle and/or its affiliates.

SQL Loader has the ability to deal with multiple data files mapped to multiple tables as well as loading data into CLOB fields.

- Using SQL Loader When Processing Conversion Data Files

Using SQL Loader When Processing Conversion Data Files

Processing of uploaded conversion data files is done via a Batch Process (for each staging table or MO) that does the following:

- Prepares parameters for the SQL Loader process and retrieves Control File for an input table or MO
- Invokes SQL Loader and processes the legacy data extract files that were uploaded to the dedicated File location on cloud
- Archive the input file(s) and upload the SQL Loader output files to the File location so they can be subsequently downloaded by the customer for review.

Conversion Artifacts

The base product includes a set of artifacts that support the conversion process. These artifacts are provided on demand (generated as a result of a user request in the system).

The available artifacts are:

- Input Data Specifications
- SQL Loader Control Files
- Files List
- Conversion Input Data Specifications
- SQL Loader Control Files

Conversion Input Data Specifications

The system will generate a text file with data specification instructions for each of the conversion tables that are supported by the base product.

The specification files are for reference ONLY and are not used by the system at run time. These files describe the data format for each table so that the data specified can be loaded correctly with the Control File that is also generated by the system.

For more details refer to Conversion Artifact Generator .

SQL Loader Control Files

The control files are for internal use and are generated for each of the conversion tables that are supported by the product. These files are used by SQL Loader to interpret and load the data that is provided in the conversion data files.

The control file contains:

- General load parameters and options
- Data field specifications and table-specific load instructions

For more details refer to Conversion Artifact Generator .

Copyright © 2026, Oracle and/or its affiliates.

Configuring Conversion Data Upload

The following sections describe the settings and configurations used by various conversion data upload processes.

- Conversion Master Configuration
- Conversion Task Types
- Conversion Artifact Generator
- Conversion Tasks
- Conversion Batch Controls

Conversion Master Configuration

The **Conversion Data Upload Master Configuration** controls generic aspects of the legacy data load process.

The **Main** section captures the definitions globally applicable for all converted objects (tables and/or maintenance objects):

- **Default Conversion Instructions for Artifact Generator** reference conversion task types that define load parameters for individual tables or maintenance objects.
  The Master Configuration captures conversion task types with default configurations for:
  - Typical Table
  - Typical maintenance Object
  - Key Table.

Special Instructions section captures the definitions that support various idiosyncratic data upload scenarios

- **Override Conversion Eligibility â€" Table** list allows you to specify tables that are not marked for conversion in the metadata, but your implementation wishes to include in the conversion data upload and cleanup scope and vice versa - to exclude tables that are marked for conversion in the metadata from the data upload-related operations

> **Note**
>
> If the table you wish to convert has a related key table, this key table has to be explicitly added to this list.
>
> The sequences and the batch run history and other batch operation-related tables are never eligible for conversion operations.

- **Override Instructions** for **Tables** and **Maintenance Objects** define Conversion Instructions (conversion task types) for objects that require special processing or data formatting. For example, the default conversion instruction defines that CLOB data is supplied as a secondary file. The special instruction for a specific table may define that CLOB data is supplied in the main data file.

Copyright © 2026, Oracle and/or its affiliates.

Conversion Task Types

To configure a new conversion task type, navigate using Admin > Conversion Support > Conversion Task Type.

Conversion task type controls the data load parameters for the target table or maintenance object. Conversion task type also defines the transactional business object for Conversion Task. The algorithms linked to the transactional business object are responsible for the generation of the Conversion Artifacts: control files, input data specifications and file lists.

> **Note**
>
> The product is delivered with several conversion task types whose names start with 'K1'. These configurations should not be changed directly as they reflect the default behavior of the product. The task types get refreshed with base values at each upgrade.. Use them as a templates; if needed, create a copy and modify it according to your requirements.

Conversion Task Types defines:

General Input Data Instructions (all Conversion Task Types Business Objects)

- **Control File Header** - a fragment of the control file that contains general load parameters. This fragment is included in the generated control file. This is a Foreign Key reference to the **Managed Content** entry.
- **Override Control File** - a full text of the control file that contains both load parameters and field definitions. The entire text is copied into generated control file 'as is'. This is a Foreign Key reference to the **Managed Content** entry.
- **Date and DateTime fields' formats**. These attributes are referencing the Extendable Lookup values for *K1-ConvLoadDateFormatLookup* and *K1-ConvLoadDateTimeFmtLookup business objects*
- Strings to use as **data delimiters** and **data enclosing characters**. These attributes are referencing the Extendable Lookup values for *K1-ConvFileDelimiterLookup* and *K1-ConvFileEnclosingCharLookup* business objects
- **CLOB data extract rule** indicates whether the CLOB data is supplied as part of the input data file or as a separate, secondary file
- **CLOB data line separator** defines how the records are separates in the secondary (CLOB) data file. This value should be set according to the operating system where the legacy data extract was created. This attribute is referencing the Extendable Lookup values for *K1-ConvCLOBLineSeparatorLookup* business object.

  For Linux CLOB Line Separator set the value to ***LF***, for Windows, set the value for ***CR+LF***

CLOB Field Instructions for Tables/MO-s that contain multiple CLOB fields (Business Objects *K1-ConvArtMultClobMOTaskType* and *K1-ConvArtMultClobTblTaskType*)

- Target Object - **Table** or **Maintenance Object**
- **Exclude CLOB Fields** list - the list of CLOB fields that are not expected to be provided with the legacy data extract and therefore should be excluded from the generated control file.

The following Conversion Task Types are pre-configured in the product:

Copyright © 2026, Oracle and/or its affiliates.

- **K1-CNV-TABLE** - meant to be used for a single table load. It is referencing a Transactional Conversion Task Business Object whose algorithms generate Table Conversion Artifacts
- **K1-CNV-MO** - meant to be used for a single maintenance object load. It is referencing a Transactional Conversion Task Business Object whose algorithms generate Maintenance Object Conversion Artifacts
- **K1-CNV-KEY-TABLE** - meant to be used for the Key Tables. Key Tables are index-organized tables (IOT) and as such cannot be loaded in multiple concurrent sessions of SQL loader. This Task Type is referencing a Control File Header (Managed Content) where ***Parallel Load*** parameter is set to ***false***

Special configurations for tables that include more than one CLOB field

- Configure a dedicated Conversion Task Type for tables with multiple CLOB fields. Determine and control which CLOB fields to migrate and avoid creating unnecessary empty data files.
- If you wish to migrate all CLOB fields, create a dedicated Conversion Task Type for the Table or Maintenance Object, using business objects *K1-ConvArtMultClobMOTaskType* or *K1-ConvArtMultClobTblTaskType*). Do not populate **Exclude CLOB Fields** list. As a result, the generated Control File and Input Data File Specifications will reference **all CLOB fields**.
- If you wish to migrate only some CLOB fields, create a dedicated Conversion Task Type for the Table or Maintenance Object, using business objects Business Objects *K1-ConvArtMultClobMOTaskType* or *K1-ConvArtMultClobTblTaskType*. Populate the **Exclude CLOB Fields** list. As a result, the generated Control File and Input Data File Specifications will **omit excluded CLOB fields.**
- If no dedicated Conversion Task Type is configured and defined on the Master Configuration as Override Instruction for the Table or MO, the system uses the default Conversion Instruction defined on the Master Configuration. In this case, the generated Control File and Input Data File Specifications will reference a **single CLOB field** among Table's multiple CLOB fields (the one with the lowest table field sequence).

Conversion Artifact Generator

Conversion Artifacts is a set of supporting files used for data upload with SQL Loader. The artifacts are generated based on the table and maintenance object metadata and the definitions from Conversion Master Configuration and Configuration Task Types. Conversion artifacts include SQL Loader control files, input data file specifications and other supplemental files.

- **Input data file specifications** provide detailed field-by-field description of the expected data type and format, field order and other instructions. Your implementation may use these specifications to create the legacy data extract.
- **Control file** defines the rules and options of the data upload
- **File List** enumerates input data files expected to be provided for upload

The generator reads Conversion Master Configuration in order to determine the conversion task type associated with the table or maintenance object.

The generator creates **Conversion Tasks**, one instance per table and/or per maintenance object, creates attachments that store the artifacts and links the attachments to the Conversion Task as Related Objects.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> IMPORTANT: The regeneration is necessary each time the metadata or conversion configurations are amended.

Conversion Tasks query is provided with the base product.

You can choose to generate artifacts for table-level and/or maintenance object-level load, or for the entire system or for specific Table or Maintenance Object:

Generate Artifacts in bulk â€" for all Tables and/or Maintenance Objects

- Submit new batch job for Batch Control K1â€“CNVGA. This process performs a complete refresh of the conversion artifacts. It deletes existing and creates new Conversion Tasks for every Table and/or Maintenance Object that includes tables marked for conversion.

Generate Artifacts for an individual Table or Maintenance Object

- Navigate to **Admin**, then **Conversion**, then **Support**, then **Generate Conversion Artifacts**.
- Select one of the options:
  - **Table search** allows you to select specific target table
  - **Maintenance Object search** allows you to select specific target Maintenance Object.

> **Note**
>
> The process overrides all previously generated artifacts and replaces existing files.

Accessing the Artifacts

Navigate to Conversion Task query and search for the task that is corresponding to a table or a maintenance object. or use other available search criteria. Explore the Attachments collection to view the artifacts.

Conversion Tasks

The following sections provide information about Conversion Tasks and generated conversion artifacts.

- Conversion Task Query
- Conversion Task

Conversion Task Query

Conversion Task query allows to search by variety of criteria:

- Creating User, Conversion Task Type and Status
- Linked Object â€" Table or Maintenance Object
- Exact Conversion Task ID

Copyright © 2026, Oracle and/or its affiliates.

Conversion Task

The purpose of the **Conversion Task** is to maintain the conversion artifacts for Table or Maintenance Object. Please consider the following with respect to conversion tasks.

- The business object lifecycle supports multiple attempts to re-generate the artifacts.
- The re-generation could be triggered either manually or by the Conversion Artifact Generator. Only one set of the artifacts is relevant at any point of time, hence only one Conversion Task at the time can exists for a given Table or Maintenance Object.
- Typically, Conversion Tasks are created by Conversion Artifact Generator. However, it can also be created manually by navigating to **Admin**, then **Conversion**, then **Support**, then **Conversion Task**.
- Use the Conversion Task Query to search for an existing entry. Select the entry from the search results to navigate to the maintenance portal to view and maintain the selected record.
- Conversion Task references a Conversion Task Type and contains standard business object record information, including the creation date/time, status update date/time and creating user.
- The task references the object - **Table or Maintenance Object** - for which the artifacts are generated.
- The collection of task's **Related Objects** store references to the Attachments that store a snapshot of the generated files. A new attachment is created after each generation run and linked to the Conversion Task for audit purposes.

Conversion Batch Controls

The following batch controls are delivered with the base product:

Batch Control	Description
K1-CNVLD	Conversion - Load Data using SQL Loader. This process facilitates the legacy data extract upload using SQL Loader. The input data files are expected to be uploaded to a file storage location. See Input and Output File Location for more information about file storage locations. One run uploads the data into a single target Table or Maintenance Object. The process is multi-threaded. This process is discussed in more details in the Data Load Batch chapter

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Description
K1-CNVDS	Conversion - Disable Conversion This process resets the environment indicators and prevents further conversion activities. Once conversion is disabled, the following actions are blocked: • Switching between Conversion and Production • Table truncation • Disable Indexes and Rebuilding the Statistics • Data insertion into Production • Data upload into staging **Note** The process should be executed only once and only after all of the conversion activities are completed and the application is ready for the real production use.
K1-CNVEN	Conversion - Enable Conversion This process resets the environment indicators and enables it for conversion activities. It should be executed at the start of the Conversion project.
K1-DRPIN	Conversion - Disable Indexes on Production Tables This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the indexes on the batch run history and other batch operation-related tables can not be disabled • If no specific target is populated, the process disables indexes on all converted tables in Production schema
K1-SDSIN	Conversion - Disable Indexes on Staging Tables This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the indexes on the batch run history and other batch operation-related tables can not be disabled • If no specific target is populated, the process disables indexes on all converted tables in Staging schema
K1-DRICMS	Conversion - Disable Customer Modification Indexes This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the process disables indexes on all conversion-eligible tables in Customer Modification schema
K1-RIUSP	Conversion - Rebuild Indexes and Update Statistics in Production This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the process rebuilds indexes and updates statistics on all converted tables in Production schema • Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size. • Specify whether you wish to rebuild indexes, to gather table statistics or both. It is recommended to rebuild the indexes and update statistics right after the data upload.

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Description
K1-RIUSS	Conversion - Rebuild Indexes and Update Statistics in Staging This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the process rebuilds indexes and updates statistics on all converted tables in Staging schema • Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size • Specify whether you wish to rebuild indexes, to gather table statistics or both It is recommended to rebuild the indexes and update statistics right after the data upload.
K1-RUSCM	Conversion - Update Customer Modification Statistics This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the process rebuilds indexes and updates statistics on all conversion-eligible tables in Customer Modification schema • Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size • Specify whether you wish to rebuild indexes, to gather table statistics or both It is recommended to rebuild the indexes and update statistics right after the data upload.
K1-CLNTB	Conversion - Truncate Tables in Production This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the sequences, the batch run history and other batch operation-related tables can not be truncated • If no specific target is populated, the truncates all converted tables in Production schema
K1-SCLTB	Conversion - Truncate Tables in Staging This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the sequences, the batch run history and other batch operation-related tables can not be truncated • If no specific target is populated, the truncates all converted tables in Staging schema
K1-CLNCM	Conversion - Truncate Customer Modification Tables This process can be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the truncates all conversion-eligible tables in Customer Modification schema
K1-CLNRT	Conversion â€” Truncate XML Resolution Tables This process is allowed to run only in the Staging schema and may be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the truncates all XML Resolution tables in Staging schema

Copyright © 2026, Oracle and/or its affiliates.

Batch Control	Description
K1-CLNKY	Conversion - Truncate Key Reference Tables This process is allowed to run only in the Staging schema and may be used in two modes: • Populate Table or Maintenance Object batch parameter to process specific object • If no specific target is populated, the truncates all Key Reference tables in Staging schema

- Configuring Data Load Batch Controls for Conversion Project

Configuring Data Load Batch Controls for Conversion Project

The generic batch process K1-CNVLD performs the load of the input data file(s) into a target table or maintenance object, specified as one of the batch parameters.

Consider creating individual batch controls per object (table or maintenance object).

It allows leverage the parallel processing capacity of the system; the data load processes for all objects can be submitted near-simultaneously, thus minimizing the time needed for the legacy data upload, which is only an initial step in the overall data conversion process.

Use batch control K1-CNVLD for rehearsal load batch run for various tables and/or maintenance objects and determine the optimal load strategy:

- Whether the data should be extracted and loaded as single Tables and/or Maintenance Objects
- Whether and how to split the data files and the number of threads for the batch
- Potential log file size and the feasible logging level

Configure data load Batch Controls for each individual Table or Maintenance Object. Pattern the configuration after K1-CNVLD and set the default parameter values for:

- **Input, Output and Archived File Storage Locations**. The location (compartments and buckets) has to be defined in advance in the cloud object storage; batch parameters are referencing Extendable Lookup ***F1-FileStorage*** value(s) corresponding to these locations. The parameter value should be composed as **file-storage:*//***<Extendable Lookup Value>/<object storage bucket name>, for example: *file-storage://OS-SHARED/CONV-Input*
- Target **Table** or **Maintenance Object** Either one of these two parameters must be populated.
- **File Extension**. When specifying the **file extension**, include all the extensions available. For example. if the file name is ***XXX.csv.gz,*** enter parameter value ***csv.gz***
- **Log Level.** Specify the value *LOG* to force SQL Loader to produce the most detailed data upload process log.
- **Retain Input Option**. This parameter controls whether the original input data files should be left 'as is' at the input data location or purged from the object storage or moved to the archived data location. When the input data is archived the timestamp is appended to the original file name.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note: if the batch process was not able to upload the input data file, this file remains at the input data location.

- **Max Number of Upload Errors**.

  The value of this parameter pertains to the desired errors threshold for the SQL Loader. The load process halts after the specified number of records in error is reached. The SQL Loader log contains a message SKIPPED=NNN where NNN marks the position of the last record in error in the input data file. The process may then be resumed from this position if the batch is submitted with the ***skip*** parameter populated with the value of SKIPPED from the log.

Default **Thread Count** depends on the type of the uploaded table. Regular tables can be uploaded in a single or multiple threads. Key tables must be uploaded in a single thread

Input Data Files

This section provides information on inputting data files for data conversion.

- Overview
- Input Data for a Staging Table
- Input Data for Maintenance Object
- Limitations for Input Data

Overview

This section provides information on the requirements and details for file upload, loading table, maintenance object, CLOB, layout, data formatting, and so on related to inputting data files for data conversion.

- File Creation and Upload
- Loading Table or Maintenance Object
- Multiple Data Files
- Loading CLOB Data
- File Layout and Data Formatting
- Defining Custom Data Delimiters and Enclosing Characters

File Creation and Upload

The extract data may be uploaded in both compressed and decompressed form.

Supported compressed formats include zip and gzip. The files with extension **.gz** and **.zip** will be automatically decompressed by the system.

> **Note**
>
> For the further details on supported compressed formats please refer to the Oracle Utilities Application Framework documentation.

Copyright © 2026, Oracle and/or its affiliates.

File Creation

Use Conversion Task query to retrieve conversion task(s) for table(s) or maintenance object(s) that you are planning to convert. Review input data specifications. Create the data extract text file, using field delimiters, enclosing characters and data formatting according to the specifications.

If the files were created under Windows operating system, it is recommended to remove the carriage return character prior to compression (not applicable for secondary files with CLOB data).

File Compression with gzip

Use gzip utility to compress the file.

- For Windows operating system, download and install gzip utility for Windows. Run the following:

```
# Navigate to the gzip folder
cd C:\gzip-1.3.12-1-bin\bin
# Run gzip.
# New file will be created in the same directory as the original the input file
gzip -k <full path to the input data file>
# Sample command for the file TABLE_NAME.txt located under
# C/myDataDirectory/:
gzip â€“k C/myDataDirectory/TABLE_NAME.txt
```

- For Unix/Linux operating systems, run the following command:

```
gzip < <inputFile name> > <outputFile name with .gz suffix>
# Sample command for the file TABLE_NAME.txt
gzip < TABLE_NAME.txt > TABLE_NAME.txt.gz
```

Loading Table or Maintenance Object

The conversion data upload supports two types of the extract:

- Data extract that contains the data for a single target staging table.

  In this case, the upload process inserts records into a single staging table. The input data file consists of similar records, each represents a row in the target table.

- Data extract that contains the data for the group of target staging tables that represent a Maintenance Object.

  In this case, the upload process populates all the tables that belong to a maintenance object. The input data file contains multiple types of records that are inserted into various tables.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the data is uploaded for a Maintenance Object, each record has to be prefixed with the table name that is used as record type indicator. This requirement is also mentioned in the input file specification

- The application stores the large volumes of data in partitioned tables. If the Maintenance Object contains partitioned tables, the data should be uploaded in a certain order:
  - During the data upload, the primary table in the Maintenance Object should be loaded first, then the indexes should be rebuilt, followed by the data upload into the child tables

Multiple Data Files

The large volume of the converted data may require the extract to be split into multiple parts.

In this case the multi-part extract for a target staging Table or Maintenance Object is loaded with a multi-threaded batch run.

Sample scenario:

10M of the Person records are being converted. The extract file is split into 10 files, PERSON1, PERSON2 PERSON3 etc., each file contains 1M records. All 10 files are uploaded to the input file location.

Batch process 'Load Person data' is submitted with 5 threads; each thread picks up 2 files and uploads the data into staging tables.

Loading CLOB Data

The CLOB data can be loaded in two ways:

- Extracted as part of the main data file. This method is suitable for the relatively small CLOB data size. Refer to Limitations for more details.
- Extracted into a separate file. This method is suitable for larger CLOB data sizes and also for large volume tables or maintenance objects. In this scenario, the secondary file has to be uploaded at the same time as the primary data file.

> **Note**
>
> When loading CLOB data as a secondary file, a separate file is created for each CLOB field in the. It means that if the table 'XXX' contains CLOB fields 'clob1Field1' and 'clobField2', the system would expect to process files XXX, XXX_clobField1_CLOB and XXX_clobField2_CLOB. See also Notes on Secondary Data Files

- Notes on Secondary Data Files (CLOB Data)

Notes on Secondary Data Files (CLOB Data)

Secondary data file is loaded simultaneously with the primary data file. The CLOB data extract should contain exactly the same number of records as a primary file.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When splitting the extract into multiple data files, make sure that the numbers of extracted records in the primary and secondary files are matching.
>
> It is applicable for both table-level and maintenance-object level load.

Sample Scenario:

Table CI_PER includes CLOB column C1_PER_DATA_AREA. The CLOB field is used to store customer's resume.

The legacy table contain 10K records and the CLOB data is available for some of the records

The primary data file, CI_PREM.csv contains 10K lines, one line per record

The secondary data file, CI_PREM_CLOB, contains 10K records too. The records with the empty CLOB are represented by CLOB delimiter.

Primary File – Regular Data Fields	Secondary File - CLOB
12345678,John,Doe,.....	John's resume <endclob>
45165609,Smith,Mary,.....	Mary Smith resume<endclob>
33336733,Brand,Brent,.....	<endclob>
06737482,Sunshine,Julie,.....	Julie Sunshine resume<endclob>

Splitting single table data extract with CLOB

The following example illustrates how to split the single table large volume extract correctly if the CLOB data is supplied in the secondary file:

Assuming the table has 10K records, split into two equal parts. The following files should be created:

- Primary File 1 with 5K records
- Secondary File 1 with 5K CLOB entries
- Primary File 2 with 5K records
- Secondary File 2 with 5K CLOB entries

Splitting maintenance object data extract with CLOB

The following example illustrates how to split the maintenance object large volume extract correctly if the CLOB data is supplied in the secondary file:

Assuming the maintenance object has two tables, one of the tables includes CLOB column and the primary table has 10K records. The data supposed to be split into two equal parts. The following files should be created:

- Primary File 1 with 5K records for the primary table and corresponding records for the child table
- Secondary File 1 with CLOB entries corresponding to the table in the Primary File 1
- Primary File 2 with 5K records for the primary table and corresponding records for the child table

Copyright © 2026, Oracle and/or its affiliates.

- Secondary File 2 with CLOB entries corresponding to the table in the Primary File 2

> **Note**
>
> For Maintenance Object-level load consider supplying CLOB data in the primary file, to avoid complicated logic of data splitting.
>
> Refer to CLOB Data - XML for limitations concerning XML CLOB data

File Layout and Data Formatting

The input data specifications file for each Table or Maintenance Object that are subject to conversion are stored as an attachment linked to the corresponding conversion tasks. The Conversion Artifact Generator creates conversion task for each table and/or maintenance object.

Specifications are generated based on the metadata and the conversion configurations defined on Conversion Task Type and Conversion Master Configuration. Configurations control delimiters and enclosing characters and define CLOB data processing rules.

The data extract file is expected to contain one record per each target table row. The data field order in the record has to be exactly the same as it is listed in the specification file.

The specifications contain:

- Data delimiters and expected secondary data file names
- Field-by-field data formatting rules for

  data types and size, Date, DateTime and Time formatting
- Special notes on CLOB fields

The fields in the record are expected to be separated by a delimiter character (or string).

For the not-nullable columns with no data, a single space enclosed in the enclosing character (or string) has to be supplied, for example " "

The CLOB data in the main record has to be enclosed in the CLOB Delimiter string, for example *<clobend>some clob data<clobend>*

The CLOB records in the secondary data file has to be separated by CLOB Delimiter string followed by the new line sign, defined as the CLOB Line Separator. IMPORTANT! the last entry in the CLOB data file has to end with the new line, too

- If the data extract is created on Linux, use Line Feed to separate CLOB records and make sure that there is no space after CLOB Delimiter string
- If the data extract is created on Windows, use Line Feed followed by Carriage Return.

Enclosing character (or string) can also be used to encapsulate data that includes a field delimiter, for example *"Hello, World"*

Defining Custom Data Delimiters and Enclosing Characters

The default configurations are delivered with the base product.

Copyright © 2026, Oracle and/or its affiliates.

Your entire legacy data or a specific table, or the data extract mechanism may require use of a different data delimiter and/or enclosing characters. To enable alternative delimiters, perform the following steps:

- Define new data delimiter string, enclosing string, CLOB delimiter string and/or CLOB Line Separator using the corresponding Extendable Lookups
- Configure Conversion Task Type using newly created delimiters
- Amend Conversion Master Configuration, setting default (for all tables or maintenance objects) or override (for specific table or maintenance Object) instructions.
- Regenerate Conversion Artifacts

Now you can create data extract according to the newly generated specifications.

Input Data for a Staging Table

The extract for a single table contains delimited data records. Depending on the conversion configuration, the CLOB data should be supplied either inline or in a separate, secondary file.

Input Data File Names

Input data file names has to follow the rules described in this chapter. The names are case-sensitive.

File Type	File Name
Single data file per table, no CLOB	File name equal target table name, such as CI_PREM
Single data file per table, **single** CLOB column in the table, CLOB data in secondary file	Primary file name equal target table name, such as CI_PREM Secondary file name is composed as target table name concatenated with '_CLOB', such as CI_PREM_CLOB
Single data file per table, **multiple** CLOB columns in the table, CLOB data in secondary file	Primary file name equal target table name, such as CI_PREM **For each CLOB column** being uploaded, the secondary file name is composed as target table name concatenated with CLOB column name concatenated with '_CLOB', such as CI_PREM_COL1_CLOB, CI_PREM_COL2_CLOB
Multiple data files per table, no CLOB	File name is composed as target table name concatenated with file number, such as CI_PREM1, CI_PREM2
Multiple data files per table, CLOB data in secondary file	Primary file name is composed as target table name concatenated with numeric file number, such as CI_PREM1, CI_PREM2, and so on. Secondary file name is composed as target table name concatenated with '_CLOB' and file number, such as CI_PREM_CLOB1, CI_PREM_CLOB2, and so on.

Copyright © 2026, Oracle and/or its affiliates.

File Type	File Name
Multiple data files per table, **multiple** CLOB columns in the table, CLOB data in secondary file	Primary file name is composed as target table name concatenated with numeric file number, such as CI_PREM1, CI_PREM2, and so on. **For each CLOB column** being uploaded, the secondary file name is composed as target table name concatenated with CLOB column name concatenated with '_CLOB' and file number, such as CI_PREM_COL1_CLOB1, CI_PREM_COL1_CLOB2, CI_PREM_COL2_CLOB1, CI_PREM_COL2_CLOB2, and so on.

Input Data for Maintenance Object

The extract for a maintenance object contains delimited data records for all tables included in the maintenance object. Depending on the conversion configuration, the CLOB data should be supplied either inline or in a separate, secondary files, one file per table containing CLOB. Refer to Notes on Secondary Data Files (CLOB Data) for limitations concerning XML CLOB data and more details.

Input Data File Names

Input data file names has to follow the rules described in this chapter. The names are case-sensitive.

File names for Maintenance Object contain Maintenance Object Code. If the code includes spaces or special characters, it should be replaced with '_', for example:

For actual MO Code: SP/MTR HIST, in the file name, use SP_MTR_HST

File Type	File Name
Single data file per Maintenance Object, no CLOB	File name equal target maintenance object name, such as PREMISE
Single data file per Maintenance Object, **single** CLOB column in the table, CLOB data in secondary file	Primary file name equal maintenance object name, such as PREMISE Secondary files name composed as maintenance object name concatenated with table name concatenated with '_CLOB'. For example, if PREMISE's CI_PREM table contains CLOB, the secondary file name is: PREMISE_CI_PREM_CLOB
Single data file per Maintenance Object, **multiple** CLOB columns in the table, CLOB data in secondary file	Primary file name equal maintenance object name, such as PREMISE **For each CLOB column** being uploaded, the secondary files name composed as maintenance object name concatenated with table name concatenated with CLOB column name concatenated with '_CLOB'. For example, if PREMISE's CI_PREM table contains two CLOB columns COL1 and COL2 , the secondary file names are: PREMISE_CI_PREM_COL1_CLOB, PREMISE_CI_PREM_COL2_CLOB

Copyright © 2026, Oracle and/or its affiliates.

File Type	File Name
Multiple data file per Maintenance Object, no CLOB	File name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc
Multiple data file per Maintenance Object, single CLOB column in the table, CLOB data in secondary file	Primary file name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc Secondary files name composed as maintenance object name concatenated with table name concatenated with '_CLOB' concatenated with file number. For example, if PREMISE's CI_PREM table contains CLOB, the secondary file names are: PREMISE_CI_PREM_CLOB1, PREMISE_CI_PREM_CLOB2, etc
Multiple data file per Maintenance Object, **multiple** CLOB columns in the table, CLOB data in secondary file	Primary file name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc **For each CLOB column** being uploaded, the secondary files name composed as maintenance object name concatenated with table name concatenated with column name concatenated with '_CLOB' concatenated with file number. For example, if PREMISE's CI_PREM table contains two CLOB columns COL1 and COL2 , the secondary file names are: PREMISE_CI_PREM_COL1_CLOB1, PREMISE_CI_PREM_COL1_CLOB2, PREMISE_CI_PREM_COL2_CLOB1, PREMISE_CI_PREM_COL2_CLOB2 etc

Limitations for Input Data

Key Tables

Due to technical limitations, Key tables cannot be loaded in multi-threaded mode. If you decide to create input data files using the Maintenance Object method, consider the following:

- Create data extract(s) for a Maintenance Object data, but exclude the Key table(s) and run data upload in multi-threaded batches
- Create separate extract for the Key table(s) only and upload them in single threaded batches.

Duplicate Records

Duplicate records are not rejected during the upload. This may result in the index corruption and other issues in the target table(a).

The default SQL Loader data upload method configured in the system is Direct Path. Among the advantages of a Direct Path method are higher performance and more efficient resources utilization. Constraints are disabled during Direct Path load and as a result the duplicate PK records are not rejected.

There are two approaches to address the issue

- Clean up the duplicate records from the legacy data extract BEFORE the upload

Copyright © 2026, Oracle and/or its affiliates.

- Customise the default configuration and switch to the Conventional Path load method.
  - **Create custom Control File Header**: Duplicate Managed Content K1-SQLLoaderControlFileHeader .and modify to use Conventional Path Load (Consult SQL Loader documentation for more detailed information).
  - **Modify Conversion Instructions** (Conversion Task Type): specify SQL Loader Control File Header created above
  - **Regenerate Conversion Artifacts**

CLOB Data Size

The maximum size of the CLOB data included in the main data file is 10K. If converted CLOB data size is expected to be larger than 10K, create secondary data files.

CLOB Data - XML

The XML data formatted using new line characters cannot be uploaded as part of the primary data file. If your CLOB data includes formatted XML, you should create secondary data file(s) for CLOB columns.

Conversion Orchestration

This section provides information on orchestrating a data conversion.

- Input and Output File Location
- Understanding Load Data Process
- Understanding the Conversion Orchestration Process

Input and Output File Location

The following locations has to be defined in order to support legacy data upload:

- File location for input data files upload
- File location for SQL Loader output files and also for archived input data files

See Object Storage chapter for more details on how to define the file location.

Understanding Load Data Process

The setup of the batch controls that is necessary to support data conversion and legacy data extract upload is described in Data Load Batch Controls .

The following sections explain the process of legacy data extract upload with SQL loader.

- Load Data Batch Run Parameters
- Multi-Threading
- Load Outcome â€“ Log, Bad and Rejected Files
- Interrupted Load
- Tables Cleanup and Index Maintenance
- Load Performance Tuning

Copyright © 2026, Oracle and/or its affiliates.

Load Data Batch Run Parameters

The data upload batch is processing single target object â€“ Table or Maintenance Object. The data extract may be supplied in one or multiple files, created and named according to the Input File instructions for a Staging Table and Maintenance Object .

The process is multi-threaded and optimized for the parallel load.

Using the batch parameters you can control:

- Level of logging
- Data extract retention
- Resume the interrupted load
- Maximum number of load errors

Multi-Threading

The process is threading by dividing the multiple data files between the threads. Each thread in turn, is processing its files one by one.

The exception is Key Tables. They are index-organized tables and have to be loaded in the single thread, due to the SQL Loader restrictions.

Load Outcome â€“ Log, Bad and Rejected Files

SQL Loader output files are available for download after each run . Refer to Input and Output File Location for details. To ensure file name uniqueness, the output file names are concatenated with the current timestamp.

- **Log file** contains the information about the run, including number of record processed, errors encountered and other useful information. The level of logging is controlled by the batch parameter ***log*** .
- **Bad** and **Rejected** files contain the records that were rejected by either the SQL Loader or the database. If the errors are not extensive and could be fixed manually, these files can be edited, renamed and used as an input again.

Interrupted Load

The data load process can be re-started. Once SQL Loader reached the maximum number of errors allowed for the run, it records the position of the last record in error in the log. The restarted load begins pulling the data from the file after skipping certain number of records.

In order to continue processing interrupted load, perform the following steps:

- Retrieve the log file. Find statement '*Specify SKIP=* ***n*** *when continuing the load .'* in the log file.
- Examine the log file and determine whether the problems with the data are easily amendable. Retrieve .bad and .rejected files from the same location where SQL Loader log files are found, review the problem and decide whether the entire extract has to be corrected or the upload should continue forward.
- Re-upload the input data file to the server
- Submit the batch once again, this time specifying the ***n*** as a ***skip*** batch parameter

Copyright © 2026, Oracle and/or its affiliates.

Simillar approach can be utilised to recover from the partial data upload. Note that in order to provide an additional safeguard for your effort and allow recovery it is recommended to create the data extract with records ordered by the primary key. This technique will allow you to determine the portion of the remainder of the un-uploaded data and make decision on the subsequent actions. If you are uploading large volumes of data and the interruption happened toward the end of the upload process it could be beneficial to complete the upload rather than to cleanup and start over.

Consider the following scenarios:

- Data Upload batch process ended abnormally due to an unrecoverable event such as software malfunction. In this case neither SQL loader nor the batch process are able to end gracefully and report the number of records processed.
- Data Upload batch process encountered a legitimate error and the batch thread and the entire batch run is now in error status.
- The SQL Loader process ended with fatal error that prevented it to log the "skipped" option properly and it is not found anywherein the SQL Loader log

If you decied to proceed with recovery, perform the following steps:

- Query the table and determine the primary key of the last record that was sucessfully uploaded - this would be possible if the data in the input file was ordered
- Then determine the position of this record in the data file
- Finaly, submit the batch once again and use the ***skip*** batch parameter to resume the load from this position

Tables Cleanup and Index Maintenance

The SQL Loader runs in parallel mode in order to maximize performance. Therefore the target tables are not emptied prior to the data insertion and must be done explicitly, by running corresponding batch jobs.

Indexes are disabled during the data upload and then need to be re-built afterwards. The product provides batch processes that support index disabling and re-building indexes and gathering table statistics.

Most of the larger volume tables in the application are partitioned. The partitioned indexes cannot be disabled implicitly by the data upload process and should be disabled implicitly before running the upload.

If your are migrating an existing on-premise Oracle Enterprise Utilities customer to the cloud, the migration requirements may include the data in the custom tables. On cloud, these tables reside in the Customer Modification schema. The product provides the dedicated batch controls to operate on the Customer Modification schema.

- Use batch control ***K1-SCLTB*** to truncate tables in **Staging** area and batch control ***K1-CLNTB*** to truncate tables in **Production**. Use batch control ***K1-CLNCM*** to truncate tables in Customer Modification schema.
- Use batch control ***K1-SDSIN*** to disable indexes in **Staging** area and batch control ***K1-DRPIN*** to disable indexes in **Production**. Use batch control ***K1-DRICM*** to disable indexes on tables in the Customer Modification schema.
- Use batch control ***K1-RIUSS*** to rebuild indexes and/or update table statistics in Staging and ***K1-RIUSP*** to perform the same in **Production**. Use batch control ***K1-RUSCM*** rebuild indexes and/or update statistics on tables Customer Modification schema.

Copyright © 2026, Oracle and/or its affiliates.

- Use batch control ***K1-CLNKY*** to truncate **Key Reference** tables in **Staging** area
- Use batch control ***K1-CLNRT*** to truncate **XML Resolution** tables in Staging Area
- Use batch process ***K1-SDSTG*** to disable Triggers in **Staging** and batch control ***K1-PDSTG*** to disable triggers in **Production**
- Use batch process ***K1-SENTG*** to enable Triggers in **Staging** and batch control ***K1-PENTG*** to enable triggers in **Production**

Supported scenarios include:

- **Truncate Specific Table or Maintenance Object**: In order to truncate the specific table, submit the corresponding batch job, specifying the `table` parameter. In order to truncate the data stored in a specific Maintenance Object, submit the corresponding batch job, specifying the `maintenanceObject` parameter. The process will also disable indexes in the truncated tables.

  > **Note**
  >
  > If the truncated table's metadata defines a Key table, the Key table gets truncated too.

- **Truncate all Converted Tables**: In order to truncate all converted tables run the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters. The process will disable indexes and truncate all tables marked for conversion in the metadata.
- **Truncate Key Reference Tables or XML Resolution Tables for Specific Table or Maintenance Object**: In order to truncate Key Reference table linked to a staging table, submit the corresponding batch job, specifying the `table` parameter. In order to truncate Key Reference tables for a specific Maintenance Object, submit the corresponding batch job, specifying the `maintenanceObject` parameter.
- **Truncate all Key Reference Tables or XML Resolution Tables**: Submit the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters to truncate all Key Reference tables.
- **Disable Indexes for Specific Table or Maintenance Object**: In order to disable indexes for the specific table, submit the corresponding batch job, specifying the `table` parameter. In order to disable indexes for the specific Maintenance Object's tables, submit the corresponding batch job, specifying the `maintenanceObject` parameter.
- **Disable Indexes for all converted Tables**: In order to disable indexes on all converted tables run the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters. The process will disable indexes and truncate all tables marked for conversion in the metadata.

> **Note**
>
> You should not use conversion batch processes to truncate and/or alter indexes on the batch and batch job stream-related tables. Attempt to cleanup the batch job history or reset sequences may render the system un-operational.

Copyright © 2026, Oracle and/or its affiliates.

Load Performance Tuning

The data specifics â€“ volumes, structure and other aspects â€“ may justify customizing the default configurations for all or some tables.

The SQL Loader is processing data according to the Control File. The fragment that contains general load parameters is stored as Managed Content that is referenced on the Conversion Task Type.

You can create new Managed Content, configure the Conversion Task Type and specify the new settings on Conversion Master Configuration, for all or specific Tables or Maintenance Objects.

Understanding the Conversion Orchestration Process

The following sections provide information on general aspects of conversion orchestration.

Enabling Conversion

Conversion activities are possible as long as conversion is enabled in the environment. The authorized Conversion Administrator may run the corresponding batch job to enable conversion.

Switching between Staging and Production Schemas

In the environment enabled for conversion, the online application is running against the database schema that contains synonyms pointing to either Staging or Production tables.

The switching is possible only while Conversion is Enabled

Navigate to **Admin > Conversion Support > Switch Schema** to switch the synonyms, .

The current schema information is displayed on the sidebar zone when Conversion is enabled in the environment.

> **Note**
>
> The operation fails if there are batch processes currently in progress. Should you receive the error message while attempting to switch synonyms, please review the running batches and wait until its all completed before repeating the attempt.

Preparing for Conversion

The preparation steps include evaluating initial conversion configurations provided by the product, customizing the configurations if needed, and finally, generating conversion artifacts.

Most of the preparation should be done when the environment is running against the Production schema, except for job streams for conversion processes that are scheduled and, published when the schema is switched to Staging.

Conversion Development

The creation of the legacy data extract and rehearsal of the data load and subsequent conversion data processing is done while the environment is running against the Staging schema.

Copyright © 2026, Oracle and/or its affiliates.

Consider creating and scheduling the job stream for the sequence of the batch processes for both data load and following validation, key generation and insertion.

Uploading Data

SQL Loader implicitly disables the indexes in the non-partitioned tables before the upload. In the partitioned tables the indexes needs to be explicitly disabled by the batch process prior to the upload. Once the data upload is completed the indexes have to be rebuilt.

Upload has to be done in a certain order if loading single tables that belong to a Maintenance Object whose primary table is partitioned. The child tables within such Maintenance Objects contain Foreign Key constraints to the primary table's Primary Key. The primary table has to be uploaded first. The indexes should be rebuilt right after upload. The subsequent tables should be uploaded according to the Maintenance Object table's hierarchy (parent first, child next)

You may combine the table truncation, data upload and the index maintenance and statistic updates into a batch stream and create a dedicated batch job stream for each converted Table or Maintenance Object.

Conversion Run

For the mock-up, or dress rehearsal or the actual go-live conversion run you may choose to amend data load batch controls for tables/maintenance objects to improve performance:

- Reduce the logging level - set ***log*** batch parameter to NOLOG
- Do not retain input data - set ***retaininput*** batch parameter to PURGE

Disabling Conversion

After conversion is completed, the various activities such as table truncation and index maintenance should not be allowed.

Upon conversion project completion, switch the schema to Production and request that the authorized Conversion Administrator run the corresponding batch job to disable Conversion in the environment.

Enabling Incremental Conversion

Legacy data conversion may be done in stages. After the first chunk is converted and the system begins running in production, you may need to convert the next portion(s) of the data. This orchestration macro-scenario is called *incremental conversion*; it imposes certain limits and restrictions on the conversion activities and it has to be approved by the management.

If the incremental conversion has been approved, request that the authorized Conversion Administrator run the corresponding batch job to enable the incremental conversion in the environment.

Security Setup for Conversion

The services associated with data conversion are combined into the following user groups:

- **Conversion Administrators** includes the services linked to the most sensitive database operations: enable/disable conversion and switching database synonyms between production and staging schema
- **Conversion Development** includes the services associated with conversion-related configurations

Copyright © 2026, Oracle and/or its affiliates.

- **Conversion Operations** provides access to services associated with data load batch processes. This group could be extended to include services associated with specific table(s) or maintenance object data loads

These user groups are included in the base product. Your implementation may chose to use it as is or create a different setup.

Conversion Reconciliation Reports

Your cloud service subscription includes online database querying capabilities via Analytics Publisher and Database Actions. These can be used to create and run data reports against both production and staging schema tables. Contact your Security Administrator for the access information.

Batch Scheduler for Cloud Implementations

This chapter provides information and requirements on batch scheduler in support of Oracle Utilities Cloud implementations.

- SaaS Batch Scheduler Overview
- Scheduler Batch Stream Definitions
- Batch Scheduler Batch Control
- Scheduler Batch Stream Operations

SaaS Batch Scheduler Overview

SaaS Batch Scheduler is a feature in Oracle Utilities Cloud Service Foundation that provides a SaaS batch scheduler engine and user interfaces. It facilitates customers and implementers to define, manage and schedule stream of batch jobs to run at periodic intervals using a user friendly interface. The SaaS batch scheduler engine is responsible for scheduling and processing the stream of batch jobs. It has a processing loop that runs every X seconds(configured by default at 10 seconds) to perform these operations. In addition, users can monitor the progress of the streams, each of the steps and also perform required exception handling. The SaaS batch scheduler engine processes the stream of batch jobs per scheduled time, within that minute.

SaaS Batch Scheduler feature is supported by the following system entities:

- Scheduler Batch Stream Definition â€“ a Business Object based on the Batch Job Stream Definition maintenance object
- Batch Scheduler Batch Control â€“ a Business Object based on the Scheduler Program maintenance object

Scheduler Batch Stream Definitions

This section provides information on Scheduler Batch Stream Definitions.

- Defining a Scheduler Batch Stream Definition
- Activating a Scheduler Batch Stream Definition

Copyright © 2026, Oracle and/or its affiliates.

- Updating an Active Scheduler Batch Stream Definition
- Inactivating a Scheduler Batch Stream Definition
- Canceling a Scheduler Batch Stream Definition
- Deleting a Scheduler Batch Stream Definition
- Migrating Scheduler Batch Stream Definitions
- Tips For Setting Up Scheduler Batch Stream Definitions
- System Override Date Support

Defining a Scheduler Batch Stream Definition

In order to define a Scheduler Batch Stream Definition, from the **Admin** menu, select **Batch Operations**, select **Scheduler Batch Stream Definition**, and select **Add**.

A Scheduler Batch Stream Definition includes specifying the batch jobs and the order in which they should be run as "Stream Steps", the user to use to run these batch jobs and a "Schedule" to state when and how often the batch job stream should run. Batch jobs can be setup to run in sequence or in parallel fashion or a combination of both.

Business Date: A Scheduler Batch Stream Definition can be optionally configured to run on a particular business date. When a business date is specified, each of the batch jobs in the stream run with the business date set. The business date can be either a specific business date or a variable business date. For example, business date set to run a day before current date. This ensures every time the stream runs the batch jobs in the stream run with current date -1.

Run On Activate: A Scheduler Batch Stream Definition can also be optionally configured to run as soon as it's activated.

Start Schedule Date/Time: It can optionally be configured to specify schedule to be effective in the future by setting a date/time to start schedule. If the start schedule date/time is not provided then the schedule on the Scheduler Batch Stream is effective as of current date.

The batch job to run for each stream step can be either specified directly or via a "Batch Scheduler Batch Control". A Batch Scheduler Batch Control enables setting custom batch parameters on a Batch Control. When a batch job is submitted for a stream step that has Batch Scheduler Batch Control specified on it, the custom batch parameters are used on the submitted batch job. Refer to Configuring a New Batch Scheduler Batch Control for more details.

A Stream step includes specifying the following details

- Step Name: the name of the step
- Batch Control: the batch control to run for that step.
- Thread Count: the number of threads to use to run this step's batch control
- Batch Scheduler Batch Control: the batch scheduler batch control to use for that step. Either a batch control or batch scheduler batch control can be specified but not both
- Step Conditions: specifies when the current step should run. It includes the following details
  - Step Name: the step that should have run prior to current step
  - Step Status: the status the prior step should be in when its finished

Copyright © 2026, Oracle and/or its affiliates.

- Step Condition: indicates how (And/Or) prior step should combine with next subsequent step conditions. For example, if step C should execute after step B and A have ended, then step C's has two step conditions 1)A Ended And 2)B Ended.

> **Note**
>
> There can be stream configurations where a step is specified and no steps run after the specified step runs. This is not a common scenario and a warning is logged on the Log tab to adjust stream configuration, if needed.

A Scheduler Batch Stream Definition's schedule can be specified using various frequencies.

- Daily, By Minutes: run every "x" minutes every day except on specified excluded days
- Daily, By Time: run at certain times every day except on specified excluded days
- Monthly, By Month Day Number: run on a specific day of the month, at specified time
- Monthly, By Week Days: run on days of a specific week in a month, at specified time
- Weekly: run every "x" weeks on specific days, at specified time
- Yearly, By Month Day Number: run on a day of the month, at given time in specified months
- Yearly, By Week Days: run on days of a specific week, at given time in specified months
- Custom: run on a specific date at specified time

A Scheduler Batch Stream Definition can be setup to have level of service algorithms. Following three level of service algorithm types are supported.

- Stream Run has failed.
- Scheduler Batch Stream not started X minutes.
- Scheduler Batch Stream Job Ran Too long (Relative Run).

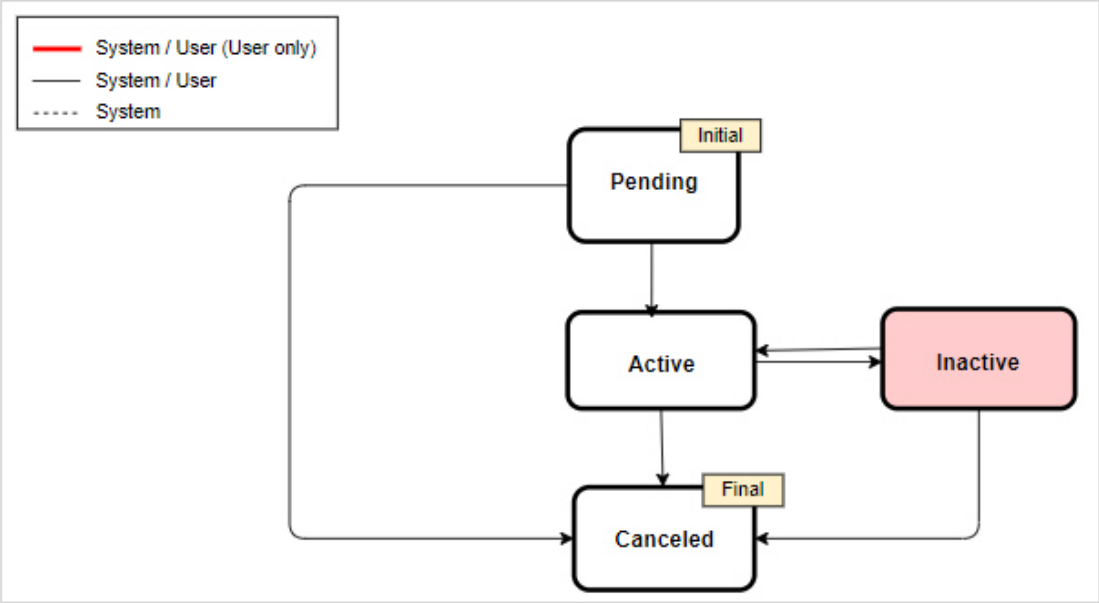
A Scheduler Batch Stream Definition can also be created without defining a schedule. When activated, it remains activated without a schedule and can be used for any manual runs.

There are custom ways in which a scheduler batch stream definition can be setup. This involves creating a Batch Scheduler Batch Control. Refer to Tips For Setting Up Batch Job Stream Definitions and Batch Scheduler Batch Control for more information.

Activating a Scheduler Batch Stream Definition

Once the Scheduler Batch Stream Definition is ready, it can be activated in the SaaS batch Scheduler to run at periodic intervals by transitioning the Scheduler Batch Stream Definition to Active state. An activated Scheduler Batch Stream Definition is called a Scheduler Batch Stream. A Scheduler Batch Stream runs based on the schedule defined on it.

Copyright © 2026, Oracle and/or its affiliates.



By default, every batch job in a Scheduler Batch Stream Definition is internally setup to run in such a way that if that batch job fails the Scheduler Batch Stream run stops execution. User then has an option to fix the issue and re-start the stream.

Updating an Active Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be changed while in Active state. While active, any changes made to the Scheduler Batch Stream Definition are reflected on the next Scheduler Batch Stream run.

Changes to a Scheduler Batch Stream Definition can be made by editing it or using the edit options in the Scheduler Batch Stream Definition steps tab.

Inactivating a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be stopped from running further by inactivating it. Once inactivated, the SaaS batch scheduler stops scheduling and processing the scheduler batch stream definition. This is useful in situations where the stream needs to stop running temporarily for a brief period of time.

A Scheduler Batch Stream Definition can be re-activated back for scheduling and running in the future.

Canceling a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be canceled so that it no longer runs. If a scheduler batch stream run is ongoing when the scheduler batch stream definition is canceled, a warning is used. Canceling a scheduler batch stream definition has no impact on that currently ongoing run. User has the option to continue with the cancelation of the scheduler batch stream definition and let the scheduler batch stream run or navigate to the scheduler batch streams operations portal to cancel the currently ongoing run and then cancel the scheduler batch stream definition. Canceled scheduler batch stream definitions are allowed to be deleted.

Copyright © 2026, Oracle and/or its affiliates.

Deleting a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be deleted in Pending, Inactive and Canceled states. In case of deleting an Inactive or a Canceled scheduler batch stream definition, respective scheduler batch stream runs are also deleted.

Migrating Scheduler Batch Stream Definitions

Scheduler Batch Stream Definitions can be migrated to other environments using the Content Migration Assistant (CMA) process or the Process Automation Tool.

Scheduler Batch Stream Definitions and Batch Scheduler Batch Controls are added in the target environment in the same status as the source environment. That is active scheduler batch stream definitions are added as active.

A Scheduler Batch Stream Definition should be marked as Inactive in source environment before migrating to target environment if it should not run as soon as it's migrated.

> **Note**
>
> If a Scheduler Batch Stream is already scheduled in target environment and has next scheduled run date/time populated and the migration from source environment results in the stream having a different schedule, the new schedule will not be picked up by SaaS batch scheduler until the next run happens on the previously scheduled time. In order for the new schedule to take effect immediately, the scheduling details of the stream definition can be edited which resets the next scheduled run date/time so SaaS batch scheduler can re-determine next scheduled run date/time per new schedule.

Tips For Setting Up Scheduler Batch Stream Definitions

Configuring scheduler batch stream definitions to avoid open-ended stream issue

The scheduler batch stream definitions should always be configured such that the stream "ends" only once all the steps in the stream are completed either by configuring it on the "end of stream" step condition or by defining the inter-dependencies between the steps. This ensures the scheduler batch stream run execution does not stop in the middle of the stream.

Transferring batch job stream definitions to SaaS batch scheduler

Customers that are upgrading from prior cloud releases using Oracle Scheduler are recommended to plan transfer and migration of existing batch job streams definitions to SaaS Batch Scheduler. The recommended steps are:

- (Optional Step) Migrate existing batch job stream definitions from production environment to any pre-prod environment.
- In any pre-prod environment, use the transfer option to move the existing batch job stream definition to scheduler batch stream definition (new object for SaaS Batch Scheduler).
- Perform thorough testing of newly created scheduler batch stream definitions.
- Before migrating to the target environments, the batch job stream definitions in the target environment should be canceled(if needed) and deleted.

Copyright © 2026, Oracle and/or its affiliates.

- Migrate the newly created scheduler batch stream definitions by using Content Migration Assistant (CMA) to production. Migration Plans for scheduler batch stream definitions are available for review.

Jobs that run past midnight must have same business date as other jobs in a stream

In situations where a scheduler batch stream runs too long and has some jobs run past midnight, all jobs in the scheduler batch stream can be set to run with same business date. Set the business date option value on the scheduler batch stream definition to 'SYSDATE' and the scheduler engine processes all the steps in the scheduler batch stream with same date when the scheduler batch stream started.

How to split a big stream into multiple smaller stream

If the Scheduler Batch Stream Definition is too long with many jobs to run, it can be split into multiple smaller streams such that one stream runs after the other. This can be done using a new batch control K1-RNSBS (Run Scheduler Batch Stream). Create a Batch Scheduler Batch Control for every sub stream, specify the batch control as K1-RNSBS, batch parameter name as the sub stream name. Use this Batch Scheduler Batch Control as a last step on the previous stream before it ends so that as soon as the previous stream's steps are completed, the last step will initiate the new sub stream. For example, if you have a stream with jobs A,B,C,B,E,F,G,H. Split the stream into stream 1 with A,B,C,D and stream 2 with E,F,G,H. Create a Batch Scheduler Batch Control with batch control K1-RNSBS, batch parameter name as stream 2. On stream 1 after step D, define a new step that invokes the created Batch Scheduler Batch Control. Note that there are no dependencies between sub-streams. That is each sub stream would be kicked off by the new batch control to run on it's own and the stream that called the sub stream does not wait for it to finish. So such configuration is to be used with only those steps that run in parallel.

> **Note**
>
> Sub-streams follow the main stream's schedule. Schedule on a sub-stream is generally not supported.

How to configure scheduler batch stream definitions to run only on OUAF working calendar dates

To setup Scheduler Batch Stream Definition to run only on working days, use the batch control K1-CWCEL. Create a Batch Scheduler Batch Control with this batch control, batch parameter Work Calendar set to required work calendar, Verify that **Run On Business Day** is set to true. Use this Batch Scheduler Batch Control as the first step to run on the stream. Define the stream to end if this step fails. Define the stream's subsequent step to execute if this step succeeds. When the stream runs on a scheduled day, the first step checks if the day falls on a working day as per the work calendar. If the day is not a working day then the stream ends execution ensuring stream runs only on desired dates.

Scheduler batch stream definitions that check existence of a file

In situations where a scheduler batch stream definition needs to check if a file has been uploaded/created by another process before it can continue with rest of it's processing, a special batch program K1-OSCFE is available. A stream step can be introduced referring to this new batch program with parameter specifying which object storage bucket to check the file in and also status to return if the file does not exist. This stream step can be added right before the step that has dependency on checking the existence of the file and continue processing or not.

Copyright © 2026, Oracle and/or its affiliates.

Handling scheduler batch stream definitions inter-dependencies

In situations where a scheduler batch stream should not run when another scheduler batch stream is running, the inter-dependencies between such scheduler batch stream definitions can be configured by using a special batch program K1-CKCCR. Following steps are needed to setup such configuration.

- The new batch program accepts two parameters script name and script data. Write a CM script that checks whether a dependent job stream is running. If it's running, then the script should terminate with an error.
- On the scheduler batch stream that needs to be paused/hold if another job stream is running, add a new stream step as the first step of the job stream. Use the new batch program on the first step(by setting the CM script as the input parameter) or create a batch scheduler batch control if the batch control is to be used across many job streams with different CM service scripts as input parameters. Set the dependent job stream name as the script data parameter.
- Update the "End Of Stream" step condition such that if first step has failed, end the stream. This ensures the scheduler batch stream does not run if the other dependent scheduler batch stream is already running.
- Update the step(s) intended to execute after the first step, the step condition that if first step succeeded only then these steps should continue.

> **Note**
>
> The special batch program is not restricted to only these situations. It can be used in many custom scenarios by writing the corresponding CM script that terminates if a desired situation is met.

System Override Date Support

This section provides information on system override date support.

A System Date can be overridden at either system level or user level or both. SaaS Batch Scheduler supports the system date override at system level only. The scheduler batch streams are scheduled and processed including submitting batch jobs on the stream steps per system date override. Any online operations performed by user on the scheduler batch stream operations portal such as skip, pause, cancel etc are also processed per system date override.

> **Note**
>
> When system date is overridden, scheduler batch stream definitions that are already scheduled and have next scheduled date set on them do not automatically get the new scheduled date per the overridden system date. This means the next run would be per the current date/time. The overridden system date reflects on the subsequent runs after the scheduler batch stream runs once. If the system override date needs to take effect immediately, the stream's schedule can be updated. This reset/blanks out the next scheduled run so it gets determined immediately per system overridden date.

Batch Scheduler Batch Control

Copyright © 2026, Oracle and/or its affiliates.

This section provides information on using batch scheduler batch controls.

- Configuring a New Batch Scheduler Batch Control

Configuring a New Batch Scheduler Batch Control

In order to configure a new Batch Scheduler Batch Control, from the **Admin** menu, select **Batch Operations**, and select **Batch Scheduler Batch Control**. Use the **Batch Scheduler Batch Control** portal to search for and view existing batch scheduler batch controls.

A Batch Scheduler Batch Control is an entity that holds details of a batch control. Typically a scheduler batch stream definition is defined with batch control specified directly on it. In this case the batch parameters on the batch control are used in batch job submissions. A Batch Scheduler Batch Control can be created when batch parameters need to be customized different from the default batch parameters. The Batch Scheduler Batch Control can then be used on the Scheduler Batch Stream Definition. Only active batch scheduler batch controls can be used on the Scheduler Batch Stream Definition.

Scheduler Batch Stream Operations

The **Scheduler Batch Stream Operations** portal provides an overview of scheduler batch streams that are currently running or suspended or completed within the last week.

To navigate to Scheduler Batch Streams Operations portal, from the **Admin** menu, select **Batch Operations**, and select **Scheduler Batch Stream Operations**.

By default the Running Search By option is displayed where streams that are running can be searched. Broadcast any of the stream run for more information on status of each of its steps.

Two more Search By options are available to search the scheduler batch streams â€” 1. Suspended 2. Completed

Running Search By Option: This option displays Scheduler Batch Stream runs that are In Progress, Stopping, Pausing or Canceling statuses. Refer to Monitoring A Running Scheduler Batch Stream for more information.

Suspended Search By Option: This option displays Scheduler Batch Stream runs that are Failed, Stopped or Paused statuses. Refer to Monitoring A Suspended Scheduler Batch Stream for more information.

Completed Search By Option: This option displays Scheduler Batch Stream runs that are Ended or Canceled statuses. Refer to Monitoring A Completed Scheduler Batch Stream for more information

- Monitoring a Running Scheduler Batch Stream
- Monitoring a Suspended Scheduler Batch Stream
- Viewing a Completed Scheduler Batch Stream
- Handling Exceptions

Monitoring a Running Scheduler Batch Stream

A Scheduler Batch Stream is considered Running if it is one of the following statuses.

- In Progress

Copyright © 2026, Oracle and/or its affiliates.

- Stopping
- Pausing
- Canceling
- Ending

You can monitor the progress of a running Scheduler Batch Stream by broadcasting it from the Running search by option search results. The broadcasted zone **View Stream Steps** shows progress of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Streams Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

> **Note**
>
> A stream is set to Ending status if stream is In Progress and the End Of Stream step has ended. This means some of the steps are still either in In Progress or Submitted status. An Ending stream is set to Ended status once no more steps are In Progress or Submitted status.

Stream Operations

When performing operations on a scheduler batch stream, the operation is not immediately processed real-time. User can only make a request for a **desired** operation. SaaS Batch Scheduler processes these requests and updates scheduler batch stream run accordingly.

Following operations are allowed on the Running scheduler batch streams.

- *Cancel:* A scheduler batch stream run can be canceled while in In Progress or Pausing state. When this operation is performed, scheduler batch stream is set to Canceling status. SaaS Batch Scheduler processes these requests and submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Canceled status. A Canceled scheduler batch stream is removed from this Running streams list and is displayed on the Completed streams list.
- *Stop:* A scheduler batch stream run can be stopped while in In Progress state. When this operation is performed, scheduler batch stream is set to Stopping status. SaaS Batch Scheduler processes these requests and submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Stopped status. A Stopped scheduler batch stream is removed from this Running streams list and is displayed on the Suspended streams list

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note the difference between canceling a scheduler batch stream and stopping a scheduler batch stream. Canceling a scheduler batch stream cancels the steps in progress and cancels the scheduler batch stream so rest of the stream steps do not run permanently. Stopping a scheduler batch stream stops/cancels the stream steps that are In Progress and stops the stream execution temporarily. A stopped stream can be re-started from the Suspended Streams list on the Scheduler Batch Streams Operations Suspended Search By Option. Re-starting a stopped stream re-starts the steps that were stopped and submits the batch jobs associated with those steps to continue rest of the processing of the scheduler batch stream.

- *Pause:* A scheduler batch stream can be paused while In Progress state. The scheduler batch stream is set to Pausing status. SaaS Batch Scheduler processes these requests and pauses the scheduler batch stream once stream steps that are In Progress have completed. While the steps are In Progress, the scheduler batch stream is set to Pausing status. Once steps are complete, the scheduler batch stream is set to Paused status. If the stream steps that are In Progress are the last set of steps to be processed, then there are no steps remaining to pause the scheduler batch stream. Scheduler batch stream is not Paused and completes processing.

> **Note**
>
> Note the difference between stopping a scheduler batch stream and pausing a scheduler batch stream. Stopping a scheduler batch stream cancels the steps that are in progress and stops the stream execution temporarily. Pausing a scheduler batch stream, the steps In Progress continue execution and complete and stream is paused after that. Rest of the stream execution is on hold temporarily.

- *Interrupt:* An interrupt operation is used when the scheduler batch stream is stuck in In Progress state for long period of time and is not progressing. When used, the interrupt operation cancels the steps that are currently In Progress and re-submits the batch jobs on those steps so that the rest of the steps in the scheduler batch stream can be processed.

Stream Step Operations:

- *Skip:* A stream step(s) can be skipped by using the skip multi-action button on this zone. Only steps that are in "Not Submitted" state can be skipped.

The stream steps can be in one of the following states:

- Not Submitted
-  Submitted
-  In Progress
-  Ended
-  Skipped

Monitoring a Suspended Scheduler Batch Stream

Copyright © 2026, Oracle and/or its affiliates.

A Scheduler Batch Stream is considered suspended if it's stopped running temporarily. A Scheduler Batch Stream is considered Suspended if it is one of the following statuses.

- Failed
- Stopped
- Paused

You can monitor the progress of a suspended Scheduler Batch Stream by broadcasting it from the Suspended search by option search results. The broadcasted zone **View Stream Steps** shows progress of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Stream Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

Stream Operations

When performing operations on a scheduler batch stream, the operation is not immediately processed real-time. User can only make a request for a **desired** operation. SaaS Batch Scheduler processes these requests and updates scheduler batch stream run accordingly.

Following operations are allowed on the Running scheduler batch streams.

- *Resume:* A scheduler batch stream run can be resumed from Paused state. When this operation is performed, scheduler batch stream is set to In Progress status. SaaS Batch Scheduler processes these requests and starts processing the rest of the steps in the scheduler batch stream. A resumed scheduler batch stream is removed from this Suspended streams list and is displayed on the Running streams list.
- *Restart Stream:* A scheduler batch stream run can be restarted while in Failed or Stopped state. When this operation is performed, scheduler batch stream is set to In Progress status. SaaS Batch Scheduler processes these requests and starts processing the rest of the steps starting with Failed or Stopped steps. A restarted scheduler batch stream is removed from this Suspended streams list and is displayed on the Running streams list
- *Cancel:* A scheduler batch stream run can be canceled while in any one of the Suspended states. When this operation is performed, scheduler batch stream is set to Canceling status. SaaS Batch Scheduler processes these requests and in case of Failed scheduler batch stream, submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Canceled status. A Canceled scheduler batch stream is removed from this Running streams list and is displayed on the Completed streams list.

Stream Step Operations:

- Skip: A stream step(s) can be skipped by using the skip multi-action button on this zone. Only steps that are in "Not Submitted" state can be skipped.

The stream steps can be in one of the following states:

- Not Submitted
-  Submitted

Copyright © 2026, Oracle and/or its affiliates.

-  Ended
-  Failed
-  Stopped
-  Skipped

Viewing a Completed Scheduler Batch Stream

A Scheduler Batch Stream is considered Completed if it is one of the following statuses.

- Ended
- Canceled

You can view completed Scheduler Batch Stream by broadcasting it from the Completed search by option search results. The broadcasted zone **View Stream Steps** shows status of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Stream Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

The stream steps can be in one of the following states:

- Not Submitted
-  Submitted
-  Ended
-  Failed
-  Skipped
-  Canceled

Handling Exceptions

Errors in Scheduler Batch Stream

A batch job submitted as part of a Scheduler Batch Stream can run into error with one of its threads being in error. It could be because of business scenario or environment issue etc. In such a case, the corresponding Scheduler Batch Stream stops execution and its step ends in failed state. The cause for the failure is noted on the Additional Details column on the stream step. You can fix the issue and restart the stream via the multi-select **Restart Stream** option on

Copyright © 2026, Oracle and/or its affiliates.

the scheduler batch stream operations completed query option. Restarting a stream will result in the Scheduler Batch Stream continuing execution from the step that failed and finish through to end.

> **Note**
>
> Once a Scheduler Batch Stream fails, its future scheduled runs no longer execute until the failed run is addressed.

Scheduler Batch Stream stuck in running status

A Scheduler Batch Stream can be stuck in Running status as one of its steps is in running status for too long. In this situation there are various options.

Cancelling a scheduler batch stream: The corresponding batch job submission is stuck in pending status as the thread pool name set on the batch control is invalid. In such case, the running Scheduler Batch Stream can be cancelled via the multi-select **Cancel** option. The thread pool name can then be set to a valid value and the Scheduler Batch Stream can be submitted manually via the **Run Manually** option on the Scheduler Batch Stream Definition or wait until it's scheduled to run next.

Interrupting a scheduler batch stream: The corresponding batch job submission is stuck in started status while the respective batch run or batch threads have ended. In this case, the running scheduler batch stream can be interrupted via the multi-select **Interrupt** option. This sets the scheduler batch stream and respective steps in the "Failed" state. The **Restart Stream** multi-select option can then be used to re-start the stream to continue execution from the step that failed and complete processing of stream.

Batch Queues Overview

Batch Queues is a new portal where users can view the status and progress of various batch jobs in the system. The portal provides visibility for users into what batch jobs are queued or running on existing thread pools DEFAULT and NOCACHE.

Users can view each of the batch job's details and broadcast them to view list of threads, their status and records details.

The list of batch jobs can be filtered by status, thread pool and batch control to view desired batch jobs.

Batch Alerts

This chapter provides information and requirements on batch alerts in support of Oracle Utilities Cloud implementations.

- Batch Alerts Overview
- Alerting On A Batch Job Stream
- Alerting On A Batch Job

Copyright © 2026, Oracle and/or its affiliates.

Batch Alerts Overview

Batch Alerts provide alerting and monitoring mechanisms for batch and batch job streams. Alerts are raised when desired batch jobs or batch job streams do not perform as expected.

Currently there are few level of service algorithms on batch controls to indicate how the corresponding batch jobs are doing.

New level of service algorithms are now introduced on the batch controls.

In addition, level of service algorithms are introduced on batch job streams to determine overall health of batch job streams.

New installation options health check algorithms are introduced for health component batch and batch job stream. Each health component's health check algorithm will invoke their level of service algorithms and return the overall results

Overall results from health check algorithms are displayed on the health check portal for monitoring purposes to view overall health of the system.

To receive alerts on overall health of the system, use the alert probe - bash script external utility that probes the system and sends email notifications to configurable set of email addresses.

The alert probe can be run on windows or linux machines. Alerts can be customized for errors and warnings. A paging alert is sent in case of errors. A non-paging alert is sent in case of warnings.

The frequency at which alerts are raised is configurable.

> **Note**
>
> The alert probe is provided on an use-as-is basis, no support is provided for it.

Alerting On A Batch Job Stream

To receive alerts on an active batch job stream, edit the batch job stream and add the desired level of service algorithm on it.

> **Note**
>
> Only algorithm types are delivered for Batch job stream level of service algorithms. A corresponding algorithm must be created before it can be added onto the batch job stream

Following level of services are currently supported on batch job streams

- Batch Job Stream has not started within 'X' minutes

Copyright © 2026, Oracle and/or its affiliates.

- Batch Job Stream has failed
- Batch Job Stream ran too long (relative run)

Alerting On A Batch Job

Additional level of services are included on batch job that will be part of alerting mechanism

- Batch Job has processed low number of records
- Batch Job throughput is too low
- Batch Job ran too long (relative run)
- Batch Job has not started within 'X' minutes

ILM Dashboard for Cloud Implementations

This chapter provides information on ILM Dashboard in support of Oracle Utilities Cloud implementations.

- ILM Dashboard Overview
- ILM Dashboard - Main Tab
- ILM Dashboard - Partitions Tab
- ILM Dashboard - Drop A Partition

ILM Dashboard Overview

ILM Dashboard is a feature that provides an all-in-one portal to view Information Lifecycle Management (ILM) configurations and manage ILM tasks in one place. ILM Dashboard has two tabs - Main tab and Partitions tab.

Main tab displays information on current ILM configuration, if any, current retention period of maintenance objects and status of ILM batch processes

Partitions tab displays information on current partition data in the system for all maintenance objects including the sub-partition data.

ILM Dashboard - Main Tab

The ILM Dashboard - Main tab has information on current ILM configurations, retention period on maintenance objects and ILM batch processes current status.

ILM Configuration: Displays the current ILM configuration in the system as well as the default retention period for all ILM eligible maintenance objects.

Retention Period Overridden Maintenance Objects: Displays the list of ILM eligible maintenance objects where the retention period is overridden and their current retention period.

Copyright © 2026, Oracle and/or its affiliates.

ILM Batch Jobs Status: Displays the list of ILM Batch Controls and their latest batch job/run details. ILM Batch Controls that never ran even once are displayed first followed by the ones that ran in descending order of their last run date/time.

ILM Batch Runs: Displays the batch runs for the ILM batch control that was broadcasted on the "ILM Batch Jobs Status" zone. The batch runs are displayed sorted by end date/time with recently ended ones displayed first. Each run's first three batch parameter values are displayed on it's run. Customers can navigate to respective batch job submission page to view list of all parameters and their values.

ILM Dashboard - Partitions Tab

The ILM Dashboard - Partitions tab displays information on current partitions in the system for all maintenance objects that are ILM enabled.

Maintenance Objects: Displays a drop down with list of maintenance objects that are ILM enabled. Selecting a maintenance object displays the maintenance object along with it's partition information such as all partitions Total Size (In GB) and Total Records Based On Last Stats. If no maintenance object is selected, all maintenance objects are displayed along with their partition information.

Checking the PMAX partitions with data displays only those maintenance objects and their PMAX partitions information that have data in the PMAX partitions. If a maintenance object and PMAX partitions with data is selected, only the PMAX partitions information of the PMAX partitions that have data in it are displayed for the selected maintenance object.

Partitions: Displays list of partitions and their details for the selected maintenance object. The partitions are sorted by partition name latest month first. By default, the last twelve months and Pmax partition are displayed by default. User can further search for historical partitions using the filter options upto last four years.

For each partition, it's Compression Type is displayed. Compression Type for each displays "Refer Sub-partition" if the partition does not have compression but one of the sub-partitions does.

For a maintenance object, at any given time, the oldest partition can be requested to be dropped via the "Drop Partition" button. A new "Partition Drop Request" object is then created for this request to go through the approval process. Refer to Dropping A Partition for more information on request to drop partitions and it's approval process.

Sub-partitions: Displays list of sub-partitions and their details for the selected partition. The sub-partitions are sorted by sub-partition name latest month first.

Partition Drop Requests: Displays list of partition drop requests along with it's current status and applicable actions for the current user. Only displayed if there are partition drop request records. Displays partition drop requests for the selected maintenance object, if none is selected then all partition drop requests for all maintenance objects are displayed.

By default, only partition drop requests in Pending and Error status are displayed. For partition drop requests in other statuses, user can use the filter options. Refer to Dropping A Partition for additional information.

ILM Dashboard - Drop A Partition

The ILM Dashboard - Partitions tab displays information on current partitions in the system for all Maintenance Objects that are ILM enabled.

Copyright © 2026, Oracle and/or its affiliates.

For a Maintenance Object, at any given time, the oldest partition of the primary table can be requested to be dropped via the "Drop Partition" button by a **Requester** user. A new "Partition Drop Request" object is then created for this request in Pending status and a To Do is created for an Approver user so the request goes through the approval process.

Requesters

A user must be setup as a Requester user in order to drop a primary table's oldest partition for a selected Maintenance Object at any given point. Please refer to Administrative User Guide for more information on how to setup these users.

> **Note**
>
> A pre-requisite exists for dropping a primary table's oldest partitions with data. The crawler Batch Control associated with the Maintenance Object must be run first prior to requesting the partition drop, so that relevant data is appropriately marked for the partition to be dropped.

> **Note**
>
> When oldest partition of the primary table for a maintenance object is requested to dropped, the oldest partitions of child tables for that maintenance object are also dropped. This is based on the referenced partitions setup for the maintenance object. If the maintenance object does not have referenced partitions setup, then only the primary table's oldest partition is dropped.

The "Partition Drop Request" object is initially in Pending status and displayed to **Approver** users in the "Partition Drop Requests" zone.

Approvers

A user must be setup as an Approver user for them to approve any Partition Drop Requests.Please refer to Administrative User Guide for more information on how to setup these users.

Viewing Drop Partition Requests

Only the Requester or Approver can view Drop Partition Requests based on their access. By default, only Pending or Error Drop Partition Requests are displayed. User can use the filter critera option to display all Drop Partition Requests or search for ones with different statuses. Depending on the user's access, the available and applicable actions are displayed to the user.

When the Approver views list of Partition Drop Requests, they have the option of either approving or rejecting the request.

Approving a request results in Partition Drop Request in In Progress status and completing the To Do previously created for that Approver. A batch job submission for Batch Control K1-ILMDR (ILM Automation - Drop Partition) is also initiated to drop the requested partition. Once the partition is successfully dropped, the Partition Drop Request transitions to Completed final status. Otherwise, if there are any errors, the Partition Drop Request transitions to Error status.

Rejecting a request requires specifying a reason as to why the request was rejected. Possible reasons for rejecting a request are 1. Approver determines that partition that is requested to drop is required and so it should not be dropped. 2. Incorrect request to drop a partition was made where Requester accidentally requested wrong partition to be dropped. Once the Approver rejects the request, the Partition Drop Request transitions to Rejected final status.

Copyright © 2026, Oracle and/or its affiliates.

When the Partition Drop Request is in Error status, the Requester has the option to re-try the request again or cancel the request. In the case of re-try option, Requester can review the errors, take any necessary steps needed and and re-try the Partition Drop Request again. For example, if the pre-requisite step was not taken prior to requesting a partition drop, the data in the partition will not be marked for dropping. In this case, an error would be generated when the partition with that data is requested to be dropped. Requester can review the error to confirm the reason and run the crawler Batch Control associated with the Maintenance Object to mark partition data to be dropped. Once it is complete, Requester can re-try the Partition Drop Request again.

A Partition Drop Request in Pending status can be canceled by the Requester user if desired. A Requester user can also cancel the Partition Drop Request that is in Error status if they wish to not re-try the request.

Recommendations

It is recommended that dropping partitions is tested thoroughly in non-production environments first before performing the same operation in a production environment.

Tips

- Once a partition is dropped, the data is also removed from the database and there is no way to restore the data back into the environment. It is imperative that customers review the partition and the respective data before requesting and approving the Partition Drop Request.
- Customers can optionally choose to raise an SR(Service Request) to request a database copy prior to dropping a partition

Oracle Cloud Infrastructure Identity and Access Management Integration

This chapter provides information about integration with Oracle Cloud Infrastructure Identity and Access Management for Oracle Utilities cloud services.

> **Note**
>
> This section does not apply to all implementations. Contact your system administrator to confirm if your implementation includes Oracle Utilities Cloud Service Foundation and Oracle Cloud Infrastructure Identity and Access Management (IAM) integration.

- Identity and Access Management Integration Overview
- Configuring Identity and Access Management Integration

Identity and Access Management Integration Overview

Oracle Cloud Infrastructure Identity and Access Management (IAM) with Identity Domains provides identity and access management functionality for the Oracle Utilities cloud services and supports single sign on (SSO) and identity federation capabilities.

It protects both online and webservices API access to the Oracle Utilities cloud service. Refer to the Detailed Oracle Cloud Services Administration guide for the instructions on how to

Copyright © 2026, Oracle and/or its affiliates.

configure and manage online and API access and maintain users, groups, application roles and integration OAuth clients.

In IAM, an identity domain is provided with each service subscription. Once provisioned, the identity domain is administered exclusively by the client.

Security administrator uses identity domains to manage application users, who are given access to a one or multiple application environment(s). The security administrator may also assign another user to an administrative role in the identity domain and delegate user management privileges.

Newly created users receive an account activation email from IAM, and must reset their password.

The integration with IAM supports Just-In-Time User Provisioning.

When a user attempts to access the utility application URL, the user is redirected to an identity domain for authentication. Once the user is successfully authenticated, they are redirected back to the application and the authentication token is evaluated and validated. The system then checks if the user already exists in the application, and if not, it triggers the Just-In-Time User Provisioning logic that creates and activates the new user.

The **F1-IDMUser** business object is used for creation of new user records sourced from the external identity management integrations.

For the integration with Oracle Cloud Infrastructure Identity and Access Management, the algorithm is defined on the **F1-IDMUser** business object to determine the Template User whose information should be copied to the newly created user.

- Just In Time User Provisioning

Just In Time User Provisioning

Just-In-Time User Provisioning is a process of creating a new user record upon first successful login. The implementation may follow either one of the main approaches:

- **Just-In-Time with Coarse-Grained User Authorization** allows security administrator to immediately begin working on the authorization and access setup, while other ("business") users might wait until the proper access and authorization is configured for them in the application.
- **Just-In-Time with Fine-Grained User Authorization** provides both security administrator and the business user with instant access to the appropriate application functionality.
- Coarse-Grained User Authorization (Default)
- Fine-Grained User Authorization

Coarse-Grained User Authorization (Default)

With course-grained user authorization, Just-In-Time Provisioning creates users with two authorization levels:

- **Security and Access Administrator.**
  - **Users with an administrative role in the identity domain** are provisioned using a **Security Administrator Template User**. The user gains the access to all transactions and services related to application security and user authorization setup
- **Authenticated User**

Copyright © 2026, Oracle and/or its affiliates.

- **Users with no administrative role(s) in the identity domain** are provisioned using a **Minimum Access Template User**. The user gains access to *My Preferences* page only. The security and access administrator completes the authorization and access setup manually.

No additional configuration effort is needed if your implementation chose to follow this approach. The scope of the actual security access for both Security Administrator and Authenticated User is customizable. Refer to Override Default Access for more information.

Fine-Grained User Authorization

With fine-grained user authorization,Just-In-Time Provisioning creates users with multiple authorization levels, according to their Identity Domain's Group membership and based on utility application configuration:

- **Security and Access Administrator**
  - **Users with an administrative role in the identity domain** are provisioned using a **Security Administrator Template User**. The user gains the access to all transactions and services related to application security and user authorization setup
- **Business/Administrative Users** with access to a specific business and/or administrative functions in the application
  - **Users that are members of a Group in the identity domain** are provisioned using a **Template User mapped to this Group**. The user gains access to the application according to the Template User's setup.
- **Authenticated User**
  - **Users with no administrative role(s) in identity domain** and **no membership in any of the identity domain's Groups** are provisioned using the **Minimum Access Template User**. The user gains access to *My Preferences* page only. The security and access administrator completes the authorization and access setup manually.

The mapping between identity domain's Groups and Template Users is stored in theIdentity and Access Management Integration Configuration.

Configuring Identity and Access Management Integration

The integration configuration in the utility application has to be done in conjunction with the corresponding setup in the Identity and Access Management system.

- Configuring Just In Time User Provisioning
- Customizing the Integration

Configuring Just In Time User Provisioning

In order to configure user authorization for Just in Time Provisioning, perform the following steps:

- Create User Groups that represent most typical business and/or administrative roles in the application
- Create Template Users that represent typical access level for major business and/or administrative functions

Copyright © 2026, Oracle and/or its affiliates.

- Request your customer's identity domain Administrator to create Groups in IAM with Identity Domains that will correspond to the Template Users create above
- Setup the mapping between the identity domain's Groups and Template Users using Identity and Access Management Integration Configuration
- Template Users and User Groups
- Identity and Access Management Integration Configuration

Template Users and User Groups

- Define User Group(s) that represent the major business and/or administrative responsibilities in the application
- Define Template User(s) that represent the typical level of access to the application functionality

Identity and Access Management Integration Configuration

The Identity and Access Management Integration Configuration master configuration stores the definitions related to the integration with Oracle Cloud Infrastructure Identity and Access Management.

For Just-In-Time User Provisioning, configure the **User Provision Rules**:

- Map the Groups defined in the identity domain to the Template Users defined in the application.

Customizing the Integration

The product is delivered with out-of-the-box support for Just in Time User Provisioning with Coarse-Grained Authorization. Possible customization scopes are:

- Customize the default access level for Coarse-Grained Authorization. See Override Default Access Level for details.
- Configure the Just in Time User Provisioning with Fine-Grained Authorization.
- Override Default Access Level
- Override Logic to Determine Template User

Override Default Access Level

Both Security Administrator and Minimum Access Template Users are specified as parameters for the K1-IDCSUSRTM algorithm .

To override the default access level:

- Create new Template Users and setup user groups for default administrative access
- Create new Template Users and setup user groups for default minimum access

Create a new version of the K1-IDCSUSRTM algorithm and specify the Template Users created above as the algorithm's parameters

Copyright © 2026, Oracle and/or its affiliates.

Override Logic to Determine Template User

The algorithm that determines the Template User is implemented as a pre-processing algorithm that determines the Template User on the **F1-IDMUser.** business object.

To override the logic used to determine the template user:

- Create a custom algorithm type and algorithm and use custom logic to determine the Template User for provisioning.
- Deactivate the existing product algorithm and specify the custom algorithm instead. Note that the algorithm that determines the Template User should be plugged before the algorithm that populates the data from the Template User.

Oracle Cloud Object Storage

Oracle Cloud Object Storage (Object Storage) is a part of Oracle Cloud Infrastructure Storage Services.

Oracle Cloud Object Storage can be accessed and managed using Oracle Cloud Infrastructure Console or programmatically using Command Line Interface (CLI), REST APIs or Java. The Object Storage that is used by the system is owned and managed by customers.

Oracle Utilities Cloud Services use Object Storage as the file storage location for all file related integrations. Batch processes that need to read or write files as input will access Object Storage.

The system requires some configuration in order to allow it to connect to the Object Storage and have access to the files that need to be processed by it. The following chapter describes the required configuration in order to connect to Object Storage as well as provides information on how to reference files in Object Storage in batch processes.

Please refer to Oracle Cloud Infrastructure documentation for more information on Object Storage.

- File Storage Configuration for Object Storage
- Referencing Files on Object Storage

File Storage Configuration for Object Storage

In order for the system to connect to Object Storage a File Storage Extendable Lookup has to be created with the following information:

- The Oracle Cloud Infrastructure tenancy details, which can be obtained from the Oracle Cloud Infrastructure Console(available to customers who purchased Object Storage).
- The API Key that is used by the system to connect to that Object Storage.

Refer to product documentation on "File Storage" for more details.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> - Each File Storage Location configuration refers to a combination of Object Storage Tenancy, Compartment and User ID.Since customers can choose to create many Compartments in their Object Storage, access to each such Compartment will require a separate File Storage configuration.
> - It is also recommended that each File Location configuration will reference a **different** Cloud Infrastructure User ID.This can limit, for example, the system's access to production files from a non-production environment.
> - API Keys are managed using Key Rings. Key Rings can be shared between File Storage Extendable Lookups but the Keys in these Key Rings cannot be shared across system environments. For example, keys generated for the Development environment **cannot** be used for the Testing or Production environments.Refer to the following topics in the Framework Administrative User Guide:
>
>   **External File Storage**, **Key Rings**.

Referencing Files on Object Storage

Access to files is typically required in Batch processes. Some batch processes include direct reference to the files they need to process (for input or output) while some processes rely on additional configuration (for example, the Content Migration Assistant (CMA) Import and Export processes that rely on the CMA Master Configuration Settings).

Direct Reference

Reference to Object Storage can be used anywhere that a file location is allowed. The format is:

file-storage://<File Location>/<Bucket>/<Filename.ext>

- <File-Location>: The File Storage Extendable Lookup value that is defined for that file.It should include all the details the system needs to access the Object Storage location for that file.
- <Bucket>: The equivalent of a Folder in the Oracle Cloud Object Storage.Since the folder structure in Object Storage is flat, Buckets cannot include other Buckets and therefore only one Bucket can be noted here.

CMA Master Configuration

CMA Master Configuration includes the location of files for export and import processing:

- Import Folder: should reference a file location for an Object Storage Bucket(i.e. just the file location without the a file name, for example: "file-storage://CMALocation/CMA-Files"), "CMALocation" is the File Location Extendable Lookup name.
- Export Folder: should reference a file location for an Object Storage Bucket (uch as "file-storage://CMALocation/CMA-Files").

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In order to take advantage of the automation of migration of configuration data from one system environment to another via the Process Automation Tool, the Import and Export directories have to point to the same location!

Conversion Data Upload Processes

The batch process that loads customer data files into the cloud (K1-CNVLD) using SQL Loader has two parameters that describes where to get the input data files and where to store the output files:

- inputFileStorage: should reference a file location for an Object Storage Bucket (such as "file-storage://CONVLocation/Input"), "CONVLocation" is the File Location Extendable Lookup name.
- outputFileStorage: should reference a file location for an Object Storage Bucket (such as file-storage://CONVLocation/Output).

Maintaining Customer Code

The system supports a process to import custom Java code to be deployed in the cloud under certain use cases.

> **Note**
>
> Refer to the Cloud Services Implementation Guide for information about how to use this functionality.

The Customer Code portal allows you to review the status of the lifecycle of the process to import code and allows you to review errors that may have been received.

To view a customer code record, navigate using **Admin**, then **Implementation Tools**, then **Customer Code**. You are brought to a query portal with options for searching for customer code. Once a record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Copyright © 2026, Oracle and/or its affiliates.

The following sections describe the features of this zone.

Search Filters

The following table describes each of the different filters supported by this zone.

Search Filter	Description
To Do Type	Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection. If the To Do Role filter is populated, the list is further restricted to valid To Do Types for the To Do Role.
Status	Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries.
Days Old	On the **To Do Management** portal, this option is represented by a pair of filter fields. Use these filters to restrict the search to To Do entries whose age falls within a given range in days. For example, entering values of 2 - 5 will restrict the search to entries that are greater than or equal to two days old and less than or equal to five days old. On the **To Do Dashboard** portal, this option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days.
To Do Role	Use this filter to restrict the search to To Do entries of a specific role. Only To Do Roles where the user is a member will be available for selection. If the To Do Type filter is populated, the list is further restricted to valid To Do Roles for the To Do Type.
Assign To User	Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access.
Priority	Use this filter to restrict the search to To Do entries with a specific priority.
Key Value	Use this filter to restrict the search to To Do entries with a sort key value containing the entered value.
Message	Use this filter to restrict the search to To Do entries with a specific message. This filter appears after the To Do Type filter has been populated and shows a list of the message texts from all non closed To Do entries of the selected To Do Type.
Characteristic	Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. If the To Do Type filter is populated and the To Do Type is 'manual', the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown. If an automated To Do Type doesn't define valid characteristics or if no To Do Type filter is populated, all valid characteristic types for the To Do Entry entity are shown. Note that if you have populated characteristic filters and then you change the To Do Type filter, the system retains the characteristic filter as long as the characteristics are applicable to the newly selected To Do Type.

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> For performance reasons, because the filter criteria may be broad, the search zone is configured to only get the first 5000 records that match the criteria and will then sort that number to then display the first 400 sorted by priority and age. The To Do Dashboard portal may be used to get exact counts of records with certain criteria. Use additional criteria to narrow down the search as needed.

Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined. The following columns are displayed initially:

- The Action column includes the button Work. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Important**
>
> **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

> **Note**
>
> **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.

> **Note**
>
> Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.

Copyright © 2026, Oracle and/or its affiliates.

- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string. By default this list is collapsed to one line to save vertical space. If additional characteristics exist, you can use the expand arrow to display them.

> **Note**
>
> A feature configuration option is supplied to opt out of the compact view. Under the **To Do Options** feature configuration type, define the **Collapse Characteristics Column** option and set the value to "N".

- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon  in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.
- **To Do Role** displays the description of the role for the To Do entry.

Actions

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

Copyright © 2026, Oracle and/or its affiliates.